aws

Guide du développeur

Amazon Pinpoint

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

Amazon Pinpoint: Guide du développeur

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

..... x
Qu'est-ce qu'Amazon Pinpoint ? 1
Utilisez Amazon Pinpoint pour envoyer des messages à des segments d'audience et analyser les données 1
Définir des segments de public ciblé 1
Planifiez des campagnes de messagerie 2
Envoyer des messages transactionnels 2
Utilisez les analyses et les rapports de mesures 2
Utilisez les points de terminaison pour définir votre public 3
Ajouter des points de terminaison 4
Exemples 5
Informations connexes 11
Associer des utilisateurs à des points de terminaison 11
Exemples 12
Informations connexes 16
Ajouter un lot de points de terminaison 17
Exemples 17
Informations connexes 25
Importer des points de terminaison 25
Avant de commencer 26
Exemples 26
Informations connexes 39
Exporter des points de terminaison depuis Amazon Pinpoint vers des compartiments Amazon S3 39
Avant de commencer 40
Exemples 40
Informations connexes 51
Rechercher des points de terminaison dans un projet Amazon Pinpoint 52
Exemples 52
Informations connexes 58
Point final de liste IDs 58
Gérez au maximum les terminaux 60
Supprimer des points de terminaison 61
Exemples 61

Création ou importation de segments 65
Créez des segments 65
Créez des segments à l'aide du AWS SDK pour Java 65
Importer des segments 69
Importez un segment avec le AWS SDK pour Java 69
Personnaliser les segments 72
Données d'événement 73
Création d'une fonction Lambda 74
Attribuer une politique de fonction Lambda 77
Attribuer une fonction Lambda à une campagne 79
Créez des campagnes Amazon Pinpoint de manière programmatique 81
Créez une campagne avec le SDK for Java 81
Création d'une campagne A/B de test 84
Gestion des balises 87
Utiliser des balises dans les politiques IAM 88
Ajout de balises aux ressources 89
Ajoutez des balises à l'aide de l'API 90
Ajoutez des balises à l'aide du AWS CLI 91
Afficher les balises pour les ressources 92
Afficher les tags à l'aide de l'API 92
Afficher les balises à l'aide du AWS CLI 93
Mettre à jour ou remplacer les balises 94
Supprimez des balises de ressources 95
Supprimer les tags à l'aide de l'API 95
Supprimez les tags à l'aide du AWS CLI 96
Intégrez à votre application 97
Travailler avec AWS SDKs 98
Connectez votre application frontale à l'aide d'Amplify 99
Étape suivante 99
Enregistrez les points de terminaison dans votre application 99
Avant de commencer 100
AWS mobile SDKs 100
AWS Amplifier 100
Étapes suivantes 101
Signalez des événements dans votre application 101
Avant de commencer 102

AWS mobile SDKs ... 102
Web et React Native ... 100
API d'événements Amazon Pinpoint ... 103
Étapes suivantes ... 103
Envoyez des messages transactionnels depuis votre application ... 104
Envoyer des e-mails transactionnels ... 104
Choisissez une méthode pour envoyer un e-mail ... 105
Choisissez entre Amazon Pinpoint et Amazon SES ... 105
Envoyer un e-mail à l'aide de l'API ... 105
Ajouter des en-têtes de désabonnement aux e-mails ... 120
Envoi de SMS ... 122
Envoi de messages vocaux ... 135
Utiliser l'API SMS et voix ... 144
Générez des mots de passe uniques ... 146
Réponse de `SendOtpMessage` ... 149
Valider les messages OTP ... 150
Réponse de `VerifyOtpMessage` ... 150
Exemples de code OTP dans Amazon Pinpoint ... 151
Générer un ID de référence ... 151
Envoyer des codes OTP ... 152
Valider les codes OTP ... 153
Personnalisez les messages intégrés à l'application ... 155
Récupérez les messages intégrés à l'application pour un point de terminaison ... 155
`GetInAppMessages`Exemple JSON de réponse d'API ... 158
Objet `InAppMessageCampaigns` ... 159
Objet `InAppMessage` ... 161
Objet `HeaderConfig` ... 162
Objet `BodyConfig` ... 163
Objet `InAppMessageContent` ... 164
Objet `Schedule` ... 165
Objet `InAppMessageButton` ... 165
Objet `DefaultButtonConfig` ... 166
Objet `OverrideButtonConfig` ... 168
Validation du numéro de téléphone ... 169
Cas d'utilisation de la validation des numéros de téléphone Amazon Pinpoint ... 169
Validez un numéro de téléphone à l'aide du AWS CLI ... 170

Réponse de validation du numéro de téléphone 171
Créez une chaîne personnalisée 176
Utiliser un webhook 176
Utiliser une fonction Lambda 176
Attribuer une fonction Lambda ou un webhook à une campagne individuelle 178
Création et configuration d'une fonction Lambda pour une campagne Amazon Pinpoint 179
Exemple de fonction Lambda 179
Format de réponse de fonction Lambda pour Amazon Pinpoint 184
Accorder l'autorisation d'invoquer la fonction 185
Diffuser les données des événements de l'application 188
Configurer le streaming des données d'événements 189
Conditions préalables 189
AWS CLI 190
AWS SDK pour Java 190
Flux de données d'événements d'applications provenant d'Amazon Pinpoint 191
Exemple d'événement d'application 191
Attributs d'événement d'application 192
Flux de données sur les événements liés à la campagne depuis Amazon Pinpoint 198
Exemple d'événement de campagne 198
Attributs d'événement de campagne 199
Données sur les événements du parcours provenant d'Amazon Pinpoint 207
Exemple d'événement Journey 207
Attributs de l'événement de parcours 208
Flux de données d'événements par e-mail provenant d'Amazon Pinpoint 213
Exemples d'événements par e-mail 214
Attributs d'événement d'e-mail 220
Flux de données d'événements par SMS depuis Amazon Pinpoint 228
Exemple d'événement SMS 228
Attributs d'événement SMS 229
Supprimer un flux d'événements 239
AWS CLI 239
AWS SDK pour Java 239
Données d'analyse des requêtes 240
Requête de statistiques dans Amazon Pinpoint 241
politiques IAM 242
Mesures standard pour les projets, les campagnes et les parcours 246

Métriques d'application pour les campagnes 248
Métriques d'application pour le courrier électronique 254
Métriques d'application pour les SMS 266
Métriques de campagne 273
Métriques d'engagement du parcours 286
Métriques d'exécution du parcours 294
Métriques d'exécution de l'activité du parcours 296
Métriques relatives à l'exécution des parcours et des campagnes 301
Consultation des données de campagne 305
Conditions préalables 305
Données de requête pour une campagne 307
Données de requête pour plusieurs campagnes 312
Interrogez les données de messagerie transactionnelle 318
Conditions préalables 319
Données de requête pour les e-mails transactionnels 320
Données de requête pour les SMS transactionnels 325
Utiliser les résultats des requêtes JSON 331
Structure JSON 332
Objets et champs JSON 337
Enregistrez les appels d'API avec CloudTrail 339
Informations Amazon Pinpoint dans CloudTrail 339
Actions de l'API dans les fichiers CloudTrail journaux 341
Actions de l'API d'e-mail dans les fichiers CloudTrail journaux 345
Actions d'API SMS et voix v1 prises en charge dans les fichiers CloudTrail journaux 346
CloudTrail exemples de saisie de journal 346
Utiliser un modèle de recommandation 352
Ajouter des recommandations aux messages 352
Invoquer une fonction Lambda pour un modèle de recommandation 355
Données d'événement en entrée 355
Données et exigences de réponse 357
Attribuer une politique pour traiter les données de recommandation 362
Autoriser Amazon Pinpoint à appeler une fonction Lambda 364
Configuration d'un modèle de recommandation 365
Supprimer les données du projet Amazon Pinpoint 366
Supprimer toutes les données du projet Amazon Pinpoint 366
Exemples de code 368

Amazon Pinpoint 369
Principes de base 370
Scénarios 462
API de messages SMS et vocaux Amazon Pinpoint 475
Principes de base 475
Sécurité 496
Protection des données 497
Chiffrement des données 499
Confidentialité du trafic inter-réseau 500
Création d'un point de terminaison de VPC d'interface pour l'API Amazon Pinpoint 501
Gestion des identités et des accès 503
Public ciblé 503
Authentification par des identités 504
Gestion de l’accès à l’aide de politiques 505
Fonctionnement d'Amazon Pinpoint avec IAM 507
Actions liées aux politiques Amazon Pinpoint 514
Identity-based exemples de politiques 546
Rôles IAM pour les tâches courantes 562
Résolution des problèmes 580
Journalisation et surveillance 582
Validation de conformité 583
Résilience 584
Sécurité de l’infrastructure 585
Analyse de la configuration et des vulnérabilités 586
Bonnes pratiques de sécurité 586
Quotas 588
Quotas de projet 588
Quotas de demande d’API 589
Quotas de campagne 591
Quotas d’e-mails 594
Quotas de messages électroniques 594
Quotas d’expéditeurs et de destinataires d’e-mails 594
Quotas d’envoi d’e-mails 596
Quotas de points de terminaison 597
Quotas d’importation de points de terminaison 598
Quotas d’ingestion d’événements 599

Quotas de parcours 600
Quotas Lambda 601
Quotas de machine learning 602
Quotas de modèles de message 604
Quotas de notifications push 605
Quotas de messages dans l'application 606
Quotas de segments 607
Quotas de SMS 607
Quotas de 10DLC 607
Quotas de messages vocaux 608
Demande d’augmentation de quota 608
Historique de la documentation 611
Mises à jour antérieures 621

Avis de fin de support : le 30 octobre 2026, le support d'Amazon Pinpoint AWS prendra fin. Après le 30 octobre 2026, vous ne pourrez plus accéder à la console Amazon Pinpoint ni aux ressources Amazon Pinpoint (points de terminaison, segments, campagnes, parcours et analytique). Pour plus d’informations, consultez Fin de la prise en charge d’Amazon Pinpoint. Remarque : les API liées aux SMS, à la voix, au push mobile, à l'OTP et à la validation des numéros de téléphone ne sont pas affectées par cette modification et sont prises en charge par AWS la messagerie utilisateur final.

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.

Qu'est-ce qu'Amazon Pinpoint ?

Amazon Pinpoint est un AWS service que vous pouvez utiliser pour communiquer avec vos clients via plusieurs canaux de messagerie. Vous pouvez utiliser Amazon Pinpoint pour envoyer des notifications push, des e-mails, des SMS ou des messages vocaux.

Les informations contenues dans ce manuel du développeur sont destinées aux développeurs d'application. Ce manuel contient des informations sur l'utilisation des fonctions d'Amazon Pinpoint par programmation. Il contient également des informations particulièrement intéressantes pour les développeurs d'applications mobiles, tels que les procédures d' intégration des fonctionnalités d'analyse et de messagerie à votre application.

Amazon Pinpoint est disponible dans plusieurs AWS régions d'Amérique du Nord, d'Europe, d'Asie et d'Océanie. Pour plus d'informations sur Régions AWS, consultez la section Gestion Régions AWS dans le Référence générale d'Amazon Web Services. Pour obtenir une liste de toutes les régions dans lesquelles Amazon Pinpoint est actuellement disponible, consultez Points de terminaison et quotas Amazon Pinpoint et Points de terminaison de service AWS dans 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.

Pour plus d'informations sur Amazon Pinpoint, consultez les guides suivants :

- Référence de l'API Amazon Pinpoint
- API SMS et voix Amazon Pinpoint
- Guide de l'utilisateur d'Amazon Pinpoint

Utilisez Amazon Pinpoint pour envoyer des messages à des segments d'audience et analyser les données

Vous pouvez utiliser Amazon Pinpoint pour définir des segments d'audience, envoyer des campagnes de messagerie et des messages transactionnels, et utiliser des indicateurs pour analyser le comportement des utilisateurs.

Définir des segments de public ciblé

Définissez des segments de public ciblé pour envoyer vos messages aux bons destinataires. Un segment identifie les utilisateurs qui reçoivent les messages envoyés par une campagne. Vous

pouvez définir des segments dynamiques basés sur les données rapportées par votre application (système d'exploitation ou type d'appareil mobile, par exemple). Vous pouvez également importer des segments statiques que vous définissez en utilisant un autre service ou une autre application.

Planifiez des campagnes de messagerie

Atteignez votre public ciblé en créant une campagne de communication. Une campagne envoie des messages personnalisés sur la base du planning que vous définissez. Vous pouvez créer des campagnes qui envoient des notifications push mobiles, des e-mails ou des SMS.

Pour essayer d'autres stratégies de campagne, configurez votre campagne comme un test A/B et évaluez les résultats à l'aide des analyses Amazon Pinpoint.

Envoyer des messages transactionnels

Tenez vos clients informés en envoyant directement à des utilisateurs ciblés des SMS et des notifications transactionnelles push mobiles, par exemple des messages d'activation d'un nouveau compte, des confirmations de commande ou des notifications de réinitialisation du mot de passe. Vous pouvez envoyer des messages transactionnels à l'aide de l'API REST Amazon Pinpoint.

Utilisez les analyses et les rapports de mesures

Obtenez des informations sur votre public ciblé et sur l'efficacité de vos campagnes en utilisant les analyses fournies par Amazon Pinpoint. Vous pouvez afficher des tendances sur le niveau d'implication, l'activité d'achat, les données démographiques de vos utilisateurs, et plus encore. Vous pouvez également surveiller le trafic de vos messages en consultant des métriques, telles que le nombre total de messages envoyés ou ouverts pour une campagne ou une application. Grâce à l'API Amazon Pinpoint, votre application peut fournir des données personnalisées, rendues disponibles pour analyse par Amazon Pinpoint. Vous pouvez également interroger des données analytiques pour certaines métriques standard.

Pour analyser ou stocker des données analytiques en dehors d'Amazon Pinpoint, vous pouvez configurer Amazon Pinpoint pour qu'il transmette les données à Amazon Kinesis.

Utilisez des points de terminaison pour représenter votre public dans Amazon Pinpoint

Dans Amazon Pinpoint, chaque membre de votre audience est représenté par un ou plusieurs points de terminaison. Lorsque vous utilisez Amazon Pinpoint pour envoyer un message, vous le redirigez vers des points de terminaison représentant les membres de votre public cible. Chaque définition de point de terminaison inclut une destination de message, comme un jeton d'appareil, une adresse e-mail ou un numéro de téléphone. Elle inclut également des données relatives à vos utilisateurs et à leurs appareils. Avant d'analyser, de segmenter ou d'impliquer votre public, vous devez ajouter des points de terminaison à votre projet Amazon Pinpoint.

Pendant que votre public s'accroît et change, vos données de point de terminaison aussi. Pour consulter les dernières informations dont dispose Amazon Pinpoint sur votre audience, vous pouvez rechercher les points de terminaison individuellement ou exporter tous les points de terminaison d'un projet Amazon Pinpoint. En consultant les données de votre terminal, vous pouvez consulter les informations suivantes concernant vos utilisateurs :

- Leur appareil et leur plateforme.
- Leur fuseau horaire.
- Les versions de votre application installées sur leur appareil.
- L'emplacement de leur ville et de leur pays.
- Autres attributs et statistiques personnalisés que vous enregistrez.

La console Amazon Pinpoint fournit également des analyses pour les données démographiques et les attributs personnalisés qui sont capturés dans vos points de terminaison.

Les rubriques suivantes expliquent comment utiliser les points de terminaison dans Amazon Pinpoint. Pour plus d'informations sur l'ajout automatique de points de terminaison à l'aide de votre appareil Android, iOS ou JavaScript client, consultezEnregistrez les points de terminaison Amazon Pinpoint dans votre application.

Rubriques

- Ajouter des points de terminaison à Amazon Pinpoint
- Associer des utilisateurs à des points de terminaison Amazon Pinpoint
- Ajouter un lot de points de terminaison à Amazon Pinpoint

- Importer des points de terminaison dans Amazon Pinpoint
- Exporter des points de terminaison depuis Amazon Pinpoint vers des compartiments Amazon S3
- Rechercher des points de terminaison dans un projet Amazon Pinpoint
- Répertorier les points IDs de terminaison avec Amazon Pinpoint
- Gérez le nombre maximum de points de terminaison dans Amazon Pinpoint
- Supprimer des points de terminaison d'Amazon Pinpoint par programmation

Ajouter des points de terminaison à Amazon Pinpoint

Un point de terminaison représente une destination à laquelle vous pouvez envoyer un message, comme un appareil mobile, un numéro de téléphone ou une adresse e-mail. Avant de pouvoir envoyer un message à un membre de votre public, vous devez définir un ou plusieurs points de terminaison pour cette personne.

Au fur et à mesure que vous ajoutez des points de terminaison à Amazon Pinpoint, celle-ci se développe comme référentiel de données sur l'audience. Ces données se composent des :

- Points de terminaison que vous ajoutez ou mettez à jour à l'aide de l'API Amazon Pinpoint.
- Points de terminaison que votre code client ajoute ou met à jour au fur et à mesure que les utilisateurs utilisent votre application.

Lorsque vous définissez un point de terminaison, vous spécifiez le canal et l'adresse. Le canal est le type de plateforme que vous utilisez pour envoyer des messages au point de terminaison. Les canaux incluent par exemple un service de notification push, des SMS et des e-mails. L'adresse spécifie à quel l'emplacement envoyer le message au point de terminaison, comme un jeton d'appareil, un numéro de téléphone ou une adresse e-mail.

Pour plus de détails sur votre public, vous pouvez enrichir vos points de terminaison avec des attributs standard et personnalisés. Ces attributs incluent des données concernant vos utilisateurs, leurs préférences, leurs appareils, les versions du client qu'ils utilisent ou leur localisation. Lorsque vous ajoutez ce type de données à vos points de terminaison, vous êtes en mesure de :

- Afficher des graphiques sur votre audience dans la console Amazon Pinpoint.
- Segmenter votre public en fonction des attributs de point de terminaison, afin de pouvoir envoyer vos messages au public cible adéquat.

- Personnaliser vos messages en intégrant des variables de message remplacées par des valeurs d'attributs utilisateur de point de terminaison.

Une application mobile ou JavaScript cliente enregistre automatiquement les points de terminaison si vous intégrez Amazon Pinpoint à l'aide de AWS la bibliothèque SDKs Mobile ou AWS JavaScript Amplify. Le client enregistre un point de terminaison pour chaque nouvel utilisateur, et il met à jour les points de terminaison pour renvoyer les utilisateurs. Pour enregistrer des points de terminaison depuis un mobile ou un JavaScript client, voirEnregistrez les points de terminaison Amazon Pinpoint dans votre application.

Exemples

Les exemples suivants vous indiquent comment ajouter un point de terminaison à un projet Amazon Pinpoint. Le point de terminaison représente un membre du public qui vit à Seattle et utilise un iPhone. Cette personne peut recevoir un message via le service de notification push Apple (APNs). L'adresse du point de terminaison est le jeton de l'appareil fourni par APNs.

AWS CLI

Vous pouvez utiliser Amazon Pinpoint en exécutant des commandes avec l' AWS CLI.

Example Commande de mise à jour du point de terminaison

Pour ajouter ou mettre à jour un point de terminaison, utilisez la commande update-endpoint :

```
$ aws pinpoint update-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id \
> --endpoint-request file://endpoint-request-file.json
```

Où :

- application-id correspond à l'ID du projet Amazon Pinpoint dans lequel vous ajoutez ou mettez à jour un point de terminaison.
- example-endpoint correspond à l'ID que vous attribuez à un nouveau point de terminaison, ou à l'ID d'un point de terminaison existant que vous mettez à jour.
- endpoint-request-file.json est le chemin d'accès à un fichier JSON local contenant l'entrée du `--endpoint-request` paramètre.

Example Fichier de demande de points de terminaison

L'exemple de commande `update-endpoint` utilise un fichier JSON comme argument pour le paramètre `--endpoint-request`. Ce fichier contient une définition de point de terminaison comme suit :

```
{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

Pour connaître les attributs que vous pouvez utiliser pour définir un point de terminaison, consultez le EndpointRequestschéma dans le manuel Amazon Pinpoint API Reference.

AWS SDK pour Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK pour Java.

Example Code

Pour ajouter un point de terminaison, initialisez un objet `EndpointRequest` et transmettez-le à la méthode `updateEndpoint` du client `AmazonPinpoint` :

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.*;
import java.util.Arrays;

public class AddExampleEndpoint {

 public static void main(String[] args) {

  final String USAGE = "\n" +
    "AddExampleEndpoint - Adds an example endpoint to an Amazon Pinpoint
 application." +
    "Usage: AddExampleEndpoint <applicationId>" +
    "Where:\n" +
    "  applicationId - The ID of the Amazon Pinpoint application to add the example
 " +
    "endpoint to.";

  if (args.length < 1) {
   System.out.println(USAGE);
   System.exit(1);
  }

  String applicationId = args[0];

  // The device token assigned to the user's device by Apple Push Notification
  // service (APNs).
  String deviceToken =
 "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f";

  // Initializes an endpoint definition with channel type and address.
  EndpointRequest wangXiulansIphoneEndpoint = new EndpointRequest()
```

```
    .withChannelType(ChannelType.APNS)
    .withAddress(deviceToken);

  // Adds custom attributes to the endpoint.
  wangXiulansIphoneEndpoint.addAttributesEntry("interests", Arrays.asList(
    "technology",
    "music",
    "travel"));

  // Adds custom metrics to the endpoint.
  wangXiulansIphoneEndpoint.addMetricsEntry("technology_interest_level", 9.0);
  wangXiulansIphoneEndpoint.addMetricsEntry("music_interest_level", 6.0);
  wangXiulansIphoneEndpoint.addMetricsEntry("travel_interest_level", 4.0);

  // Adds standard demographic attributes.
  wangXiulansIphoneEndpoint.setDemographic(new EndpointDemographic()
    .withAppVersion("1.0")
    .withMake("apple")
    .withModel("iPhone")
    .withModelVersion("8")
    .withPlatform("ios")
    .withPlatformVersion("11.3.1")
    .withTimezone("America/Los_Angeles"));

  // Adds standard location attributes.
  wangXiulansIphoneEndpoint.setLocation(new EndpointLocation()
    .withCountry("US")
    .withCity("Seattle")
    .withPostalCode("98121")
    .withLatitude(47.61)
    .withLongitude(-122.33));

  // Initializes the Amazon Pinpoint client.
  AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
    .withRegion(Regions.US_EAST_1).build();

  // Updates or creates the endpoint with Amazon Pinpoint.
  UpdateEndpointResult result = pinpointClient.updateEndpoint(new
UpdateEndpointRequest()
    .withApplicationId(applicationId)
    .withEndpointId("example_endpoint")
    .withEndpointRequest(wangXiulansIphoneEndpoint));
```

```
  System.out.format("Update endpoint result: %s\n",
 result.getMessageBody().getMessage());

 }
}
```

HTTP

Vous pouvez utiliser Amazon Pinpoint en effectuant des demandes HTTP directement à l'API REST.

Example Demande de point de terminaison PUT

Pour ajouter un point de terminaison, envoyez une demande PUT à la ressource Point de terminaison à l'URI suivante :

`/v1/apps/application-id/endpoints/endpoint-id`

Où :

- application-id correspond à l'ID du projet Amazon Pinpoint dans lequel vous ajoutez ou mettez à jour un point de terminaison.
- endpoint-id correspond à l'ID que vous attribuez à un nouveau point de terminaison, ou à l'ID d'un point de terminaison existant que vous mettez à jour.

Dans votre demande, incluez les en-têtes requis et fournissez le EndpointRequestJSON comme corps :

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180428T004705Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180428/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

```
{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

Si votre demande aboutit, vous recevez une réponse similaire à ce qui suit :

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

Informations connexes

Pour plus d'informations sur la ressource Point de terminaison dans l'API Amazon Pinpoint, notamment les méthodes HTTP prises en charge et les paramètres de demande, consultez la section Point de terminaison dans la Référence de l'API Amazon Pinpoint.

Pour plus d'informations sur la personnalisation des messages avec des variables, consultez Variables de message dans le Guide de l'utilisateur Amazon Pinpoint.

Pour plus d'informations sur les quotas qui s'appliquent aux points de terminaison, par exemple le nombre d'attributs que vous pouvez affecter, consultez the section called “Quotas de points de terminaison”.

Associer des utilisateurs à des points de terminaison Amazon Pinpoint

Un point de terminaison peut inclure des attributs qui définissent un utilisateur, c'est-à-dire une personne de votre public. Par exemple, un utilisateur peut représenter une personne qui a installé votre application mobile, ou une personne disposant d'un compte sur votre site web.

Vous définissez un utilisateur en spécifiant un ID utilisateur unique et, éventuellement, des attributs d'utilisateur personnalisés. Si une personne utilise votre application sur plusieurs appareils, ou si cette personne peut recevoir un message à plusieurs adresses, vous pouvez attribuer le même ID d'utilisateur à plusieurs points de terminaison. Dans ce cas, Amazon Pinpoint synchronise les attributs utilisateur entre les points de terminaison. Par conséquent, si vous ajoutez un attribut utilisateur à un point de terminaison, Amazon Pinpoint ajoute cet attribut pour chaque point de terminaison qui inclut le même ID d'utilisateur.

Vous pouvez ajouter des attributs utilisateur pour suivre les données qui s'appliquent à une personne et ne varient pas en fonction de l'appareil que celle-ci utilise. Par exemple, vous pouvez ajouter des attributs pour le nom, l'âge ou le statut du compte d'une personne.

> **Tip**
>
> Si votre application utilise des groupes d'utilisateurs Amazon Cognito pour gérer l'authentification des utilisateurs, Amazon Cognito peut ajouter automatiquement des IDs utilisateurs et des attributs à vos points de terminaison. Pour la valeur d'ID utilisateur du point de terminaison, Amazon Cognito attribue la valeur `sub` affectée à l'utilisateur dans le groupe

d'utilisateurs. Pour en savoir plus sur l'ajout d'utilisateurs avec Amazon Cognito, consultez Utilisation d'Amazon Pinpoint Analytics avec des groupes d'utilisateurs Amazon Cognito dans le Guide du développeur d'Amazon Cognito.

Une fois que vous avez ajouté des définitions utilisateur à vos points de terminaison, vous disposez d'autres options pour segmenter votre public. Vous pouvez définir un segment en fonction des attributs utilisateur ou vous pouvez définir un segment en important une liste d'utilisateurs IDs. Lorsque vous envoyez un message à un segment basé sur des utilisateurs, les destinations potentielles incluent chaque point de terminaison associé à chaque utilisateur du segment.

Vous disposez également d'autres options pour envoyer des messages à votre public. Vous pouvez utiliser une campagne pour envoyer un message à un segment d'utilisateurs, ou vous pouvez envoyer un message directement à une liste d'utilisateurs IDs. Pour personnaliser votre message, vous pouvez inclure des variables de message remplacées par des valeurs d'attributs utilisateur.

Exemples

Les exemples suivants vous indiquent comment ajouter une définition d'utilisateur à un point de terminaison.

AWS CLI

Vous pouvez utiliser Amazon Pinpoint en exécutant des commandes avec l' AWS CLI.

Example Commande de mise à jour du point de terminaison

Pour ajouter un utilisateur à un point de terminaison, utilisez la commande update-endpoint. Pour le paramètre `--endpoint-request`, vous pouvez définir un nouveau point de terminaison, qui peut inclure un utilisateur. Sinon, pour mettre à jour un point de terminaison existant, vous pouvez fournir uniquement les attributs que vous voulez modifier. L'exemple suivant ajoute un utilisateur à un point de terminaison existant en fournissant uniquement les attributs utilisateur :

```
$ aws pinpoint update-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id \
> --endpoint-request file://endpoint-request-file.json
```

Où :

- *application-id*est l'ID du projet Amazon Pinpoint dans lequel vous ajoutez ou mettez à jour un point de terminaison.
- *endpoint-id*est l'ID que vous attribuez à un nouveau point de terminaison ou l'ID d'un point de terminaison existant que vous mettez à jour.
- *endpoint-request-file.json*est le chemin d'accès à un fichier JSON local contenant l'entrée du `--endpoint-request` paramètre.

Example Fichier de demande de points de terminaison

L'exemple de commande `update-endpoint` utilise un fichier JSON comme argument pour le paramètre `--endpoint-request`. Ce fichier contient une définition d'utilisateur comme suit :

```
{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":["Wang"],
            "LastName":["Xiulan"],
            "Gender":["Female"],
            "Age":["39"]
        }
    }
}
```

Pour connaître les attributs que vous pouvez utiliser pour définir un utilisateur, consultez l'`User`objet dans le EndpointRequestschéma du manuel Amazon Pinpoint API Reference.

AWS SDK pour Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK pour Java.

Example Code

Pour ajouter un utilisateur à un point de terminaison, initialisez un EndpointRequest objet et transmettez-le à la `updateEndpoint` méthode du `AmazonPinpoint` client. Vous pouvez utiliser cet objet pour définir un nouveau point de terminaison, qui peut inclure un utilisateur. Sinon, pour mettre à jour un point de terminaison existant, vous pouvez mettre à jour uniquement les propriétés que vous voulez modifier. L'exemple suivant ajoute un utilisateur à un point de terminaison existant en y ajoutant un EndpointUser EndpointRequest objet :

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
   public static void updatePinpointEndpoint(PinpointClient pinpoint, String
 applicationId, String endPointId) {
       try {
           List<String> wangXiList = new ArrayList<>();
           wangXiList.add("cooking");
           wangXiList.add("running");
           wangXiList.add("swimming");

           Map myMapWang = new HashMap<>();
           myMapWang.put("interests", wangXiList);

           List<String> myNameWang = new ArrayList<>();
           myNameWang.add("Wang ");
           myNameWang.add("Xiulan");

           Map wangName = new HashMap<>();
           wangName.put("name", myNameWang);

           EndpointUser wangMajor = EndpointUser.builder()
                   .userId("example_user_10")
                   .userAttributes(wangName)
                   .build();

           // Create an EndpointBatchItem object for Mary Major.
           EndpointRequest wangXiulanEndpoint = EndpointRequest.builder()
                   .channelType(ChannelType.EMAIL)
                   .address("wang_xiulan@example.com")
                   .attributes(myMapWang)
                   .user(wangMajor)
                   .build();
```

```
            // Adds multiple endpoint definitions to a single request object.
            UpdateEndpointRequest endpointList = UpdateEndpointRequest.builder()
                    .applicationId(applicationId)
                    .endpointRequest(wangXiulanEndpoint)
                    .endpointId(endPointId)
                    .build();

            UpdateEndpointResponse result = pinpoint.updateEndpoint(endpointList);
            System.out.format("Update endpoint result: %s\n",
result.messageBody().message());

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

Pour l'exemple complet du SDK, consultez AddExampleUser.java on. GitHub

HTTP

Vous pouvez utiliser Amazon Pinpoint en effectuant des demandes HTTP directement à l'API REST.

Example Demande de point de terminaison PUT avec une définition d'utilisateur

Pour ajouter un utilisateur à un point de terminaison, envoyez une demande PUT à la ressource Point de terminaison à l'URI suivante :

`/v1/apps/`*`application-id`*`/endpoints/`*`endpoint-id`*

Où :

- *`application-id`*est l'ID du projet Amazon Pinpoint dans lequel vous ajoutez ou mettez à jour un point de terminaison.
- *`endpoint-id`*est l'ID que vous attribuez à un nouveau point de terminaison ou l'ID d'un point de terminaison existant que vous mettez à jour.

Dans votre demande, incluez les en-têtes requis et fournissez le EndpointRequestJSON comme corps. Le corps de la demande peut définir un nouveau point de terminaison, qui peut inclure un utilisateur. Sinon, pour mettre à jour un point de terminaison existant, vous pouvez fournir

uniquement les attributs que vous voulez modifier. L'exemple suivant ajoute un utilisateur à un point de terminaison existant en fournissant uniquement les attributs utilisateur :

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":"Wang",
            "LastName":"Xiulan",
            "Gender":"Female",
            "Age":"39"
        }
    }
}
```

Si la demande aboutit, vous recevez une réponse similaire à ce qui suit :

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

Informations connexes

Pour plus d'informations sur la ressource Point de terminaison dans l'API Amazon Pinpoint, notamment les méthodes HTTP prises en charge et les paramètres de demande, consultez la section Point de terminaison dans la Référence de l'API Amazon Pinpoint.

Pour plus d'informations sur la personnalisation des messages avec des variables, consultez Variables de message dans le Guide de l'utilisateur Amazon Pinpoint.

Pour savoir comment définir un segment en important une liste d'utilisateurs IDs, consultez la section Importation de segments dans le guide de l'utilisateur Amazon Pinpoint.

Pour plus d'informations sur l'envoi d'un message direct à un maximum de 100 utilisateurs IDs, consultez la section Messages des utilisateurs dans le manuel Amazon Pinpoint API Reference.

Pour plus d'informations sur les quotas qui s'appliquent aux points de terminaison, y compris le nombre d'attributs utilisateur que vous pouvez attribuer, consultez the section called “Quotas de points de terminaison”.

Ajouter un lot de points de terminaison à Amazon Pinpoint

Vous pouvez ajouter ou mettre à jour plusieurs points de terminaison simultanément en fournissant les points de terminaison par lots. Chaque demande de traitement par lots peut inclure jusqu'à 100 définitions de point de terminaison.

Si vous souhaitez ajouter ou mettre à jour plus de 100 points de terminaison simultanément, consultez Importer des points de terminaison dans Amazon Pinpoint à la place.

Exemples

Les exemples suivants vous indiquent comment ajouter deux points de terminaison simultanément en incluant les points de terminaison dans une demande de traitement par lots.

AWS CLI

Vous pouvez utiliser Amazon Pinpoint en exécutant des commandes avec l' AWS CLI.

Example Commande de mise à jour des points de terminaison par lots

Pour envoyer une demande de traitement par lots de points de terminaison, utilisez la commande `update-endpoints-batch` :

```
$ aws pinpoint update-endpoints-batch \
> --application-id application-id \
> --endpoint-batch-request file://endpoint_batch_request_file.json
```

Où :

- *application-id*est l'ID du projet Amazon Pinpoint dans lequel vous ajoutez ou mettez à jour les points de terminaison.

- *endpoint_batch_request_file.json*est le chemin d'accès à un fichier JSON local contenant l'entrée du `--endpoint-batch-request` paramètre.

Example Fichier de demande de lot de points de terminaison

L'exemple de commande `update-endpoints-batch` utilise un fichier JSON comme argument pour le paramètre `--endpoint-request`. Ce fichier contient un lot de définitions de points de terminaison comme suit :

```
{
    "Item": [
        {
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
            },
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
```

```
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
            },
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
                }
            }
        }
    ]
}
```

Pour connaître les attributs que vous pouvez utiliser pour définir un lot de points de terminaison, consultez le EndpointBatchRequestschéma dans le manuel Amazon Pinpoint API Reference.

AWS SDK pour Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK pour Java.

Example Code

Pour envoyer une demande de traitement par lots de points de terminaison, initialisez un objet `EndpointBatchRequest` et transmettez-le à la méthode `updateEndpointsBatch` du client `AmazonPinpoint`. L'exemple suivant remplit un `EndpointBatchRequest` objet avec deux objets `EndpointBatchItem` :

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
```

```
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class AddExampleEndpoints {

        public static void main(String[] args) {
                final String usage = """

                                Usage:    <appId>

                                Where:
                                   appId - The ID of the application.

                                """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }
```

```
                String applicationId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                updateEndpointsViaBatch(pinpoint, applicationId);
                pinpoint.close();
        }

        public static void updateEndpointsViaBatch(PinpointClient pinpoint, String
applicationId) {
                try {
                        List<String> myList = new ArrayList<>();
                        myList.add("music");
                        myList.add("books");

                        Map myMap = new HashMap<String, List>();
                        myMap.put("attributes", myList);

                        List<String> myNames = new ArrayList<String>();
                        myList.add("Richard");
                        myList.add("Roe");

                        Map myMap2 = new HashMap<String, List>();
                        myMap2.put("name", myNames);

                        EndpointUser richardRoe = EndpointUser.builder()
                                        .userId("example_user_1")
                                        .userAttributes(myMap2)
                                        .build();

                        // Create an EndpointBatchItem object for Richard Roe.
                        EndpointBatchItem richardRoesEmailEndpoint =
EndpointBatchItem.builder()
                                        .channelType(ChannelType.EMAIL)
                                        .address("richard_roe@example.com")
                                        .id("example_endpoint_1")
                                        .attributes(myMap)
                                        .user(richardRoe)
                                        .build();

                        List<String> myListMary = new ArrayList<String>();
                        myListMary.add("cooking");
```

```
                        myListMary.add("politics");
                        myListMary.add("finance");

                        Map myMapMary = new HashMap<String, List>();
                        myMapMary.put("interests", myListMary);

                        List<String> myNameMary = new ArrayList<String>();
                        myNameMary.add("Mary ");
                        myNameMary.add("Major");

                        Map maryName = new HashMap<String, List>();
                        myMapMary.put("name", myNameMary);

                        EndpointUser maryMajor = EndpointUser.builder()
                                        .userId("example_user_2")
                                        .userAttributes(maryName)
                                        .build();

                        // Create an EndpointBatchItem object for Mary Major.
                        EndpointBatchItem maryMajorsSmsEndpoint =
EndpointBatchItem.builder()
                                        .channelType(ChannelType.SMS)
                                        .address("+16145550100")
                                        .id("example_endpoint_2")
                                        .attributes(myMapMary)
                                        .user(maryMajor)
                                        .build();

                        // Adds multiple endpoint definitions to a single request
object.
                        EndpointBatchRequest endpointList =
EndpointBatchRequest.builder()
                                        .item(richardRoesEmailEndpoint)
                                        .item(maryMajorsSmsEndpoint)
                                        .build();

                        // Create the UpdateEndpointsBatchRequest.
                        UpdateEndpointsBatchRequest batchRequest =
UpdateEndpointsBatchRequest.builder()
                                        .applicationId(applicationId)
                                        .endpointBatchRequest(endpointList)
                                        .build();

                        // Updates the endpoints with Amazon Pinpoint.
```

```
                        UpdateEndpointsBatchResponse result =
 pinpoint.updateEndpointsBatch(batchRequest);
                        System.out.format("Update endpoints batch result: %s\n",
 result.messageBody().message());

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

Pour l'exemple complet du SDK, consultez AddExampleEndpoints.java on. GitHub

HTTP

Vous pouvez utiliser Amazon Pinpoint en effectuant des demandes HTTP directement à l'API REST.

Example Demande de points de terminaison PUT

Pour envoyer une demande de lot de points de terminaison, envoyez une demande PUT à la ressource Points de terminaison à l'URI suivante :

`/v1/apps/`*`application-id`*`/endpoints`

Où se *`application-id`* trouve l'ID du projet Amazon Pinpoint dans lequel vous ajoutez ou mettez à jour les points de terminaison ?

Dans votre demande, incluez les en-têtes requis et fournissez le EndpointBatchRequestJSON comme corps :

```
PUT /v1/apps/application_id/endpoints HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180501T184948Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
```

```
    "Item": [
        {
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
            },
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
            },
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
```

```
                }
            }
        }
    ]
}
```

Si votre demande aboutit, vous recevez une réponse similaire à ce qui suit :

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

Informations connexes

Pour plus d'informations sur la ressource Point de terminaison dans l'API Amazon Pinpoint, notamment les méthodes HTTP prises en charge et les paramètres de demande, consultez Point de terminaison dans la Référence de l'API Amazon Pinpoint.

Importer des points de terminaison dans Amazon Pinpoint

Vous pouvez ajouter ou mettre à jour des points de terminaison en nombre en les important à partir d'un compartiment Amazon S3. L'importation de points de terminaison est utile si vous disposez d'enregistrements sur votre audience en dehors d'Amazon Pinpoint, et si vous souhaitez ajouter ces informations à un projet Amazon Pinpoint. Dans ce cas, vous devez :

1. Créer des définitions de point de terminaison basées sur les données de votre propre public.
2. Enregistrez ces définitions de point de terminaison dans un ou plusieurs fichiers, et chargez les fichiers dans un compartiment Amazon S3.
3. Ajoutez les points de terminaison à votre projet Amazon Pinpoint en les important à partir du compartiment.

Chaque tâche d'importation peut transférer jusqu'à 1 Go de données. Lors d'une tâche courante, au cours de laquelle chaque point de terminaison a une taille inférieure ou égale à 4 Ko, vous pouvez importer environ 250 000 points de terminaison. Vous pouvez exécuter jusqu'à deux tâches d'importation simultanées par AWS compte. Si vous avez besoin de plus de bande passante pour

vos tâches d'importation, vous pouvez envoyer une demande d'augmentation du quota de service à Support. Pour de plus amples informations, veuillez consulter Demande d'augmentation de quota.

Avant de commencer

Avant de pouvoir importer des points de terminaison, vous devez disposer des ressources suivantes dans votre compte AWS :

- Un compartiment Amazon S3. Pour créer un compartiment, consultez la section Créer un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service.
- Rôle Gestion des identités et des accès AWS (IAM) qui accorde à Amazon Pinpoint des autorisations de lecture pour votre compartiment Amazon S3. Pour créer ce rôle, consultez Rôle IAM pour l'importation de points de terminaison ou de segments.

Exemples

Les exemples suivants expliquent comment ajouter des définitions de point de terminaison à votre compartiment Amazon S3, puis comment les importer dans un projet Amazon Pinpoint.

Fichiers avec définitions de point de terminaison

Les fichiers que vous ajoutez à votre compartiment Amazon S3 peuvent contenir des définitions de point de terminaison au format CSV ou JSON délimité par des nouvelles lignes. Pour connaître les attributs que vous pouvez utiliser pour définir vos points de terminaison, consultez le schéma EndpointRequestJSON dans le manuel Amazon Pinpoint API Reference.

CSV

Vous pouvez importer des points de terminaison définis dans un fichier CSV, comme dans l'exemple suivant :

```
ChannelType,Address,Location.Country,Demographic.Platform,Demographic.Make,User.UserId
SMS,12065550182,CN,Android,LG,example-user-id-1
APNS,1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f,US,iOS,Apple,example-user-id-2
EMAIL,john.stiles@example.com,US,iOS,Apple,example-user-id-2
```

La première ligne est l'en-tête, qui contient les attributs de point de terminaison. Spécifiez les attributs imbriqués à l'aide de la notation de points, comme dans `Location.Country`.

Les lignes suivantes définissent les points de terminaison en fournissant les valeurs de chacun des attributs de l'en-tête.

Pour inclure une virgule ou un guillemet double dans une valeur, placez la valeur entre guillemets doubles, comme dans `"aaa,bbb"`.

Les sauts de ligne ne sont pas pris en charge au sein d'une valeur du CSV.

JSON

Vous pouvez importer des points de terminaison définis dans un fichier JSON délimité par des nouvelles lignes, comme dans l'exemple suivant :

```
{"ChannelType":"SMS","Address":"12065550182","Location":
{"Country":"CN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

Dans ce cas, chaque ligne est un objet JSON complet qui contient une définition de point de terminaison distincte.

Demandes de tâche d'importation

Les exemples suivants vous indiquent comment ajouter des définitions de point de terminaison à Amazon S3 en chargeant un fichier local dans un compartiment. Ensuite, les exemples importent les définitions de point de terminaison dans un projet Amazon Pinpoint.

AWS CLI

Vous pouvez utiliser Amazon Pinpoint en exécutant des commandes avec l' AWS CLI.

Example Commande CP S3

Pour charger un fichier local dans un compartiment Amazon S3, utilisez la commande cp d'Amazon S3 :

```
$ aws s3 cp ./endpoints-file s3://bucket-name/prefix/
```

Où :

- ./endpoints-file correspond au chemin d'accès à un fichier local qui contient les définitions de point de terminaison.
- bucket-name/prefix/ correspond au nom de votre compartiment Amazon S3 et, éventuellement, un préfixe vous permettant d'organiser les objets de votre compartiment de manière hiérarchique. Par exemple, un préfixe utile peut être `pinpoint/imports/endpoints/`.

Example Commande de création d'une tâche d'importation

Pour importer des définitions de points de terminaison à partir d'un compartiment Amazon S3, utilisez la commande `create-import-job` :

```
$ aws pinpoint create-import-job \
> --application-id application-id \
> --import-job-request \
> S3Url=s3://bucket-name/prefix/key,\
> RoleArn=iam-import-role-arn,\
> Format=format,\
> RegisterEndpoints=true
```

Où :

- application-id correspond à l'ID du projet Amazon Pinpoint pour lequel vous importez des points de terminaison.
- bucket- name/prefix/key est l'emplacement dans Amazon S3 qui contient un ou plusieurs objets à importer. L'emplacement peut se terminer par la clé d'un objet individuel, ou il peut se terminer par un préfixe qui qualifie plusieurs objets.
- iam-import-role-arnest le nom de ressource Amazon (ARN) d'un rôle IAM qui accorde à Amazon Pinpoint un accès en lecture au bucket.
- format peut être `JSON` ou `CSV`, selon le format utilisé pour définir vos points de terminaison. Si l'emplacement Amazon S3 inclut plusieurs objets de formats mixtes, Amazon Pinpoint importe uniquement les objets correspondant au format spécifié.
- RegisterEndpoints peut avoir la valeur `true` ou `false`. Lorsque le paramètre est défini sur true, la tâche d'importation enregistre les points de terminaison auprès d'Amazon Pinpoint, lorsque les définitions de points de terminaison sont importées.

RegisterEndpoints et DefineSegments combinaisons

RegisterEndpoints	DefineSegments	Description
true	true	Amazon Pinpoint importera les points de terminaison et créera un segment contenant les points de terminaison.
vrai	false	Amazon Pinpoint importera les points de terminaison et ne créera pas de segment.
false	true	Amazon Pinpoint importera les points de terminaison et créera un segment contenant les points de terminaison. Les points de terminaison ne seront pas enregistrés et ne remplaceront pas les points de terminaison existants.
false	false	Amazon Pinpoint rejettera cette demande.

La réponse inclut des détails sur la tâche d'importation :

```
{
    "ImportJobResponse": {
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
```

```
            "S3Url": "s3://bucket-name/prefix/key"
        },
        "Id": "d5ecad8e417d498389e1d5b9454d4e0c",
        "JobStatus": "CREATED",
        "Type": "IMPORT"
    }
}
```

La réponse fournit l'ID de tâche avec l'attribut `Id`. Vous pouvez utiliser cet ID pour vérifier le statut actuel de la tâche d'importation.

Example Commande de génération d'une tâche d'importation

Pour vérifier le statut actuel d'une tâche d'importation, utilisez la commande `get-import-job` :

```
$ aws pinpoint get-import-job \
> --application-id application-id \
> --job-id job-id
```

Où :

- application-id correspond à l'ID du projet Amazon Pinpoint pour lequel la tâche d'importation a été initiée.
- job-id correspond à l'ID de la tâche d'importation que vous vérifiez.

La réponse à cette commande fournit l'état actuel de la tâche d'importation :

```
{
    "ImportJobResponse": {
        "ApplicationId": "application-id",
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-24T21:26:45.308Z",
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
            "S3Url": "s3://s3-bucket-name/prefix/endpoint-definitions.json"
        },
        "FailedPieces": 0,
```

```
        "Id": "job-id",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 3,
        "Type": "IMPORT"
    }
}
```

La réponse fournit le statut de la tâche avec l'attribut `JobStatus`.

AWS SDK pour Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK pour Java.

Example Code

Pour charger un fichier avec des définitions de point de terminaison dans Amazon S3, utilisez la méthode `putObject` du client `AmazonS3`.

Pour importer les points de terminaison dans un projet Amazon Pinpoint, initialisez un objet `CreateImportJobRequest`. Transmettez ensuite cet objet à la méthode `createImportJob` du client `AmazonPinpoint`.

```
package com.amazonaws.examples.pinpoint;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.CreateImportJobRequest;
import com.amazonaws.services.pinpoint.model.CreateImportJobResult;
import com.amazonaws.services.pinpoint.model.Format;
import com.amazonaws.services.pinpoint.model.GetImportJobRequest;
import com.amazonaws.services.pinpoint.model.GetImportJobResult;
import com.amazonaws.services.pinpoint.model.ImportJobRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
```

```
import java.util.List;
import java.util.concurrent.TimeUnit;

public class ImportEndpoints {

    public static void main(String[] args) {

        final String USAGE = "\n" +
                "ImportEndpoints - Adds endpoints to an Amazon Pinpoint application
 by: \n" +
                "1.) Uploading the endpoint definitions to an Amazon S3 bucket. \n"
 +
                "2.) Importing the endpoint definitions from the bucket to an Amazon
 Pinpoint " +
                "application.\n\n" +
                "Usage: ImportEndpoints <endpointsFileLocation> <s3BucketName>
 <iamImportRoleArn> " +
                "<applicationId>\n\n" +
                "Where:\n" +
                "  endpointsFileLocation - The relative location of the JSON file
 that contains the " +
                "endpoint definitions.\n" +
                "  s3BucketName - The name of the Amazon S3 bucket to upload the
 JSON file to. If the " +
                "bucket doesn't exist, a new bucket is created.\n" +
                "  iamImportRoleArn - The ARN of an IAM role that grants Amazon
 Pinpoint read " +
                "permissions to the S3 bucket.\n" +
                "  applicationId - The ID of the Amazon Pinpoint application to add
 the endpoints to.";

        if (args.length < 1) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String endpointsFileLocation = args[0];
        String s3BucketName = args[1];
        String iamImportRoleArn = args[2];
        String applicationId = args[3];

        Path endpointsFilePath = Paths.get(endpointsFileLocation);
        File endpointsFile = new
 File(endpointsFilePath.toAbsolutePath().toString());
```

```
        uploadToS3(endpointsFile, s3BucketName);

        importToPinpoint(endpointsFile.getName(), s3BucketName, iamImportRoleArn,
applicationId);

    }

    private static void uploadToS3(File endpointsFile, String s3BucketName) {

        // Initializes Amazon S3 client.
        final AmazonS3 s3 = AmazonS3ClientBuilder.defaultClient();

        // Checks whether the specified bucket exists. If not, attempts to create
one.
        if (!s3.doesBucketExistV2(s3BucketName)) {
            try {
                s3.createBucket(s3BucketName);
                System.out.format("Created S3 bucket %s.\n", s3BucketName);
            } catch (AmazonS3Exception e) {
                System.err.println(e.getErrorMessage());
                System.exit(1);
            }
        }

        // Uploads the endpoints file to the bucket.
        String endpointsFileName = endpointsFile.getName();
        System.out.format("Uploading %s to S3 bucket %s . . .\n", endpointsFileName,
s3BucketName);
        try {
            s3.putObject(s3BucketName, "imports/" + endpointsFileName,
endpointsFile);
            System.out.println("Finished uploading to S3.");
        } catch (AmazonServiceException e) {
            System.err.println(e.getErrorMessage());
            System.exit(1);
        }
    }

    private static void importToPinpoint(String endpointsFileName, String
s3BucketName,
            String iamImportRoleArn, String applicationId) {

        // The S3 URL that Amazon Pinpoint requires to find the endpoints file.
        String s3Url = "s3://" + s3BucketName + "/imports/" + endpointsFileName;
```

```
        // Defines the import job that Amazon Pinpoint runs.
        ImportJobRequest importJobRequest = new ImportJobRequest()
                .withS3Url(s3Url)
                .withRegisterEndpoints(true)
                .withRoleArn(iamImportRoleArn)
                .withFormat(Format.JSON);
        CreateImportJobRequest createImportJobRequest = new CreateImportJobRequest()
                .withApplicationId(applicationId)
                .withImportJobRequest(importJobRequest);

        // Initializes the Amazon Pinpoint client.
        AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
                .withRegion(Regions.US_EAST_1).build();

        System.out.format("Importing endpoints in %s to Amazon Pinpoint application
%s . . .\n",
                endpointsFileName, applicationId);

        try {

            // Runs the import job with Amazon Pinpoint.
            CreateImportJobResult importResult =
pinpointClient.createImportJob(createImportJobRequest);

            String jobId = importResult.getImportJobResponse().getId();
            GetImportJobResult getImportJobResult = null;
            String jobStatus = null;

            // Checks the job status until the job completes or fails.
            do {
                getImportJobResult = pinpointClient.getImportJob(new
GetImportJobRequest()
                        .withJobId(jobId)
                        .withApplicationId(applicationId));
                jobStatus =
getImportJobResult.getImportJobResponse().getJobStatus();
                System.out.format("Import job %s . . .\n", jobStatus.toLowerCase());
                TimeUnit.SECONDS.sleep(3);
            } while (!jobStatus.equals("COMPLETED") && !jobStatus.equals("FAILED"));

            if (jobStatus.equals("COMPLETED")) {
                System.out.println("Finished importing endpoints.");
            } else {
```

```
                System.err.println("Failed to import endpoints.");
                System.exit(1);
            }

            // Checks for entries that failed to import.
            // getFailures provides up to 100 of the first failed entries for the
 job, if
            // any exist.
            List<String> failedEndpoints =
 getImportJobResult.getImportJobResponse().getFailures();
            if (failedEndpoints != null) {
                System.out.println("Failed to import the following entries:");
                for (String failedEndpoint : failedEndpoints) {
                    System.out.println(failedEndpoint);
                }
            }

        } catch (AmazonServiceException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }

    }

}
```

HTTP

Vous pouvez utiliser Amazon Pinpoint en effectuant des demandes HTTP directement à l'API REST.

Example Demande PUT Object S3

Pour ajouter vos définitions de point de terminaison à un compartiment, utilisez l'opération Amazon S3 PUT Object et fournissez les définitions de point de terminaison en tant que corps :

```
PUT /prefix/key HTTP/1.1
Content-Type: text/plain
Accept: application/json
Host: bucket-name.s3.amazonaws.com
X-Amz-Content-Sha256:
 c430dc094b0cec2905bc88d96314914d058534b14e2bc6107faa9daa12fdff2d
X-Amz-Date: 20180605T184132Z
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/s3/aws4_request, SignedHeaders=accept;cache-control;content-
length;content-type;host;postman-token;x-amz-content-sha256;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{"ChannelType":"SMS","Address":"2065550182","Location":
{"Country":"CAN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"USA"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"USA"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

Où :

- /prefix/key correspond au préfixe et au nom de clé de l'objet qui contient les définitions de point de terminaison après le chargement. Vous pouvez utiliser le préfixe pour organiser vos objets de manière hiérarchique. Par exemple, un préfixe utile peut être `pinpoint/imports/endpoints/`.
- bucket-name correspond au nom du compartiment Amazon S3 auquel vous ajoutez les définitions de point de terminaison.

Example Demande de tâche d'importation POST

Pour importer des définitions de points de terminaison à partir d'un compartiment Amazon S3, envoyez une demande POST à la ressource Tâches d'importation. Dans votre demande, incluez les en-têtes requis et fournissez le ImportJobRequestJSON comme corps :

```
POST /v1/apps/application_id/jobs/import HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T214912Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

```
{
  "S3Url": "s3://bucket-name/prefix/key",
  "RoleArn": "iam-import-role-arn",
  "Format": "format",
  "RegisterEndpoints": true
}
```

Où :

- application-id correspond à l'ID du projet Amazon Pinpoint pour lequel vous importez des points de terminaison.
- bucket- name/prefix/key est l'emplacement dans Amazon S3 qui contient un ou plusieurs objets à importer. L'emplacement peut se terminer par la clé d'un objet individuel, ou il peut se terminer par un préfixe qui qualifie plusieurs objets.
- iam-import-role-arnest le nom de ressource Amazon (ARN) d'un rôle IAM qui accorde à Amazon Pinpoint un accès en lecture au bucket.
- format peut être `JSON` ou `CSV`, selon le format utilisé pour définir vos points de terminaison. Si l'emplacement Amazon S3 inclut plusieurs fichiers de formats mixtes, Amazon Pinpoint importe uniquement les fichiers correspondant au format spécifié.

Si votre demande aboutit, vous recevez une réponse similaire à ce qui suit :

```
{
    "Id": "a995ce5d70fa44adb563b7d0e3f6c6f5",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-05T21:49:15.288Z",
    "Type": "IMPORT",
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key",
        "RoleArn": "iam-import-role-arn",
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

La réponse fournit l'ID de tâche avec l'attribut `Id`. Vous pouvez utiliser cet ID pour vérifier le statut actuel de la tâche d'importation.

Example Demande de tâche d'importation GET

Pour vérifier le statut actuel d'une tâche d'importation, envoyez une demande `GET` à la ressource Tâche d'importation :

```
GET /v1/apps/application_id/jobs/import/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T220744Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-
type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

Où :

- application-id correspond à l'ID du projet Amazon Pinpoint pour lequel la tâche d'importation a été initiée.
- job-id correspond à l'ID de la tâche d'importation que vous vérifiez.

Si votre demande aboutit, vous recevez une réponse similaire à ce qui suit :

```
{
    "ApplicationId": "application_id",
    "Id": "70a51b2cf442447492d2c8e50336a9e8",
    "JobStatus": "COMPLETED",
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-05T22:04:49.213Z",
    "CompletionDate": "2018-06-05T22:04:58.034Z",
    "Type": "IMPORT",
    "TotalFailures": 0,
    "TotalProcessed": 3,
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key.json",
        "RoleArn": "iam-import-role-arn",
```

```
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

La réponse fournit le statut de la tâche avec l'attribut `JobStatus`.

Informations connexes

Pour plus d'informations concernant la ressource Tâches d'importation dans l'API Amazon Pinpoint, notamment les méthodes HTTP prises en charge et les paramètres de demande, consultez Tâches d'importation dans la Référence de l'API Amazon Pinpoint.

Exporter des points de terminaison depuis Amazon Pinpoint vers des compartiments Amazon S3

Pour obtenir toutes les informations dont dispose Amazon Pinpoint sur votre audience, vous pouvez exporter les définitions de points de terminaison appartenant à un projet. Lorsque vous effectuez l'exportation, Amazon Pinpoint place les définitions de points de terminaison dans un compartiment Amazon S3 que vous spécifiez. L'exportation de points de terminaison est utile si vous souhaitez :

- Afficher les données les plus récentes sur les points de terminaison nouveaux et existants que votre application client a enregistré auprès d'Amazon Pinpoint.
- Synchroniser les données de point de terminaison dans Amazon Pinpoint avec votre propre système de gestion des relations client (CRM).
- Créer des rapports ou analyser vos données client.

> **Note**
>
> Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Si vous devez supprimer des données de point de terminaison que vous avez exportées vers un compartiment Amazon S3, vous devez le faire dans Amazon S3. Pour plus d'informations sur la suppression de données sensibles, consultez Comment vider un compartiment S3 ? ou Comment supprimer un compartiment S3 ? .

Avant de commencer

Avant de pouvoir exporter des points de terminaison, vous devez disposer des ressources suivantes dans votre compte AWS :

- Un compartiment Amazon S3. Pour créer un compartiment, consultez la section Créer un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service.
- Rôle Gestion des identités et des accès AWS (IAM) qui accorde à Amazon Pinpoint des autorisations d'écriture pour votre compartiment Amazon S3. Pour créer ce rôle, consultez Rôle IAM pour l'exportation de points de terminaison ou de segments.

Exemples

Les exemples suivants expliquent comment exporter des points de terminaison à partir d'un projet Amazon Pinpoint, puis les télécharger à partir de votre compartiment Amazon S3.

AWS CLI

Vous pouvez utiliser Amazon Pinpoint en exécutant des commandes avec l' AWS CLI.

Example Commande de création d'une tâche d'exportation

Pour exporter les points de terminaison dans votre projet Amazon Pinpoint, utilisez la commande `create-export-job` :

```
$ aws pinpoint create-export-job \
> --application-id application-id \
> --export-job-request \
> S3UrlPrefix=s3://bucket-name/prefix/,\
> RoleArn=iam-export-role-arn
```

Où :

- *`application-id`* correspond à l'ID du projet Amazon Pinpoint contenant les points de terminaison.
- *`bucket-name/prefix/`* correspond au nom de votre compartiment Amazon S3 et, éventuellement, à un préfixe vous permettant d'organiser les objets de votre compartiment de manière hiérarchique. Par exemple, un préfixe utile peut être `pinpoint/exports/endpoints/`.

- *iam-export-role-arn* correspond à l'Amazon Resource Name (ARN) d'un rôle IAM qui accorde à Amazon Pinpoint un accès en écriture au compartiment.

La réponse à cette commande fournit des détails sur la tâche d'exportation :

```
{
    "ExportJobResponse": {
        "CreationDate": "2018-06-04T22:04:20.585Z",
        "Definition": {
            "RoleArn": "iam-export-role-arn",
            "S3UrlPrefix": "s3://s3-bucket-name/prefix/"
        },
        "Id": "7390e0de8e0b462380603c5a4df90bc4",
        "JobStatus": "CREATED",
        "Type": "EXPORT"
    }
}
```

La réponse fournit l'ID de tâche avec l'attribut `Id`. Vous pouvez utiliser cet ID pour vérifier le statut actuel de la tâche d'exportation.

Example Commande de génération d'une tâche d'exportation

Pour vérifier le statut actuel d'une tâche d'exportation, utilisez la commande `get-export-job` :

```
$ aws pinpoint get-export-job \
> --application-id application-id \
> --job-id job-id
```

Où :

- *application-id* correspond à l'ID du projet Amazon Pinpoint à partir duquel les points de terminaison ont été exportés.
- *job-id* correspond à l'ID de la tâche que vous vérifiez.

La réponse à cette commande fournit l'état actuel de la tâche d'exportation :

```
{
    "ExportJobResponse": {
        "ApplicationId": "application-id",
```

```
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-08T22:16:48.228Z",
        "CreationDate": "2018-05-08T22:16:44.812Z",
        "Definition": {},
        "FailedPieces": 0,
        "Id": "6c99c463f14f49caa87fa27a5798bef9",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 215,
        "Type": "EXPORT"
    }
}
```

La réponse fournit le statut de la tâche avec l'attribut `JobStatus`. Lorsque la valeur du statut de la tâche est `COMPLETED`, vous pouvez obtenir vos points de terminaison exportés dans votre compartiment Amazon S3.

Example Commande CP S3

Pour télécharger vos points de terminaison exportés, utilisez la commande cp Amazon S3 :

```
$ aws s3 cp s3://bucket-name/prefix/key.gz /local/directory/
```

Où :

- *`bucket-name/prefix/key`* correspond à l'emplacement du fichier .gz que Amazon Pinpoint a ajouté à votre compartiment lorsque vous avez exporté vos points de terminaison. Ce fichier contient les définitions de points de terminaison exportées. Par exemple, dans l'URL `https://PINPOINT-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/Exports/example.csv`, « `PINPOINT-EXAMPLE-BUCKET` » est le nom du compartiment et « `Exports/example.csv` » est la clé. Pour plus d'informations sur les Clés, consultez Clés dans le Guide de l'utilisateur Amazon S3.
- */local/directory/* correspond au chemin d'accès au répertoire local dans lequel vous souhaitez télécharger les points de terminaison.

AWS SDK pour Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK pour Java.

Example Code

Pour exporter les points de terminaison à partir d'un projet Amazon Pinpoint, initialisez un objet `CreateExportJobRequest`. Transmettez ensuite cet objet à la méthode `createExportJob` du client `AmazonPinpoint`.

Pour télécharger les points de terminaison exportés à partir d'Amazon Pinpoint, utilisez la méthode `getObject` du client `AmazonS3`.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;
```

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
```

```
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-
export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = """

                This program performs the following steps:

                1. Exports the endpoints to an Amazon S3 bucket.
                2. Downloads the exported endpoints files from Amazon S3.
                3. Parses the endpoints files to obtain the endpoint IDs and prints
 them.
```

```
                Usage: ExportEndpoints <applicationId> <s3BucketName>
<iamExportRoleArn> <path>

                Where:
                  applicationId - The ID of the Amazon Pinpoint application that has
the endpoint.
                  s3BucketName - The name of the Amazon S3 bucket to export the JSON
file to.\s
                  iamExportRoleArn - The ARN of an IAM role that grants Amazon
Pinpoint write permissions to the S3 bucket.  path - The path where the files
downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
                """;

       if (args.length != 4) {
           System.out.println(usage);
           System.exit(1);
       }

       String applicationId = args[0];
       String s3BucketName = args[1];
       String iamExportRoleArn = args[2];
       String path = args[3];
       System.out.println("Deleting an application with ID: " + applicationId);

       Region region = Region.US_EAST_1;
       PinpointClient pinpoint = PinpointClient.builder()
               .region(region)
               .build();

       S3Client s3Client = S3Client.builder()
               .region(region)
               .build();

       exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
       pinpoint.close();
       s3Client.close();
   }

   public static void exportAllEndpoints(PinpointClient pinpoint,
           S3Client s3Client,
           String applicationId,
           String s3BucketName,
           String path,
```

```
            String iamExportRoleArn) {

        try {
            List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
 s3BucketName, iamExportRoleArn,
                    applicationId);
            List<String> endpointFileKeys = objectKeys.stream().filter(o ->
 o.endsWith(".gz"))
                    .collect(Collectors.toList());
            downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static List<String> exportEndpointsToS3(PinpointClient pinpoint, S3Client
 s3Client, String s3BucketName,
            String iamExportRoleArn, String applicationId) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
        String endpointsKeyPrefix = "exports/" + applicationId + "_" +
 dateFormat.format(new Date());
        String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
 "/";
        List<String> objectKeys = new ArrayList<>();
        String key;

        try {
            // Defines the export job that Amazon Pinpoint runs.
            ExportJobRequest jobRequest = ExportJobRequest.builder()
                    .roleArn(iamExportRoleArn)
                    .s3UrlPrefix(s3UrlPrefix)
                    .build();

            CreateExportJobRequest exportJobRequest =
 CreateExportJobRequest.builder()
                    .applicationId(applicationId)
                    .exportJobRequest(jobRequest)
                    .build();
```

```
            System.out.format("Exporting endpoints from Amazon Pinpoint application
%s to Amazon S3 " +
                    "bucket %s . . .\n", applicationId, s3BucketName);

            CreateExportJobResponse exportResult =
pinpoint.createExportJob(exportJobRequest);
            String jobId = exportResult.exportJobResponse().id();
            System.out.println(jobId);
            printExportJobStatus(pinpoint, applicationId, jobId);

            ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                    .bucket(s3BucketName)
                    .prefix(endpointsKeyPrefix)
                    .build();

            // Create a list of object keys.
            ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
            List<S3Object> objects = v2Response.contents();
            for (S3Object object : objects) {
                key = object.key();
                objectKeys.add(key);
            }

            return objectKeys;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void printExportJobStatus(PinpointClient pinpointClient,
            String applicationId,
            String jobId) {

        GetExportJobResponse getExportJobResult;
        String status;

        try {
            // Checks the job status until the job completes or fails.
            GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
                    .jobId(jobId)
                    .applicationId(applicationId)
```

```
                    .build();

            do {
                getExportJobResult = pinpointClient.getExportJob(exportJobRequest);
                status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
                System.out.format("Export job %s . . .\n", status);
                TimeUnit.SECONDS.sleep(3);

            } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

            if (status.equals("COMPLETED")) {
                System.out.println("Finished exporting endpoints.");
            } else {
                System.err.println("Failed to export endpoints.");
                System.exit(1);
            }

        } catch (PinpointException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Download files from an Amazon S3 bucket and write them to the path location.
    public static void downloadFromS3(S3Client s3Client, String path, String
s3BucketName, List<String> objectKeys) {

        String newPath;
        try {
            for (String key : objectKeys) {
                GetObjectRequest objectRequest = GetObjectRequest.builder()
                        .bucket(s3BucketName)
                        .key(key)
                        .build();

                ResponseBytes<GetObjectResponse> objectBytes =
s3Client.getObjectAsBytes(objectRequest);
                byte[] data = objectBytes.asByteArray();

                // Write the data to a local file.
                String fileSuffix = new
SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
                newPath = path + fileSuffix + ".gz";
```

```
                File myFile = new File(newPath);
                OutputStream os = new FileOutputStream(myFile);
                os.write(data);
            }
            System.out.println("Download finished.");

        } catch (S3Exception | NullPointerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

Pour l'exemple complet du SDK, consultez ExportEndpoints.java on. GitHub

HTTP

Vous pouvez utiliser Amazon Pinpoint en effectuant des demandes HTTP directement à l'API REST.

Example Demande de tâche d'exportation POST

Pour exporter les points de terminaison dans votre projet Amazon Pinpoint, envoyez une demande POST à la ressource Tâches d'exportation :

```
POST /v1/apps/application_id/jobs/export HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T001238Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3UrlPrefix": "s3://bucket-name/prefix",
  "RoleArn": "iam-export-role-arn"
}
```

Où :

- *application-id* correspond à l'ID du projet Amazon Pinpoint contenant les points de terminaison.
- *bucket-name/prefix* correspond au nom de votre compartiment Amazon S3 et, éventuellement, à un préfixe vous permettant d'organiser les objets de votre compartiment de manière hiérarchique. Par exemple, un préfixe utile peut être `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* correspond à l'Amazon Resource Name (ARN) d'un rôle IAM qui accorde à Amazon Pinpoint un accès en écriture au compartiment.

La réponse à cette demande fournit des détails sur la tâche d'exportation :

```
{
    "Id": "611bdc54c75244bfa51fe7001ddb2e36",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-06T00:12:43.271Z",
    "Type": "EXPORT",
    "Definition": {
        "S3UrlPrefix": "s3://bucket-name/prefix",
        "RoleArn": "iam-export-role-arn"
    }
}
```

La réponse fournit l'ID de tâche avec l'attribut `Id`. Vous pouvez utiliser cet ID pour vérifier le statut actuel de la tâche d'exportation.

Example Demande de tâche d'exportation GET

Pour vérifier le statut actuel d'une tâche d'exportation, envoyez une demande `GET` à la ressource Tâche d'exportation :

```
GET /v1/apps/application_id/jobs/export/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T002443Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-
type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

Où :

- *application-id* correspond à l'ID du projet Amazon Pinpoint à partir duquel les points de terminaison ont été exportés.
- *job-id* correspond à l'ID de la tâche que vous vérifiez.

La réponse à cette demande fournit l'état actuel de la tâche d'exportation :

```
{
    "ApplicationId": "application_id",
    "Id": "job_id",
    "JobStatus": "COMPLETED",
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-06T00:12:43.271Z",
    "CompletionDate": "2018-06-06T00:13:01.141Z",
    "Type": "EXPORT",
    "TotalFailures": 0,
    "TotalProcessed": 217,
    "Definition": {}
}
```

La réponse fournit le statut de la tâche avec l'attribut `JobStatus`. Lorsque la valeur du statut de la tâche est `COMPLETED`, vous pouvez obtenir vos points de terminaison exportés dans votre compartiment Amazon S3.

Informations connexes

Pour trouver l'ID de point de terminaison d'un point de terminaison spécifique, vous devez déterminer à quel segment appartient le point de terminaison, puis exporter le segment depuis Amazon Pinpoint. Les données exportées englobent l'ID de chaque point de terminaison. Vous pouvez exporter un segment vers un fichier à l'aide de la console Amazon Pinpoint. Pour plus d'informations sur l'exportation de segments, consultez Exporter des segments dans le guide de l'utilisateur Amazon Pinpoint.

Pour plus d'informations concernant la ressource Tâches d'exportation dans l'API Amazon Pinpoint, notamment les méthodes HTTP prises en charge et les paramètres de demande, consultez Tâches d'exportation dans la Référence de l'API Amazon Pinpoint.

Rechercher des points de terminaison dans un projet Amazon Pinpoint

Vous pouvez rechercher les détails de n'importe quel point de terminaison individuel ajouté à un projet Amazon Pinpoint. Ces détails peuvent inclure l'adresse de destination de vos messages, le canal de messagerie, des données relatives à l'appareil de l'utilisateur, des données relatives à l'emplacement de l'utilisateur, et des attributs personnalisés enregistrés dans vos points de terminaison.

Pour rechercher un point de terminaison, vous avez besoin de l'ID du point de terminaison. Si vous ne connaissez pas l'ID, vous pouvez obtenir les données du point de terminaison en exportant celui-ci à la place. Pour exporter des points de terminaison, consultez the section called “Exporter des points de terminaison depuis Amazon Pinpoint vers des compartiments Amazon S3”.

Exemples

Les exemples suivants vous indiquent comment rechercher un point de terminaison individuel en spécifiant son ID.

AWS CLI

Vous pouvez utiliser Amazon Pinpoint en exécutant des commandes avec l' AWS CLI.

Example Commande de recherche d'un point de terminaison

Pour rechercher un point de terminaison, utilisez la commande `get-endpoint` :

```
$ aws pinpoint get-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id
```

Où :

- *`application-id`* correspond à l'ID du projet Amazon Pinpoint contenant le point de terminaison.
- *`endpoint-id`* correspond à l'ID du point de terminaison que vous recherchez.

La réponse à cette commande est la définition JSON du point de terminaison, comme dans l'exemple suivant :

```
{
    "EndpointResponse": {
        "Address":
 "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
        "ApplicationId": "application-id",
        "Attributes": {
            "Interests": [
                "Technology",
                "Music",
                "Travel"
            ]
        },
        "ChannelType": "APNS",
        "CohortId": "63",
        "CreationDate": "2018-05-01T17:31:01.046Z",
        "Demographic": {
            "AppVersion": "1.0",
            "Make": "apple",
            "Model": "iPhone",
            "ModelVersion": "8",
            "Platform": "ios",
            "PlatformVersion": "11.3.1",
            "Timezone": "America/Los_Angeles"
        },
        "EffectiveDate": "2018-05-07T19:03:29.963Z",
        "EndpointStatus": "ACTIVE",
        "Id": "example_endpoint",
        "Location": {
            "City": "Seattle",
            "Country": "US",
            "Latitude": 47.6,
            "Longitude": -122.3,
            "PostalCode": "98121"
        },
        "Metrics": {
            "music_interest_level": 6.0,
            "travel_interest_level": 4.0,
            "technology_interest_level": 9.0
        },
        "OptOut": "ALL",
        "RequestId": "7f546cac-6858-11e8-adcd-2b5a07aab338",
        "User": {
            "UserAttributes": {
```

```
                "Gender": "Female",
                "FirstName": "Wang",
                "LastName": "Xiulan",
                "Age": "39"
            },
            "UserId": "example_user"
        }
    }
}
```

AWS SDK pour Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK pour Java.

Example Code

Pour rechercher un point de terminaison, initialisez un objet `GetEndpointRequest`. Transmettez ensuite cet objet à la méthode `getEndpoint` du client `AmazonPinpoint` :

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
```

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <endpoint>

                Where:
                  appId - The ID of the application to delete.
                  endpoint - The ID of the endpoint.\s
                  """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpoint = args[1];
        System.out.println("Looking up an endpoint point with ID: " + endpoint);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        lookupPinpointEndpoint(pinpoint, appId, endpoint);
        pinpoint.close();
    }

    public static void lookupPinpointEndpoint(PinpointClient pinpoint, String appId,
 String endpoint) {
        try {
            GetEndpointRequest appRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpoint)
                    .build();

            GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
            EndpointResponse endResponse = result.endpointResponse();

            // Uses the Google Gson library to pretty print the endpoint JSON.
```

```
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

            String endpointJson = gson.toJson(endResponse);
            System.out.println(endpointJson);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

Pour imprimer les données de point de terminaison dans un format accessible en lecture, cet exemple utilise la bibliothèque Google GSON pour convertir l'objet `EndpointResponse` en chaîne JSON.

HTTP

Vous pouvez utiliser Amazon Pinpoint en effectuant des demandes HTTP directement à l'API REST.

Example Demande de point de terminaison GET

Pour rechercher un point de terminaison, envoyez une demande `GET` à la ressource Endpoint :

```
GET /v1/apps/application_id/endpoints/endpoint_id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

Où :

- *`application-id`* correspond à l'ID du projet Amazon Pinpoint contenant le point de terminaison.
- *`endpoint-id`* correspond à l'ID du point de terminaison que vous recherchez.

La réponse à cette demande est la définition JSON du point de terminaison, comme dans l'exemple suivant :


```
{
    "ChannelType": "APNS",
    "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
    "EndpointStatus": "ACTIVE",
    "OptOut": "NONE",
    "RequestId": "b720cfa8-6924-11e8-aeda-0b22e0b0fa59",
    "Location": {
        "Latitude": 47.6,
        "Longitude": -122.3,
        "PostalCode": "98121",
        "City": "Seattle",
        "Country": "US"
    },
    "Demographic": {
        "Make": "apple",
        "Model": "iPhone",
        "ModelVersion": "8",
        "Timezone": "America/Los_Angeles",
        "AppVersion": "1.0",
        "Platform": "ios",
        "PlatformVersion": "11.3.1"
    },
    "EffectiveDate": "2018-06-06T00:58:19.865Z",
    "Attributes": {
        "Interests": [
            "Technology",
            "Music",
            "Travel"
        ]
    },
    "Metrics": {
        "music_interest_level": 6,
        "travel_interest_level": 4,
        "technology_interest_level": 9
    },
    "User": {},
    "ApplicationId": "application_id",
    "Id": "example_endpoint",
    "CohortId": "39",
    "CreationDate": "2018-06-06T00:58:19.865Z"
```

```
}
```

Informations connexes

Pour plus d'informations sur la ressource Point de terminaison dans l'API Amazon Pinpoint, consultez la section Endpoint dans la Référence de l'API Amazon Pinpoint.

Répertorier les points IDs de terminaison avec Amazon Pinpoint

Pour mettre à jour ou supprimer un point de terminaison, vous avez besoin de l'ID du point de terminaison. Ainsi, si vous souhaitez effectuer ces opérations sur tous les points de terminaison d'un projet Amazon Pinpoint, la première étape consiste à répertorier tous les points de IDs terminaison appartenant à ce projet. Vous pouvez ensuite effectuer une itération sur ceux-ci IDs pour, par exemple, ajouter un attribut globalement ou supprimer tous les points de terminaison de votre projet.

L'exemple suivant utilise AWS SDK pour Java et effectue les opérations suivantes :

1. Appelle la `exportEndpointsToS3` méthode d'exemple à partir de l'exemple de code dans Exporter les points de terminaison depuis Amazon Pinpoint. Cette méthode exporte les définitions de points de terminaison à partir d'un projet Amazon Pinpoint. Les définitions de points de terminaison sont ajoutées en tant que fichiers gzip dans un compartiment Amazon S3.
2. Télécharge les fichiers gzip exportés.
3. Lit les fichiers gzip et obtient l'ID de point de terminaison de chaque définition JSON de point de terminaison.
4. Imprime le point IDs de terminaison sur la console.
5. Nettoie en supprimant les fichiers qu'Amazon Pinpoint a ajoutés à Amazon S3.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;
```

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListEndpointIds {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <applicationId> <userId>

                Where:
                   applicationId - The ID of the Amazon Pinpoint application that has
 the endpoint.
                   userId - The user id applicable to the endpoints""";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
        String userId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllEndpoints(pinpoint, applicationId, userId);
        pinpoint.close();
    }

    public static void listAllEndpoints(PinpointClient pinpoint,
            String applicationId,
            String userId) {
```

```
        try {
            GetUserEndpointsRequest endpointsRequest =
 GetUserEndpointsRequest.builder()
                    .userId(userId)
                    .applicationId(applicationId)
                    .build();

            GetUserEndpointsResponse response =
 pinpoint.getUserEndpoints(endpointsRequest);
            List<EndpointResponse> endpoints = response.endpointsResponse().item();

            // Display the results.
            for (EndpointResponse endpoint : endpoints) {
                System.out.println("The channel type is: " + endpoint.channelType());
                System.out.println("The address is  " + endpoint.address());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Pour l'exemple complet du SDK, consultez ListEndpointIs.java on. GitHub

Gérez le nombre maximum de points de terminaison dans Amazon Pinpoint

Chaque membre de votre audience peut avoir un maximum de 15 points de terminaison associés à son UserId, voirQuotas de points de terminaison. Si vous essayez d'ajouter un 16e point de terminaison, selon le cas ChannelType, vous obtiendrez BadRequestExceptionou vous réussirez en supprimant le point de terminaison le plus ancien EffectiveDate.

Ajouter un 16e point de terminaison

- Si le nouveau type de canal pour le point de terminaison est SMS, PUSH, VOICE, EMAIL, CUSTOM ou IN_APP, il BadRequestExceptionest renvoyé car le membre du public a atteint son nombre maximal de points de terminaison. Vous devez supprimer un point de terminaison associé

au membre de l'audience et réessayer. Consultez Supprimer des points de terminaison d'Amazon Pinpoint par programmation.

- Si le nouveau type de canal pour le point de terminaison est ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX ou BAIDU :
  - Vérifiez qu'au moins un point de terminaison actuellement associé au membre de l'audience possède un ADM, un GCM, un ChannelTypeAPNS, un APNS_VOICE, un APNS_VOIP_SANDBOX ou un BAIDU. Si ce n'est pas le cas, il BadRequestExceptionest renvoyé et un point de terminaison doit être supprimé avant de réessayer, voirSupprimer des points de terminaison d'Amazon Pinpoint par programmation.
  - Sinon, le point de terminaison le plus ancien EffectiveDateest défini comme ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX `INACTIVE` ou BAIDU. ChannelType
    - Le point UserIdde terminaison de l'ancien point de terminaison est supprimé.
    - Le nouveau point de terminaison est associé au membre de l'audience et celui-ci possède toujours le nombre maximum de points de terminaison.

Le point de terminaison peut être réactivé en définissant le statut sur `ACTIVE` et en le UserIdréajoutant au point de terminaison.

Supprimer des points de terminaison d'Amazon Pinpoint par programmation

Un point de terminaison représente une seule méthode de contact d'un de vos clients. Chaque point de terminaison peut faire référence à l'adresse e-mail d'un client, à l'identifiant d'un appareil mobile, au numéro de téléphone ou à tout autre type de destination auquel vous pouvez envoyer des messages. Dans de nombreux juridictions, ce type de données peut être considéré comme des renseignements personnels. Vous pouvez supprimer des points de terminaison lorsque vous ne souhaitez plus envoyer de messages à une destination spécifique, par exemple lorsque la destination devient inaccessible, ou lorsqu'un client clôt un compte.

Exemples

Les exemples suivants vous indiquent comment supprimer un point de terminaison.

AWS CLI

Vous pouvez utiliser Amazon Pinpoint en exécutant des commandes avec l' AWS CLI.

Example Commande de suppression d'un point de terminaison

Pour supprimer un point de terminaison, utilisez la commande `delete-endpoint` :

```
$ aws pinpoint delete-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id
```

Où :

- application-id correspond à l'ID du projet Amazon Pinpoint contenant le point de terminaison.
- endpoint-id correspond à l'ID du point de terminaison que vous supprimez.

La réponse à cette commande est la définition JSON du point de terminaison que vous avez supprimé.

AWS SDK pour Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK pour Java.

Example Code

Pour supprimer un point de terminaison, utilisez la méthode `deleteEndpoint` du client `AmazonPinpoint`. Fournissez un objet `DeleteEndpointRequest` comme argument de la méthode :

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
```

```
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appName> <endpointId >

                Where:
                  appId - The id of the application to delete.
                  endpointId - The id of the endpoint to delete.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinEncpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEncpoint(PinpointClient pinpoint, String appId,
 String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id  " + id);
```

```
        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

Pour l'exemple complet du SDK, consultez DeleteEndpoint.java on. GitHub

HTTP

Vous pouvez utiliser Amazon Pinpoint en effectuant des demandes HTTP directement à l'API REST.

Example Demande de point de terminaison DELETE

Pour supprimer un point de terminaison, envoyez une demande `DELETE` à la ressource Endpoint :

```
DELETE /v1/apps/application-id/endpoints/endpoint-id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

Où :

- application-id correspond à l'ID du projet Amazon Pinpoint contenant le point de terminaison.
- endpoint-id correspond à l'ID du point de terminaison que vous supprimez.

La réponse à cette demande est la définition JSON du point de terminaison que vous avez supprimé.

Création ou importation de segments dans Amazon Pinpoint

Un segment d'utilisateurs représente un sous-ensemble d'utilisateurs basé sur des caractéristiques communes, telles que la date à laquelle les utilisateurs ont récemment utilisé votre application ou la plate-forme de l'appareil qu'ils utilisent. Un segment identifie les utilisateurs qui reçoivent les messages envoyés par une campagne. Définissez les segments de façon à atteindre le public approprié lorsque vous voulez l'inviter à réutiliser votre application, proposer des offres spéciales ou encore accroître l'implication des utilisateurs et leurs achats.

Une fois que vous avez créé un segment, vous pouvez l'utiliser dans une ou plusieurs campagnes. Une campagne transmet des messages sur mesure aux utilisateurs du segment.

Pour plus d'informations, consultez Segments.

Rubriques

- Créez des segments dans Amazon Pinpoint
- Importer des segments dans Amazon Pinpoint
- Personnalisez les segments Amazon Pinpoint à l'aide d'une fonction AWS Lambda

Créez des segments dans Amazon Pinpoint

Pour atteindre le public visé pour une campagne, construisez un segment basé sur les données rapportées par votre application. Par exemple, pour atteindre les utilisateurs n'ayant pas utilisé votre application récemment, vous pouvez définir un segment pour ceux qui n'ont pas utilisé votre application au cours des 30 derniers jours.

Pour plus d'exemples de code, consultez la section Exemples de code.

Créez des segments à l'aide du AWS SDK pour Java

L'exemple suivant montre comment créer un segment avec le kit AWS SDK pour Java. L'exemple crée un segment d'utilisateurs dont l'équipe est constituée `Lakers` et a été active au cours des 30 derniers jours. Une fois le segment créé, vous pouvez l'utiliser dans le cadre d'une campagne ou d'un parcours. Pour un exemple d'utilisation d'un segment avec une campagne, voirCréez des campagnes Amazon Pinpoint de manière programmatique.

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateSegment {
        public static void main(String[] args) {
```

```
                final String usage = """

                                Usage:   <appId>

                                Where:
                                  appId - The application ID to create a segment for.

                                """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String appId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                SegmentResponse result = createSegment(pinpoint, appId);
                System.out.println("Segment " + result.name() + " created.");
                System.out.println(result.segmentType());
                pinpoint.close();
        }

        public static SegmentResponse createSegment(PinpointClient client, String
appId) {
                try {
                        Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();
                        segmentAttributes.put("Team", AttributeDimension.builder()
                                        .attributeType(AttributeType.INCLUSIVE)
                                        .values("Lakers")
                                        .build());

                        RecencyDimension recencyDimension = RecencyDimension.builder()
                                        .duration("DAY_30")
                                        .recencyType("ACTIVE")
                                        .build();

                        SegmentBehaviors segmentBehaviors = SegmentBehaviors.builder()
                                        .recency(recencyDimension)
                                        .build();
```

```
                        SegmentDemographics segmentDemographics = SegmentDemographics
                                        .builder()
                                        .build();

                        SegmentLocation segmentLocation = SegmentLocation
                                        .builder()
                                        .build();

                        SegmentDimensions dimensions = SegmentDimensions
                                        .builder()
                                        .attributes(segmentAttributes)
                                        .behavior(segmentBehaviors)
                                        .demographic(segmentDemographics)
                                        .location(segmentLocation)
                                        .build();

                        WriteSegmentRequest writeSegmentRequest =
 WriteSegmentRequest.builder()
                                        .name("MySegment")
                                        .dimensions(dimensions)
                                        .build();

                        CreateSegmentRequest createSegmentRequest =
 CreateSegmentRequest.builder()
                                        .applicationId(appId)
                                        .writeSegmentRequest(writeSegmentRequest)
                                        .build();

                        CreateSegmentResponse createSegmentResult =
 client.createSegment(createSegmentRequest);
                        System.out.println("Segment ID: " +
 createSegmentResult.segmentResponse().id());
                        System.out.println("Done");
                        return createSegmentResult.segmentResponse();

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
                return null;
        }
}
```

Lorsque vous exécutez cet exemple, les éléments suivants s'affichent dans la fenêtre de la console de votre IDE :

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

Pour l'exemple complet du SDK, consultez CreateSegment.java on. GitHub

Importer des segments dans Amazon Pinpoint

Avec Amazon Pinpoint, vous pouvez définir un segment d'utilisateurs en important les informations sur les points de terminaison qui appartiennent au segment. Un point de terminaison est une destination de messagerie unique, comme un jeton d'appareil mobile en mode push, un numéro de téléphone portable ou une adresse e-mail.

L'importation de segments est utile si vous avez déjà créé des segments de vos utilisateurs en dehors d'Amazon Pinpoint, mais que vous souhaitez impliquer vos utilisateurs dans les campagnes Amazon Pinpoint.

Lorsque vous importez un segment, Amazon Pinpoint récupère les points de terminaison du segment à partir d'Amazon Simple Storage Service (Amazon S3). Avant l'importation, vous ajoutez les points de terminaison à Amazon S3 et vous créez un rôle IAM qui accorde à Amazon Pinpoint l'accès à Amazon S3. Ensuite, vous fournissez à Amazon Pinpoint l'emplacement Amazon S3 où les points de terminaison sont stockés et Amazon Pinpoint ajoute chaque point de terminaison au segment.

Pour créer le rôle IAM, consultez Rôle IAM pour l'importation de points de terminaison ou de segments. Pour plus d'informations sur l'importation d'un segment à l'aide de la console Amazon Pinpoint, consultez Importation de segments dans le Guide de l'utilisateur Amazon Pinpoint.

Pour plus d'exemples de code, consultez la section Exemples de code.

Importez un segment avec le AWS SDK pour Java

L'exemple suivant montre comment importer un segment à l'aide du kit AWS SDK pour Java.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
```

```
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <bucket> <key> <roleArn>\s

                Where:
                  appId - The application ID to create a segment for.
                  bucket - The name of the Amazon S3 bucket that contains the segment
 definitons.
                  key - The key of the S3 object.
                  roleArn - ARN of the role that allows Amazon Pinpoint to access S3.
 You need to set trust management for this to work. See https://docs.aws.amazon.com/
IAM/latest/UserGuide/reference_policies_elements_principal.html
                  """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
```

```
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket, key,
roleArn);
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for import
job status.");
        System.out.println("See application " + response.jobStatus() + " for import job
status.");
        pinpoint.close();
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
            String appId,
            String bucket,
            String key,
            String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                    .defineSegment(true)
                    .registerEndpoints(true)
                    .roleArn(roleArn)
                    .format(Format.JSON)
                    .s3Url("s3://" + bucket + "/" + key)
                    .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                    .importJobRequest(importRequest)
                    .applicationId(appId)
                    .build();

            CreateImportJobResponse jobResponse = client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
        return null;
    }
}
```

Pour l'exemple complet du SDK, consultez ImportingSegments.java on. GitHub

Personnalisez les segments Amazon Pinpoint à l'aide d'une fonction AWS Lambda

> Ceci est une documentation préliminaire pour une fonctionnalité en version bêta publique. Elle est susceptible d’être modifiée.

Vous pouvez l'utiliser AWS Lambda pour personnaliser la manière dont une campagne Amazon Pinpoint engage votre public cible. Vous pouvez ainsi modifier le segment de la campagne au moment où Amazon Pinpoint envoie le message de la campagne. AWS Lambda

AWS Lambda est un service de calcul que vous pouvez utiliser pour exécuter du code sans provisionner ni gérer de serveurs. Vous pouvez intégrer votre code et le charger sur Lambda en tant que fonctions Lambda. Lambda exécute une fonction lorsqu'elle est appelée, ce que vous pouvez faire manuellement, ou automatiquement en réponse à des événements. Pour plus d’informations, consultez le Guide du développeur AWS Lambda.

Pour attribuer une fonction Lambda à une campagne, vous devez définir les paramètres `CampaignHook` de la campagne à l'aide de la ressource Campaign de l'API Amazon Pinpoint. Ces paramètres incluent le nom de la fonction Lambda. Sont également définis le mode `CampaignHook`, qui détermine si Amazon Pinpoint reçoit une valeur renvoyée par la fonction.

Une fonction Lambda attribuée à une campagne est appelée extension Amazon Pinpoint.

Une fois les paramètres `CampaignHook` définis, Amazon Pinpoint appelle automatiquement la fonction Lambda lorsqu'il exécute la campagne, avant de diffuser le message de la campagne. Lorsqu'il invoque la fonction, Amazon Pinpoint fournit des données d'événement sur la diffusion du message. Ces données incluent le segment de la campagne, qui répertorie les points de terminaison auxquels Amazon Pinpoint envoie le message.

Si le mode `CampaignHook` est défini sur `FILTER`, Amazon Pinpoint autorise la fonction à modifier et à renvoyer le segment avant d'envoyer le message. Par exemple, la fonction peut mettre à jour les

définitions des points de terminaison avec des attributs qui contiennent des données d'une source externe vers Amazon Pinpoint. La fonction peut également filtrer le segment en supprimant certains points de terminaison, selon les conditions de votre code de fonction. Après avoir reçu le segment modifié de votre fonction, Amazon Pinpoint envoie le message à chacun des points de terminaison du segment en utilisant le canal de livraison de la campagne.

En traitant vos segments avec AWS Lambda, vous pouvez mieux contrôler à qui vous envoyez des messages et leur contenu. Vous pouvez adapter vos campagnes en temps réel, au moment de l'envoi des messages de la campagne. Le filtrage des segments vous permet de définir plus précisément les sous-ensembles de vos segments. L'ajout ou la mise à jour des attributs de points de terminaison permet également de rendre de nouvelles données disponibles pour les variables de messages.

> **Note**
>
> Vous pouvez également utiliser les paramètres `CampaignHook` pour attribuer une fonction Amazon Pinpoint qui gère la diffusion du message. Ce type de fonction est utile pour diffuser des messages par le biais de canaux personnalisés qu'Amazon Pinpoint ne prend pas en charge, comme les plateformes de réseaux sociaux. Pour de plus amples informations, veuillez consulter Créez une chaîne personnalisée dans Amazon Pinpoint à l'aide d'un webhook ou d'une fonction Lambda.
> Lorsque vous invoquez un hook Lambda à l’aide d’Amazon Pinpoint, la fonction Lambda doit également se trouver dans la même région que le projet Amazon Pinpoint.

Pour modifier les segments de campagne avec AWS Lambda, créez d'abord une fonction qui traite les données d'événement envoyées par Amazon Pinpoint et renvoie un segment modifié. Ensuite, autorisez Amazon Pinpoint à invoquer la fonction en affectant une stratégie de fonction Lambda. Pour finir, attribuez la fonction à une ou plusieurs campagnes en définissant les paramètres `CampaignHook`.

Pour plus d'exemples de code, consultez la section Exemples de code.

Données d’événement

Quand Amazon Pinpoint appelle votre fonction Lambda, la charge applicative suivante est fournie en tant que données d'événement :

```
{
```

```
  "MessageConfiguration": {Message configuration}
  "ApplicationId": ApplicationId,
  "CampaignId": CampaignId,
  "TreatmentId": TreatmentId,
  "ActivityId": ActivityId,
  "ScheduledTime": Scheduled Time,
  "Endpoints": {
    EndpointId: {Endpoint definition}
    . . .
  }
}
```

AWS Lambda transmet les données de l'événement à votre code de fonction. Les données d'événements fournissent les attributs suivants :

- `MessageConfiguration` : possède la même structure que l'objet `DirectMessageConfiguration` de la ressource Messages de l'API Amazon Pinpoint.
- `ApplicationId` : ID du projet Amazon Pinpoint auquel la campagne appartient.
- `CampaignId` : ID de la campagne Amazon Pinpoint pour laquelle la fonction est invoquée.
- `TreatmentId`— L'ID d'une variante de campagne utilisée à des fins de A/B test.
- `ActivityId` : ID de l'activité effectuée par la campagne.
- `ScheduledTime` : date et heure de diffusion des messages de la campagne au format ISO 8601.
- `Endpoints`— Une carte qui associe un point de terminaison à IDs des définitions de point de terminaison. Chaque charge applicative de données d'événement contient un maximum de 50 points de terminaison. Si le segment de campagne contient plus de 50 points de terminaison, Amazon Pinpoint invoque la fonction à plusieurs reprises, avec un maximum de 50 points de terminaison à la fois, jusqu'à ce que tous les points de terminaison aient été traités.

Création d'une fonction Lambda

Pour savoir comment créer une fonction Lambda, consultez Démarrage dans le Guide du développeur AWS Lambda . Lorsque vous créez votre fonction, n'oubliez pas que la diffusion des messages échoue dans les conditions suivantes :

- La fonction Lambda a besoin de plus de 15 secondes pour renvoyer le segment modifié.
- Amazon Pinpoint ne peut pas décoder la valeur de retour de la fonction.

- La fonction a besoin de plus de trois tentatives d'Amazon Pinpoint pour invoquer la valeur avec succès.

Amazon Pinpoint accepte uniquement les définitions des points de terminaison dans la valeur de retour de la fonction. La fonction ne peut pas modifier d'autres éléments dans les données d'événement.

Exemple de fonction Lambda

Votre fonction Lambda traite les données d'événement envoyées par Amazon Pinpoint et renvoie les points de terminaison modifiés comme indiqué dans l'exemple suivant écrit en Node.js :

```
'use strict';

exports.handler = (event, context, callback) => {
    for (var key in event.Endpoints) {
        if (event.Endpoints.hasOwnProperty(key)) {
            var endpoint = event.Endpoints[key];
            var attr = endpoint.Attributes;
            if (!attr) {
                attr = {};
                endpoint.Attributes = attr;
            }
            attr["CreditScore"] = [ Math.floor(Math.random() * 200) + 650];
        }
    }
    console.log("Received event:", JSON.stringify(event, null, 2));
    callback(null, event.Endpoints);
};
```

Lambda transmet les données d'événement au gestionnaire en tant que paramètre `event`.

Dans cet exemple, le gestionnaire parcourt chaque point de terminaison de l’objet `event.Endpoints` et y ajoute un nouvel attribut, `CreditScore`. La valeur de l’attribut `CreditScore` est simplement un nombre aléatoire.

La `console.log()` déclaration enregistre l'événement dans CloudWatch Logs.

L'instruction `callback()` renvoie les points de terminaison modifiés à Amazon Pinpoint. Généralement, le paramètre `callback` est facultatif dans les fonctions Lambda écrites en Node.js,

mais il est requis dans ce contexte, car la fonction doit renvoyer les points de terminaison mis à jour à Amazon Pinpoint.

Votre fonction doit renvoyer les points de terminaison dans le même format que celui fourni par les données d'événement, à savoir une carte qui associe le point de terminaison IDs aux définitions de points de terminaison, comme dans l'exemple suivant :

```
{
    "eqmj8wpxszeqy/b3vch04sn41yw": {
        "ChannelType": "GCM",
        "Address": "4d5e6f1a2b3c4d5e6f7g8h9i0j1a2b3c",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
            "Make": "android"
        },
        "EffectiveDate": "2017-11-02T21:26:48.598Z",
        "User": {}
    },
    "idrexqqtn8sbwfex0ouscod0yto": {
        "ChannelType": "APNS",
        "Address": "1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
            "Make": "apple"
        },
        "EffectiveDate": "2017-11-02T21:26:48.598Z",
        "User": {}
    }
}
```

La fonction de l'exemple modifie et renvoie l'objet `event.Endpoints` qu'elle a reçu dans les données d'événement.

Vous pouvez également inclure les attributs `BodyOverride` et `TitleOverride` dans les définitions des points de terminaison que vous renvoyez.

> **Note**
>
> Lorsque vous utilisez cette solution pour envoyer des messages, Amazon Pinpoint respecte uniquement les attributs `TitleOverride` et `BodyOverride` pour les points de terminaison

où la valeur de l'attribut `ChannelType` correspond à l'un des éléments suivants : `ADM`, `APNS`, `APNS_SANDBOX`, `APNS_VOIP`, `APNS_VOIP_SANDBOX`, `BAIDU`, `GCM` ou `SMS`.
Amazon Pinpoint ne respecte pas ces attributs pour les points de terminaison dont la valeur de l'attribut `ChannelType` est `EMAIL`.

Attribuer une politique de fonction Lambda

Avant de pouvoir utiliser votre fonction Lambda pour traiter vos points de terminaison, vous devez autoriser Amazon Pinpoint à invoquer votre fonction Lambda. Pour accorder l'autorisation d'invocation, attribuez une stratégie de fonction Lambda à la fonction. Une stratégie de fonction Lambda est une stratégie d'autorisations basée sur les ressources qui désigne les entités qui peuvent utiliser votre fonction et les actions que ces entités peuvent effectuer.

Pour plus d'informations, consultez Utilisation des stratégies fondées sur les ressources pour AWS Lambda dans le Guide de l'utilisateur AWS Lambda .

Exemple de stratégie de fonction

La politique suivante autorise le responsable du service Amazon Pinpoint à utiliser l'`lambda:InvokeFunction`action pour une campagne spécifique () *`campaign-id`* :

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:account-id:function:function-name}",
  "Condition": {
    "StringEquals": {
      "AWS:SourceAccount": "111122223333"
    },
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/campaigns/campaign-id"
    }
  }
}
```

Votre stratégie de fonction exige un bloc `Condition` qui inclut une clé `AWS:SourceArn`. Ce code indique la campagne Amazon Pinpoint qui est autorisée à invoquer la fonction. Dans cet exemple, la stratégie accorde l'autorisation à une seule campagne. Le `Condition` bloc doit également inclure une `AWS:SourceAccount` clé qui contrôle quel AWS compte peut invoquer l'action.

Pour écrire une stratégie plus générique, utilisez des caractères génériques (*) correspondant à plusieurs caractères. Par exemple, vous pouvez utiliser le `Condition` bloc suivant pour autoriser n'importe quelle campagne d'un projet Amazon Pinpoint spécifique (*`application-id`*) à invoquer la fonction :

```
...
"Condition": {
  "StringEquals": {
    "AWS:SourceAccount": "111122223333"
  },
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/
campaigns/*"
  }
}
...
```

Si vous souhaitez que la fonction Lambda soit la fonction par défaut utilisée par toutes les campagnes pour un projet, nous vous recommandons de configurer le bloc `Condition` pour la stratégie de la manière précédente. Pour plus d'informations sur la définition d'une fonction Lambda comme fonction par défaut pour toutes les campagnes d'un projet, consultez Attribuer une fonction Lambda à une campagne.

Accorder l'autorisation d'invocation à Amazon Pinpoint

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour ajouter des autorisations à la politique de fonction Lambda attribuée à votre fonction Lambda. Pour autoriser Amazon Pinpoint à invoquer une fonction pour une campagne donnée, utilisez la commande Lambda `add-permission`, comme illustré dans l'exemple suivant :

```
$ aws lambda add-permission \
> --function-name function-name \
> --statement-id sid \
> --action lambda:InvokeFunction \
> --principal pinpoint.us-east-1.amazonaws.com \
```

```
> --source-account 111122223333
> --source-arn arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/
campaigns/campaign-id
```

Vous pouvez consulter votre campagne IDs en utilisant la commande get-campaigns dans le. AWS CLI Vous pouvez également rechercher l'ID de votre application à l'aide de la commande get-apps.

Lorsque vous exécutez la commande `add-permission` de Lambda, celui-ci renvoie le résultat suivant :

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/application-id/
campaigns/campaign-id\"}}
      {\"StringEquals\":
        {\"AWS:SourceAccount\":
          \"111122223333\"}}}
}
```

La valeur `Statement` est une version de la chaîne JSON correspondant à l'instruction ajoutée à la politique de la fonction Lambda.

Attribuer une fonction Lambda à une campagne

Vous pouvez attribuer une fonction Lambda à une campagne Amazon Pinpoint. Vous pouvez également définir la fonction Lambda comme fonction utilisée par défaut par toutes les campagnes d'un projet, excepté pour celles auxquelles vous attribuez une fonction individuellement.

Pour assigner une fonction Lambda à une campagne, utilisez l'API d'Amazon Pinpoint pour créer ou mettre à jour un objet `Campaign` et définir son attribut `CampaignHook`. Pour définir une fonction Lambda par défaut pour toutes les campagnes d'un projet, créez ou mettez à jour la ressource `Settings` de ce projet et définissez son objet `CampaignHook`.

Dans les deux cas, définissez les attributs `CampaignHook` suivants :

- `LambdaFunctionName` : nom ou ARN de la fonction Lambda qu'Amazon Pinpoint invoque avant d'envoyer les messages de la campagne.
- `Mode` – Défini sur `FILTER`. Avec ce mode, Amazon Pinpoint invoque la fonction et attend qu'elle renvoie les points de terminaison modifiés. Après les avoir reçus, Amazon Pinpoint envoie le message. Amazon Pinpoint patiente jusqu'à 15 secondes avant de déclarer l'échec de la distribution du message.

Une fois les paramètres `CampaignHook` de la campagne définis, Amazon Pinpoint invoque la fonction Lambda spécifiée avant d'envoyer les messages de la campagne. Amazon Pinpoint attend de recevoir les points de terminaison modifiés de la part de la fonction. Si Amazon Pinpoint reçoit les points de terminaison mis à jour, il procède à la diffusion de message à l'aide des données des points de terminaison mis à jour.

Créez des campagnes Amazon Pinpoint de manière programmatique

Pour aider à augmenter l'engagement entre votre application et ses utilisateurs, utilisez Amazon Pinpoint pour créer et gérer des campagnes de notification push qui accèdent à des segments d'utilisateurs particuliers.

Par exemple, votre campagne peut inviter les utilisateurs qui n'ont pas exécuté récemment votre application à le faire ou proposer des promotions spéciales pour les utilisateurs qui n'ont pas acheté récemment.

Une campagne envoie un message sur mesure à un segment d'utilisateurs que vous spécifiez. La campagne peut envoyer le message à tous les utilisateurs du segment ou vous pouvez allouer des données d'exclusion, qui correspondent au pourcentage d'utilisateurs qui ne reçoit aucun message.

Vous pouvez définir le calendrier de la campagne de façon à envoyer le message une seule fois ou à une fréquence récurrente, telle qu'une fois par semaine. Pour empêcher que les utilisateurs ne reçoivent le message à des moments inopportuns, le calendrier peut inclure une pause pendant laquelle aucun message n'est envoyé.

Pour tester d'autres stratégies de campagne, configurez votre campagne comme A/B test. Un A/B test inclut au moins deux traitements du message ou du planning. Les traitements sont des variations de votre message ou calendrier. Tandis que vos utilisateurs répondent à la campagne, vous pouvez afficher les analyses de campagne afin de comparer l'efficacité de chaque traitement.

Pour plus d'informations, consultez Campagnes dans le guide de l'API REST Amazon Pinpoint ou Campagnes dans le guide de l'utilisateur Amazon Pinpoint.

Créez une campagne Amazon Pinpoint standard

Une campagne standard envoie une notification push personnalisée à un segment spécifié selon un calendrier que vous définissez. L'exemple suivant montre comment créer une campagne avec le kit AWS SDK pour Java. Pour un exemple de création d'un segment à transmettre, consultezCréez des segments dans Amazon Pinpoint.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
```

```
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = """

                Usage:   <appId> <segmentId>

                Where:
                  appId - The ID of the application to create the campaign in.
                  segmentId - The ID of the segment to create the campaign from.
                """;

        if (args.length != 2) {
```

```
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId, String
segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
    }

    public static CampaignResponse createCampaign(PinpointClient client, String appID,
String segmentID) {

        try {
            Schedule schedule = Schedule.builder()
                    .startTime("IMMEDIATE")
                    .build();

            Message defaultMessage = Message.builder()
                    .action(Action.OPEN_APP)
                    .body("My message body.")
                    .title("My message title.")
                    .build();

            MessageConfiguration messageConfiguration = MessageConfiguration.builder()
                    .defaultMessage(defaultMessage)
                    .build();

            WriteCampaignRequest request = WriteCampaignRequest.builder()
                    .description("My description")
                    .schedule(schedule)
                    .name("MyCampaign")
                    .segmentId(segmentID)
```

```
                    .messageConfiguration(messageConfiguration)
                    .build();

            CreateCampaignResponse result =
 client.createCampaign(CreateCampaignRequest.builder()
                    .applicationId(appID)
                    .writeCampaignRequest(request).build());

            System.out.println("Campaign ID: " + result.campaignResponse().id());
            return result.campaignResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return null;
    }
}
```

Lorsque vous exécutez cet exemple, les éléments suivants s'affichent dans la fenêtre de la console de votre IDE :

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Pour l'exemple complet du SDK, reportez-vous CreateCampaign.javaà la section suivante. GitHub

Créez une campagne A/B de test Amazon Pinpoint à l'aide du AWS SDK pour Java

Une campagne de A/B test se comporte comme une campagne standard, mais vous permet de définir différents traitements pour le message ou le calendrier de la campagne. Un A/B test inclut au moins deux traitements du message ou du planning. Les traitements sont des variations de votre message ou calendrier. Tandis que vos utilisateurs répondent à la campagne, vous pouvez afficher les analyses de campagne afin de comparer l'efficacité de chaque traitement.

L'exemple suivant montre comment créer une campagne de A/B test avec AWS SDK pour Java.

```
import com.amazonaws.services.pinpoint.AmazonPinpointClient;
import com.amazonaws.services.pinpoint.model.Action;
import com.amazonaws.services.pinpoint.model.CampaignResponse;
```

```
import com.amazonaws.services.pinpoint.model.CreateCampaignRequest;
import com.amazonaws.services.pinpoint.model.CreateCampaignResult;
import com.amazonaws.services.pinpoint.model.Message;
import com.amazonaws.services.pinpoint.model.MessageConfiguration;
import com.amazonaws.services.pinpoint.model.Schedule;
import com.amazonaws.services.pinpoint.model.WriteCampaignRequest;
import com.amazonaws.services.pinpoint.model.WriteTreatmentResource;

import java.util.ArrayList;
import java.util.List;

public class PinpointCampaignSample {

    public CampaignResponse createAbCampaign(AmazonPinpointClient client, String appId,
 String segmentId) {
        Schedule schedule = new Schedule()
                .withStartTime("IMMEDIATE");

        // Default treatment.
        Message defaultMessage = new Message()
                .withAction(Action.OPEN_APP)
                .withBody("My message body.")
                .withTitle("My message title.");

        MessageConfiguration messageConfiguration = new MessageConfiguration()
                .withDefaultMessage(defaultMessage);

        // Additional treatments
        WriteTreatmentResource treatmentResource = new WriteTreatmentResource()
                .withMessageConfiguration(messageConfiguration)
                .withSchedule(schedule)
                .withSizePercent(40)
                .withTreatmentDescription("My treatment description.")
                .withTreatmentName("MyTreatment");

        List<WriteTreatmentResource> additionalTreatments = new
 ArrayList<WriteTreatmentResource>();
        additionalTreatments.add(treatmentResource);

        WriteCampaignRequest request = new WriteCampaignRequest()
                .withDescription("My description.")
                .withSchedule(schedule)
                .withSegmentId(segmentId)
                .withName("MyCampaign")
```

```
                .withMessageConfiguration(messageConfiguration)
                .withAdditionalTreatments(additionalTreatments)
                .withHoldoutPercent(10); // Hold out of A/B test

        CreateCampaignRequest createCampaignRequest = new CreateCampaignRequest()
                .withApplicationId(appId).withWriteCampaignRequest(request);

        CreateCampaignResult result = client.createCampaign(createCampaignRequest);

        System.out.println("Campaign ID: " + result.getCampaignResponse().getId());

        return result.getCampaignResponse();
    }

}
```

Lorsque vous exécutez cet exemple, les éléments suivants s'affichent dans la fenêtre de la console de votre IDE :

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Gérer les balises de ressources Amazon Pinpoint

Une balise est une étiquette que vous pouvez éventuellement définir et associer à AWS des ressources, notamment à certains types de ressources Amazon Pinpoint. Les balises peuvent vous aider à classer et à gérer ces types de ressources de différentes façons, notamment par objectif, par propriétaire, par environnement ou selon d'autres critères. Par exemple, vous pouvez utiliser des balises pour appliquer des stratégies ou une automatisation, ou pour identifier les ressources qui sont soumises à certaines exigences en matière de conformité. Vous pouvez ajouter des balises aux types de ressources Amazon Pinpoint suivants :

- Campagnes
- Modèles de messages
- Projets (applications)
- Segments

Une ressource peut avoir jusqu'à 50 balises. Chaque balise est constituée d'une clé de balise obligatoire et d'une valeur de balise facultative que vous définissez. Une clé de balise est une étiquette générale qui fait office de catégorie pour des valeurs de balise plus spécifiques. Une valeur de balise tient lieu de descripteur pour une clé de balise.

Une clé de balise peut contenir jusqu'à 128 caractères. Une valeur de balise peut contenir jusqu'à 256 caractères. Les caractères peuvent être des lettres Unicode, des chiffres un espace ou l'une des symboles suivants : _ . : / = + -. Les restrictions supplémentaires suivantes s'appliquent aux balises :

- Les clés et les valeurs des balises distinguent les majuscules et minuscules.
- Pour chaque ressource associée, chaque clé de balise doit être unique et ne peut avoir qu'une seule valeur.
- Le `aws:` préfixe est réservé à l'usage de AWS ; vous ne pouvez pas l'utiliser dans les clés de balise ou les valeurs que vous définissez. De plus, vous ne pouvez pas modifier ou supprimer des noms ou des valeurs de balise ayant ce préfixe. Les balises qui utilisent ce préfixe ne sont pas comptabilisées dans le quota de 50 balises par ressource.
- Vous ne pouvez pas mettre à jour ou supprimer une ressource uniquement en fonction de ses balises. Vous devez également spécifier l'Amazon Resource Name (ARN) ou l'ID de ressource selon l'opération que vous utilisez.

- Vous pouvez associer des balises à des ressources publiques ou partagées. Toutefois, les balises ne sont disponibles que pour votre AWS compte, et non pour les autres comptes partageant la ressource. En outre, les balises ne sont disponibles que pour les ressources situées dans la AWS région spécifiée pour votre AWS compte.

Pour ajouter, afficher, mettre à jour et supprimer des clés et des valeurs de balise dans les ressources Amazon Pinpoint, vous pouvez utiliser le AWS Command Line Interface (AWS CLI), l'API Amazon Pinpoint, Groupes de ressources AWS l'API de balisage ou un SDK. AWS Pour gérer les clés de balise et les valeurs de toutes les AWS ressources situées dans une AWS région spécifique pour votre AWS compte (y compris les ressources Amazon Pinpoint), utilisez l'API de Groupes de ressources AWS balisage.

Pour plus d'informations sur les commandes de l'interface de ligne de commande que vous pouvez utiliser pour gérer les ressources Amazon Pinpoint, consultez la section Amazon Pinpoint du document Référence de commande AWS CLI.

Pour plus d'informations sur les ressources de l'API Amazon Pinpoint, y compris sur les méthodes HTTP(S), les paramètres et les schémas pris en charge, consultez la Référence de l'API Amazon Pinpoint.

Utiliser les balises Amazon Pinpoint dans les politiques IAM et les opérations d'API

Une fois que vous avez commencé à implémenter des balises, vous pouvez appliquer des autorisations basées sur des balises au niveau des ressources aux politiques Gestion des identités et des accès AWS (IAM) et aux opérations d'API. Cela inclut les opérations qui prennent en charge l'ajout de balises aux ressources lors de la création de ressources. En utilisant les balises de cette manière, vous pouvez contrôler de manière précise quels groupes et utilisateurs de votre AWS compte sont autorisés à créer et à étiqueter des ressources, et quels groupes et utilisateurs sont autorisés à créer, mettre à jour et supprimer des balises de manière plus générale.

Par exemple, vous pouvez créer une stratégie qui permet à un utilisateur d'avoir un accès complet à toutes les ressources Amazon Pinpoint dont le nom est une valeur dans la balise `Owner` de la ressource :

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "ModifyResourceIfOwner",
         "Effect": "Allow",
         "Action": "mobiletargeting:*",
         "Resource": "*",
         "Condition": {
            "StringEqualsIgnoreCase": {
               "aws:ResourceTag/Owner": "${aws:username}"
            }
         }
      }
   ]
}
```

Si vous définissez des autorisations au niveau des ressources basées sur des balises, les autorisations prennent effet immédiatement. Vos ressources sont ainsi plus sécurisées dès leur création et vous pouvez rapidement commencer à appliquer l'utilisation des balises pour les nouvelles ressources. Vous pouvez également utiliser des autorisations au niveau des ressources afin de contrôler les clés et les valeurs de balise qui peuvent être associés à des ressources nouvelles et existantes. Pour plus d'informations, consultez Contrôle de l'accès à l'aide de balises dans le Guide de l'utilisateur AWS IAM.

Ajoutez des balises aux ressources Amazon Pinpoint par programmation

Les exemples suivants montrent comment ajouter une balise à une ressource Amazon Pinpoint à l'aide de l'AWS CLI et de l'API REST Amazon Pinpoint. Vous pouvez également utiliser n'importe quel AWS SDK compatible pour ajouter une balise à une ressource.

Pour ajouter une balise à plusieurs ressources Amazon Pinpoint en une seule opération, utilisez les opérations de balisage des groupes de ressources de l'API AWS CLI ou de l'API de balisage.Groupes de ressources AWS

Ajoutez des balises à l'aide de l'API

Pour créer une nouvelle ressource et y ajouter une balise à l'aide de l'API REST d'Amazon Pinpoint, envoyez une requête POST à l'URI de ressource approprié. Dans le corps de la demande, incluez le paramètre et les valeurs `tags`. L'exemple suivant montre comment définir une balise lorsque vous créez un nouveau projet.

```
POST /v1/apps HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Cache-Control: no-cache

{
   "Name":"MyProject",
   "tags":{
      "key1":"value1"
   }
}
```

Pour ajouter une balise à une ressource existante, envoyez une requête POST à l'URI Tags. Incluez l'ARN (Amazon Resource Name) de la ressource dans l'URI. L'ARN doit être codé par URL. Dans le corps de la requête, incluez le paramètre et les valeurs `tags`, comme indiqué dans l'exemple suivant.

```
POST /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache

{
   "tags":{
      "key1":"value1"
   }
}
```

Ajoutez des balises à l'aide du AWS CLI

Pour créer une nouvelle ressource et y ajouter une étiquette à l'aide de AWS CLI, utilisez la `create` commande appropriée pour la ressource. Incluez le paramètre et les valeurs `tags`. L'exemple suivant montre comment définir des balises lorsque vous créez un nouveau projet.

Linux, macOS, or Unix

```
$ aws pinpoint create-app \
  --create-application-request '{
    "Name":"MyProject",
    "tags": {
      "key1":"value1",
      "key2":"value2"
    }
  }'
```

Windows Command prompt

```
C:\> aws pinpoint create-app ^
     --create-application-request Name=MyProject,tags={key1=value1,key2=value2}
```

Dans l'exemple précédent, procédez comme suit :

- *MyProject*Remplacez-le par le nom que vous souhaitez attribuer au projet.
- Remplacez *key1* et *key2* par les clés des balises que vous souhaitez ajouter à la ressource.
- Remplacez *value1* et *value2* par les valeurs des balises que vous souhaitez ajouter pour les clés respectives.

Pour plus d'informations sur les commandes que vous pouvez utiliser pour créer une ressource Amazon Pinpoint, consultez Référence des commandes AWS CLI.

Pour ajouter une balise à une ressource existante, utilisez la commande `tag-resource` et spécifiez les valeurs appropriées pour les paramètres requis :

Linux, macOS, or Unix

```
$ aws pinpoint tag-resource \
  --resource-arn resource-arn \
```

```
  --tags-model '{
    "tags": {
      "key1":"value1",
      "key2":"value2"
    }
  }'
```

Windows Command Prompt

```
C:\> aws pinpoint tag-resource ^
     --resource-arn resource-arn ^
     --tags-model tags={key1=value1,key2=value2}
```

Dans l'exemple précédent, procédez comme suit :

- *resource-arn*Remplacez-le par le Amazon Resource Name (ARN) de la ressource à laquelle vous souhaitez ajouter une balise.
- Remplacez *key1* et *key2* par les clés des balises que vous souhaitez ajouter à la ressource.
- Remplacez *value1* et *value2* par les valeurs des balises que vous souhaitez ajouter pour les clés respectives.

Afficher les balises pour les ressources Amazon Pinpoint de manière programmatique

Les exemples suivants montrent comment utiliser l'AWS CLI et l'API REST Amazon Pinpoint pour afficher une liste de toutes les balises (clés et valeurs) associées à une ressources Amazon Pinpoint. Vous pouvez également utiliser n'importe quel AWS SDK compatible pour afficher les balises associées à une ressource.

Afficher les tags à l'aide de l'API

Pour utiliser l'API REST Amazon Pinpoint afficher toutes les balises qui sont associées à une ressource spécifique, envoyez une demande GET à l'URI Tags en incluant l'Amazon Resource Name (ARN) de la ressource dans l'URI. L'ARN doit être codé par URL. Par exemple, la demande suivante récupère tous les tags associés à une campagne spécifiée (*resource-arn*) :

```
GET /v1/tags/resource-arn HTTP/1.1
```

```
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

La réponse JSON à la demande inclut un objet `tags`. L'objet `tags` répertorie toutes les clés et valeurs de balise associées à la campagne.

Pour afficher toutes les balises qui sont associées à plusieurs ressources du même type, envoyez une demande GET à l'URI approprié pour ce type de ressource. Par exemple, la demande suivante permet de récupérer des informations sur toutes les campagnes du projet spécifié (*`application-id`*) :

```
GET /v1/apps/application-id/campaigns HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

La réponse JSON à la demande répertorie toutes les campagnes du projet. L'objet `tags` de chaque campagne répertorie toutes les clés et les valeurs de balise qui sont associées à la campagne.

Afficher les balises à l'aide du AWS CLI

AWS CLI Pour afficher la liste des balises associées à une ressource spécifique, exécutez la `list-tags-for-resource` commande et spécifiez le nom de ressource Amazon (ARN) de la ressource pour le `resource-arn` paramètre, comme indiqué dans l'exemple suivant.

Linux, macOS, or Unix

```
$ aws pinpoint list-tags-for-resource \
  --resource-arn resource-arn
```

Windows Command Prompt

```
C:\> aws pinpoint list-tags-for-resource ^
     --resource-arn resource-arn
```

Pour afficher la liste de toutes les ressources Amazon Pinpoint dotées de balises, ainsi que de toutes les balises associées à chacune de ces ressources, utilisez la commande get-resources de l'API

de balisage. Groupes de ressources AWS Définissez le paramètre `resource-type-filters` sur `mobiletargeting`, comme illustré dans l'exemple suivant.

Linux, macOS, or Unix

```
$ aws resourcegroupstaggingapi get-resources \
     --resource-type-filters "mobiletargeting"
```

Windows Command Prompt

```
C:\> aws resourcegroupstaggingapi get-resources ^
     --resource-type-filters "mobiletargeting"
```

Le résultat de la commande est une liste de toutes ARNs les ressources Amazon Pinpoint dotées de balises. La liste inclut toutes les clés et les valeurs de balise qui sont associées à chaque ressource.

Mettre à jour ou remplacer les balises pour les ressources Amazon Pinpoint par programmation

Il existe plusieurs façons de mettre à jour (remplacer) une balise pour une ressource Amazon Pinpoint. La meilleure façon de mettre à jour une balise dépend :

- Du type de ressource pour lequel vous souhaitez mettre à jour les balises.
- Si vous voulez mettre à jour une balise pour une seule ressource ou pour plusieurs ressources en même temps.
- De si vous voulez mettre à jour une clé de balise, une valeur de balise, ou les deux.

Pour mettre à jour une balise pour un projet Amazon Pinpoint ou pour plusieurs ressources en même temps, utilisez les opérations de balisage des groupes de ressources de l'API AWS CLI ou de l'API de balisage.Groupes de ressources AWS L'API Amazon Pinpoint ne fournit actuellement aucune prise en charge directe pour l'une ou l'autre de ces tâches.

Pour mettre à jour une balise pour une ressource, vous pouvez supprimer la balise actuelle et ajouter une nouvelle balise à l'aide de l'API Amazon Pinpoint.

Supprimer les balises des ressources Amazon Pinpoint par programmation

Les exemples suivants montrent comment supprimer une balise (la clé et la valeur) d'une ressource Amazon Pinpoint à l'aide de l'AWS CLI et l'API REST Amazon Pinpoint. Vous pouvez également utiliser n'importe quel AWS SDK compatible pour supprimer une balise d'une ressource.

Pour supprimer une balise de plusieurs ressources Amazon Pinpoint en une seule opération, utilisez les opérations de balisage des groupes de ressources de l'API AWS CLI ou de l'API de balisage.Groupes de ressources AWS Pour supprimer uniquement une valeur de balise spécifique, pas une clé de balise, d'une ressource, mettez à jour les balises de la ressource.

Supprimer les tags à l'aide de l'API

Pour supprimer une balise d'une ressource à l'aide de l'API REST Amazon Pinpoint, envoyez une demande DELETE à l'URI des balises. Pour l'URI, incluez l'Amazon Resource Name (ARN) de la ressource dont vous souhaitez supprimer une balise, suivi du paramètre `tagKeys` et de la balise à supprimer. Par exemple :

```
https://endpoint/v1/tags/resource-arn?tagKeys=key
```

Où :

- *`endpoint`*est le point de terminaison Amazon Pinpoint pour la AWS région qui héberge la ressource.
- *`resource-arn`*est l'ARN de la ressource dont vous souhaitez supprimer une balise.
- *`key`*est le tag que vous souhaitez supprimer de la ressource.

Tous les paramètres doivent être codés par URL.

Pour supprimer plusieurs clés de balises et leurs valeurs associées d'une ressource, ajoutez le paramètre et l'argument `tagKeys` pour chaque balise à supprimer, séparés par une esperluette (&). Par exemple :

```
https://endpoint/v1/tags/resource-arn?tagKeys=key1&tagKeys=key2
```

Tous les paramètres doivent être codés par URL.

Supprimez les tags à l'aide du AWS CLI

Pour supprimer une balise d'une ressource à l'aide de AWS CLI, exécutez la `untag-resource` commande. Indiquez le paramètre et l'argument `tag-keys`, comme le montre l'exemple suivant.

Linux, macOS, or Unix

```
$ aws pinpoint untag-resource \
  --resource-arn resource-arn \
  --tag-keys key1 key2
```

Windows Command Prompt

```
C:\> aws pinpoint untag-resource ^
     --resource-arn resource-arn ^
     --tag-keys key1 key2
```

Dans l'exemple précédent, apportez les modifications suivantes :

- *resource-arn*Remplacez-le par l'ARN de la ressource dont vous souhaitez supprimer les balises.
- Remplacez *key1* et *key2* par les clés des balises que vous souhaitez supprimer de la ressource.

Intégrez Amazon Pinpoint à votre application

Intégrez Amazon Pinpoint à votre code client pour comprendre et impliquer vos utilisateurs.

Une fois que vous avez intégré et que vos utilisateurs ont lancé votre application, celle-ci se connecte au service Amazon Pinpoint pour ajouter ou mettre à jour des points de terminaison. Les points de terminaison représente les destinations auxquelles vous pouvez envoyer des messages, comme les périphériques utilisateur, les adresses de messagerie et les numéros de téléphone.

Votre application peut ensuite fournir des données d'utilisation ou des événements. Affichez les données d'événement dans la console Amazon Pinpoint pour découvrir le nombre d'utilisateurs que vous avez, à quelle fréquence ils utilisent votre application, à quel moment ils les utilisent, et bien plus encore.

Une fois que votre application a fourni les points de terminaison et les événements, vous pouvez utiliser ces informations pour personnaliser les campagnes de messages pour des publics, ou pour des segments spécifiques. (Vous pouvez également adresser directement le message à de simples listes de destinataires sans créer de campagnes.)

Utilisez les rubriques de cette section pour intégrer Amazon Pinpoint à un client mobile ou web. Ces rubriques incluent des exemples de code et des procédures à intégrer à une application Android JavaScript, Swift ou Flutter. Pour démarrer l'intégration de vos applications, consultez the section called “Connectez votre application frontale à l'aide d'Amplify”.

En dehors de votre client, vous pouvez utiliser l'API compatible AWS SDKs ou Amazon Pinpoint pour importer des points de terminaison, exporter des données d'événements, définir des segments de clientèle, créer et exécuter des campagnes, etc.

Rubriques

- Utilisation d'Amazon Pinpoint avec un SDK AWS
- Connectez votre application frontale à Amazon AWS Pinpoint à l'aide d'Amplify
- Enregistrez les points de terminaison Amazon Pinpoint dans votre application
- Signalez les événements Amazon Pinpoint dans votre application

Utilisation d'Amazon Pinpoint avec un SDK AWS

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK pour C++	AWS SDK pour C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK pour Go	AWS SDK pour Go exemples de code
AWS SDK pour Java	AWS SDK pour Java exemples de code
AWS SDK pour JavaScript	AWS SDK pour JavaScript exemples de code
AWS SDK pour Kotlin	AWS SDK pour Kotlin exemples de code
AWS SDK pour .NET	AWS SDK pour .NET exemples de code
AWS SDK pour PHP	AWS SDK pour PHP exemples de code
Outils AWS pour PowerShell	Outils AWS pour PowerShell exemples de code
AWS SDK pour Python (Boto3)	AWS SDK pour Python (Boto3) exemples de code
AWS SDK pour Ruby	AWS SDK pour Ruby exemples de code
AWS SDK pour Rust	AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
AWS SDK pour Swift	AWS SDK pour Swift exemples de code

Pour des exemples spécifiques à Amazon Pinpoint, consultez Exemples de code pour Amazon Pinpoint à l'aide de AWS SDKs

> **Exemple de disponibilité**
>
> Vous n'avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Provide feedback (Fournir un commentaire) en bas de cette page.

Connectez votre application frontale à Amazon AWS Pinpoint à l'aide d'Amplify

Utilisez AWS Amplify pour intégrer votre application à. AWS Pour les applications Swift, consultez Démarrer dans la documentation Amplify pour Swift. Pour les applications Android, consultez Démarrer dans la documentation du kit SDK Android. Pour l'application React Native, voir Getting started dans la documentation Amplify JavaScript . Pour les applications Flutter, consultez Démarrer dans la documentation du kit SDK Flutter. Ces rubriques vous aident à :

- Configurer vos ressources de backend.
- Connectez votre application aux ressources du backend à l'aide des bibliothèques Amplify.

Pour en savoir plus sur la connexion de votre application frontale à Amazon Pinpoint à des fins d'analyse, de messagerie intégrée et de notifications push, consultez Amplify.AWS

Étape suivante

Après avoir intégré AWS Amplify à votre application, mettez à jour votre code pour enregistrer les appareils de vos utilisateurs en tant que points de terminaison. Pour de plus amples informations, veuillez consulter Enregistrez les points de terminaison Amazon Pinpoint dans votre application.

Enregistrez les points de terminaison Amazon Pinpoint dans votre application

Lorsqu'un utilisateur démarre une session (par exemple, en lançant votre application mobile), votre application mobile ou web peut automatiquement enregistrer (ou mettre à jour) un point de terminaison avec Amazon Pinpoint. Le point de terminaison représente le périphérique avec lequel l'utilisateur démarre la session. Il inclut les attributs qui décrivent l'appareil. Il peut également inclure des attributs personnalisés que vous définissez. Les points de terminaison peuvent également

représenter d'autres moyens de communication avec les clients, par exemple des adresses e-mail ou des numéros de téléphone mobile.

Une fois que votre application a enregistré les points de terminaison, vous pouvez segmenter votre public en fonction des attributs de point de terminaison. Vous pouvez ensuite lancer ces segments avec des campagnes de messagerie sur mesure. Vous pouvez également utiliser la page Analytics de la console Amazon Pinpoint pour afficher des graphiques sur l'enregistrement et l'activité des points de terminaison, notamment les graphiques New endpoints et les Daily active endpoints.

Vous pouvez attribuer un même ID utilisateur à plusieurs points de terminaison. Un ID d'utilisateur représente un seul utilisateur, tandis que chaque point de terminaison qui est affecté à l'ID utilisateur représente l'une des appareils de l'utilisateur. Une fois que vous avez attribué un utilisateur IDs à vos points de terminaison, vous pouvez consulter les graphiques relatifs à l'activité des utilisateurs dans la console, tels que les utilisateurs actifs quotidiens et les utilisateurs actifs mensuels.

Avant de commencer

Si ce n'est pas déjà fait, intégrez le SDK AWS mobile pour Android ou iOS ou intégrez la bibliothèque AWS JavaScript Amplify à votre application. Pour de plus amples informations, veuillez consulter Connectez votre application frontale à Amazon AWS Pinpoint à l'aide d'Amplify.

Enregistrez les terminaux sur le AWS mobile SDKs pour Android ou iOS

Vous pouvez utiliser le AWS Mobile SDKs pour Android ou iOS pour enregistrer et personnaliser les terminaux. Pour plus d'informations et pour afficher les exemples de code, consultez les documents suivants :

- Enregistrement des points de terminaison dans votre application dans la documentation du kit SDK Android.
- Enregistrement des points de terminaison dans votre application dans la documentation du kit SDK iOS.

Enregistrez les points de terminaison dans la bibliothèque AWS Amplify JavaScript

Vous pouvez utiliser la JavaScript bibliothèque AWS Amplify pour enregistrer et mettre à jour les points de terminaison dans vos applications. Pour plus d'informations et pour consulter des exemples de code, voir Mettre à jour le point de terminaison dans la documentation AWS Amplify JavaScript .

Étapes suivantes

Une fois que vous avez mis à jour votre application pour enregistrer les points de terminaison, les informations sur l'appareil et les attributs personnalisés sont fournis à Amazon Pinpoint lorsque les utilisateurs lancent votre application. Vous pouvez utiliser ces informations pour définir les segments de public ciblé. Vous pouvez également utiliser la console pour consulter les statistiques des points de terminaison et les utilisateurs auxquels un utilisateur est attribué IDs. Vous pouvez également suivre les étapes décrites Signalez les événements Amazon Pinpoint dans votre application pour mettre à jour votre application afin de signaler les données d'utilisation.

Signalez les événements Amazon Pinpoint dans votre application

Dans votre application mobile ou Web, vous pouvez utiliser AWS Mobile SDKs ou l'API d'événements Amazon Pinpoint pour signaler des données d'utilisation, ou des événements, à Amazon Pinpoint. Vous pouvez rapporter les événements pour capturer des informations telles que la durée des sessions, le comportement d'achat des utilisateurs, les tentatives de connexion ou tout autre type d'événement personnalisé utile à vos besoins.

Une fois que votre application rapporte les événements, vous pouvez afficher les analyses dans la console Amazon Pinpoint. Les graphiques de la page Analytics (Analyse) fournissent des métriques sur de nombreux aspects du comportement des utilisateurs. Pour plus d'informations, consultez Référence graphique pour les analyses Amazon Pinpoint dans le Guide de l'utilisateur Amazon Pinpoint.

Pour analyser et stocker vos données d'événement en dehors d'Amazon Pinpoint, vous pouvez configurer Amazon Pinpoint pour qu'il diffuse les données en streaming vers Amazon Kinesis. Pour de plus amples informations, veuillez consulter Diffusez les données d'événements des applications via Kinesis et Firehose à l'aide d'Amazon Pinpoint.

En utilisant les JavaScript bibliothèques AWS Mobile SDKs et AWS Amplify, vous pouvez appeler l'API Amazon Pinpoint pour signaler les types d'événements suivants :

Événements de session

Indiquent quand et à quelle fréquence les utilisateurs ouvrent et ferment votre application.

Une fois que votre application a rapporté les événements de session, utilisez la page Analytics de la console Amazon Pinpoint pour afficher les graphiques Sessions, Daily active endpoints, 7-day retention rate, etc.

Evénements personnalisés

Événements non standard que vous définissez en affectant un type d'événement personnalisé. Vous pouvez ajouter des métriques et des attributs personnalisés à un événement personnalisé.

Sur la page Analytics de la console, l'onglet Events affiche les métriques de tous les événements personnalisés rapportés par votre application.

Événements de monétisation

Représentent les revenus générés par votre application et le nombre d'articles achetés par les utilisateurs.

Sur la page Analytics, l'onglet Revenue affiche les graphiques sur les revenus (Revenue), les utilisateurs payants (Paying users), le nombre d'unités vendues (Units sold), etc.

Événements d’authentification

Indiquez la fréquence à laquelle les utilisateurs s'authentifient auprès de votre application.

Sur la page Analytics (Analyse), l'onglet Utilisateurs affiche les graphiques pour Sign-ins (Connexions), Sign-ups (Inscriptions) et Authentication failures (Échecs d'authentification).

Avant de commencer

Si vous ne l'avez pas déjà fait, effectuez les opérations suivantes :

- Intégrez votre application à AWS Amplify. Consultez Connectez votre application frontale à Amazon AWS Pinpoint à l'aide d'Amplify.
- Mettez à jour votre application pour enregistrer les points de terminaison. Consultez Enregistrez les points de terminaison Amazon Pinpoint dans votre application.

Signaler des événements avec le AWS mobile SDKs pour Android ou iOS

Vous pouvez activer une application mobile pour signaler des événements à Amazon Pinpoint à l'aide de AWS Mobile pour SDKs iOS et Android.

Pour plus d'informations sur la mise à jour de votre application afin d'enregistrer et de soumettre des événements à Amazon Pinpoint, consultez les pages suivantes de la documentation Amplify AWS :

- Analyse dans la documentation du kit SDK iOS

- Analyse dans la documentation du kit SDK Android

Signalez des événements avec la bibliothèque AWS Amplify JavaScript

Vous pouvez activer JavaScript les applications React Native pour signaler les événements d'utilisation des applications à Amazon Pinpoint à l'aide de la bibliothèque Amplify AWS . JavaScript Pour plus d'informations sur la mise à jour de votre application afin de soumettre des événements à Amazon Pinpoint, consultez Analytics dans la documentation d'Amplify AWS . JavaScript

Signalez des événements avec l'API Amazon Pinpoint

Vous pouvez utiliser l'API Amazon Pinpoint ou un AWS SDK pour envoyer des événements à Amazon Pinpoint en masse. Pour plus d'informations, consultez Événements dans la Référence de l'API Amazon Pinpoint.

Étapes suivantes

Une fois que vous avez mis à jour votre application pour signaler des événements, elle envoie les données d'utilisation à Amazon Pinpoint. Vous pouvez consulter ces données dans la console et les diffuser sur Amazon Kinesis. Vous pouvez également mettre à jour votre application pour gérer les notifications push que vous envoyez avec Amazon Pinpoint. Pour plus d'informations, consultez les rubriques suivantes dans le Guide de l'utilisateur AWS final pour la messagerie push.

- Configuration des notifications push
- Configuration des notifications Swift Push
- Configuration des notifications push Android
- Configuration des notifications push Flutter
- Configuration des notifications push React Native
- Créer un projet
- Gestion des notifications push

Envoyez des messages transactionnels depuis votre application à l'aide d'Amazon Pinpoint

Vous pouvez utiliser l'API Amazon Pinpoint et AWS SDKs envoyer des messages transactionnels directement depuis votre application. Les messages transactionnels sont des messages que vous envoyez à des destinataires spécifiques, par opposition aux messages que vous envoyez à des segments. Vous pouvez souhaiter envoyer des messages transactionnels plutôt que des messages basés sur des campagnes pour plusieurs raisons . Par exemple, vous pouvez envoyer une confirmation de commande par e-mail lorsqu'un client passe une commande. Vous pouvez également envoyer un mot de passe à usage unique par SMS ou message vocal, qu'un client peut utiliser pour terminer le processus de création d'un compte pour votre service.

Cette section inclut un exemple de code dans plusieurs langages de programmation, que vous pouvez utiliser pour commencer à envoyer des e-mails transactionnels, des SMS et des messages vocaux.

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

Rubriques de cette section :

- Envoyez des e-mails transactionnels à l'aide d'Amazon Pinpoint
- Envoyez des SMS transactionnels à l'aide d'Amazon Pinpoint
- Envoyer des messages vocaux à l'aide d'Amazon Pinpoint

Envoyez des e-mails transactionnels à l'aide d'Amazon Pinpoint

Cette section fournit des exemples de code complet que vous pouvez utiliser pour envoyer des e-mails transactionnels via Amazon Pinpoint.

- En utilisant l' SendMessages opération de l'API Amazon Pinpoint : vous pouvez utiliser l'`SendMessages`opération de l'API Amazon Pinpoint pour envoyer des messages sur tous les canaux pris en charge par Amazon Pinpoint, y compris les canaux de notification push, les SMS, les appels vocaux et les e-mails.

  L'utilisation de cette opération a pour avantage que la syntaxe de demande pour l'envoi des messages est très similaire sur tous les canaux. Il est ainsi plus facile de remanier votre code

existant. L'`SendMessages`opération vous permet également de remplacer le contenu de vos e-mails et d'envoyer des e-mails au point de terminaison Amazon Pinpoint IDs plutôt qu'à des adresses e-mail spécifiques.

Cette section inclut un exemple de code dans plusieurs langages de programmation, que vous pouvez utiliser pour commencer à envoyer des e-mails transactionnels.

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

Choisissez une méthode pour envoyer un e-mail

La meilleure méthode à utiliser pour l'envoi d'e-mails transactionnels dépend de votre cas d'utilisation. Par exemple, si vous devez envoyer un e-mail à l'aide d'une application tierce, ou si aucun AWS SDK n'est disponible pour votre langage de programmation, vous devrez peut-être utiliser l'interface SMTP. Si vous souhaitez envoyer des messages dans d'autres canaux que ceux pris en charge par Amazon Pinpoint et que vous souhaitez utiliser un code cohérent pour créer ces demandes, vous devez utiliser l'opération `SendMessages` de l'API Amazon Pinpoint.

Choisissez entre Amazon Pinpoint et Amazon SES

Si vous envoyez un grand nombre d'e-mails transactionnels, tels que des confirmations d'achat ou des messages de réinitialisation de mot de passe, envisagez d'utiliser Amazon SES. Amazon SES dispose d'une API et d'une interface SMTP, qui sont toutes deux adaptées à l'envoi d'e-mails à partir de vos applications ou services. Il offre également des fonctionnalités de messagerie supplémentaires, notamment des fonctionnalités de réception des e-mails, des ensembles de configuration et des fonctionnalités d'autorisation d'envoi.

Amazon SES inclut également une interface SMTP que vous pouvez intégrer à vos applications tierces existantes, y compris les services de gestion de la relation client (CRM) tels que Salesforce. Pour plus d'informations sur l'envoi d'e-mails à l'aide d'Amazon SES, veuillez consulter le manuel du développeur Amazon Simple Email Service.

Envoyer un e-mail à l'aide de l'API Amazon Pinpoint

Cette section contient des exemples de code complets que vous pouvez utiliser pour envoyer des e-mails via l'API Amazon Pinpoint à l'aide d'un AWS SDK. Vous devez avoir vérifié une adresse e-mail ou un domaine avant de pouvoir envoyer un message.

C#

Utilisez cet exemple pour envoyer des e-mails à l'aide du kit AWS SDK pour .NET. Cette procédure suppose que vous avez déjà installé et configuré le kit SDK pour .NET. Pour plus d'informations, consultez Mise en route avec AWS SDK pour .NET dans le Guide du développeur AWS SDK pour .NET .

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 AWS informations d'identification dans le Guide du AWS SDK pour .NET développeur.

Cet exemple de code a été testé à l'aide de la AWS SDK pour .NET version 3.3.29.13 et de la version 2.1.2 du runtime .NET Core.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load test settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally load local settings.
        .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;
```

```
        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
        string toAddress = configuration["ToAddress"]!;

        // The Amazon Pinpoint project/application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or application
        // that you choose.
        string appId = configuration["AppId"]!;

        try
        {
            await SendEmailMessage(region, appId, toAddress, senderAddress);
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }

    public static async Task<MessageResponse> SendEmailMessage(
        string region, string appId, string toAddress, string senderAddress)
    {
        var client = new
 AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        // The subject line of the email.
        string subject = "Amazon Pinpoint Email test";

        // The body of the email for recipients whose email clients don't
        // support HTML content.
        string textBody = @"Amazon Pinpoint Email Test (.NET)"
                          + "\n---------------------------------"
                          + "\nThis email was sent using the Amazon Pinpoint API
 using the AWS SDK for .NET.";

        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK for .NET)</
h1>"
```

```
                            + "\n  <p>This email was sent using the "
                            + "\n    <a href='https://aws.amazon.com/pinpoint/'>Amazon
 Pinpoint</a> API "
                            + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                            + "\n  </p>"
                            + "\n</body>"
                            + "\n</html>";

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        string charset = "UTF-8";

        var sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses = new Dictionary<string, AddressConfiguration>
                {
                    {
                        toAddress,
                        new AddressConfiguration
                        {
                            ChannelType = ChannelType.EMAIL
                        }
                    }
                },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    EmailMessage = new EmailMessage
                    {
                        FromAddress = senderAddress,
                        SimpleEmail = new SimpleEmail
                        {
                            HtmlPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = htmlBody
                            },
                            TextPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = textBody
```

```
                            },
                            Subject = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = subject
                            }
                        }
                    }
                }
            }
        };
        Console.WriteLine("Sending message...");
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        Console.WriteLine("Message sent!");
        return response.MessageResponse;
    }
}
```

Java

Utilisez cet exemple pour envoyer des e-mails à l'aide du kit AWS SDK pour Java. Cette procédure suppose que vous avez déjà installé et configuré le kit AWS SDK for Java 2.x. Pour de plus amples informations, veuillez consulter Mise en route dans le Manuel du développeur AWS SDK for Java 2.x .

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 Définir les informations d'identification et la région par défaut dans le Guide du développeur AWS SDK pour Java .

Cet exemple de code a été testé avec la version 2.3.1 et la AWS SDK pour Java version 11.0.1 d'OpenJDK.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
```

```
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
```

```
 */
public class SendEmailMessage {

       // The character encoding the you want to use for the subject line and
       // message body of the email.
       public static String charset = "UTF-8";

   // The body of the email for recipients whose email clients support HTML
 content.
   static final String body = """
       Amazon Pinpoint test (AWS SDK for Java 2.x)

       This email was sent through the Amazon Pinpoint Email API using the AWS SDK
 for Java 2.x

       """;

       public static void main(String[] args) {
              final String usage = """

                                Usage:    <subject> <appId> <senderAddress>
 <toAddress>

           Where:
              subject - The email subject to use.
              senderAddress - The from address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email\s
              toAddress - The to address. This address has to be verified in Amazon
 Pinpoint in the region you're using to send email\s
           """;

       if (args.length != 3) {
           System.out.println(usage);
           System.exit(1);
       }

       String subject = args[0];
       String senderAddress = args[1];
       String toAddress = args[2];
       System.out.println("Sending a message");
       PinpointEmailClient pinpoint = PinpointEmailClient.builder()
           .region(Region.US_EAST_1)
           .build();
```

```
        sendEmail(pinpoint, subject, senderAddress, toAddress);
        System.out.println("Email was sent");
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress) {
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

```
}
```

Pour l'exemple complet du SDK, consultez SendEmailMessage.java on. GitHub

JavaScript (Node.js)

Utilisez cet exemple pour envoyer un e-mail 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 Node.js. Pour plus d'informations, consultez Getting started in the AWS SDK for JavaScript in Node.js Developer Guide.

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.

Cet exemple de code a été testé à l'aide du SDK pour JavaScript Node.js version 2.388.0 et Node.js version 11.7.0.

```
"use strict";

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

// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";

// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
```

```
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
// message body of the email.
var charset = "UTF-8";

// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
```

```
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

//Try to send the email.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"][toAddress]["MessageId"]
    );
  }
});
```

Python

Utilisez cet exemple pour envoyer des e-mails 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). Pour plus d'informations, consultez Démarrage rapide dans la Référence d'API du kit AWS SDK pour Python (Boto).

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

logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
 account
                         is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and message
                     body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
 can
                         display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
```

```
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": sender,
                        "SimpleEmail": {
                            "Subject": {"Charset": char_set, "Data": subject},
                            "HtmlPart": {"Charset": char_set, "Data": html_message},
                            "TextPart": {"Charset": char_set, "Data": text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]["Result"].items()
        }


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -------------------------------------
    This email was sent with Amazon Pinpoint using the AWS SDK for Python (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
                """
    html_message = """<html>
    <head></head>
    <body>
      <h1>Amazon Pinpoint Test (SDK for Python (Boto3)</h1>
      <p>This email was sent with
        <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
        <a href='https://aws.amazon.com/sdk-for-python/'>
          AWS SDK for Python (Boto3)</a>.</p>
```

```
    </body>
    </html>
                """

    print("Sending email.")
    message_ids = send_email_message(
        boto3.client("pinpoint"),
        app_id,
        sender,
        [to_address],
        char_set,
        subject,
        html_message,
        text_message,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

Vous pouvez également utiliser des modèles de messages pour envoyer des e-mails, comme illustré dans l'exemple suivant :

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

logger = logging.getLogger(__name__)


def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
 template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
```

```
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
                         account is in the sandbox, these addresses must be
 verified.
    :param template_name: The name of the email template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {"EmailMessage": {"FromAddress": sender}},
                "TemplateConfiguration": {
                    "EmailTemplate": {
                        "Name": template_name,
                        "Version": template_version,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]["Result"].items()
        }


def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
```

```
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
        template_version,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

Ces exemples supposent 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).

Ajouter des en-têtes de désabonnement à un e-mail à l'aide d'Amazon Pinpoint

> **Note**
>
> Avant de pouvoir utiliser les en-têtes d'e-mail, vous devez configurer un rôle d'envoi d'orchestration des e-mails si vous envoyez des e-mails dans le cadre d'une campagne ou d'un parcours. Pour l'envoi direct d'e-mails, vous devez disposer des autorisations pour `ses:SendEmail` et`ses:SendRawEmail`. Pour plus d'informations, consultez la section Création d'un rôle d'envoi d'orchestration d'e-mails dans le guide de l'utilisateur Amazon Pinpoint.

L'inclusion d'un lien de désabonnement dans votre e-mail est une bonne pratique, et dans certains pays, c'est une obligation légale. Pour ajouter un lien de désabonnement en un clic, ajoutez les en-têtes suivants :

1. Définissez le nom de l'en-tête sur `List-Unsubscribe` et définissez la valeur sur votre lien de désinscription. Ce lien doit prendre en charge les demandes HTTP POST pour permettre le traitement de la demande de désinscription du destinataire.

2. Définissez le nom de l'en-tête sur `List-Unsubscribe-Post` et définissez la valeur sur`List-Unsubscribe=One-Click`.

Vous pouvez ajouter jusqu'à 15 en-têtes à un e-mail. Pour obtenir la liste des en-têtes pris en charge, consultez les champs d'en-tête Amazon SES dans le manuel Amazon Simple Email Service Developer Guide.

L'exemple suivant montre comment envoyer un e-mail avec des en-têtes de désabonnement à l'aide du AWS Command Line Interface. Pour plus d'informations sur la configuration du AWS CLI, voir Configurer le AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Dans la commande suivante, effectuez les opérations suivantes :

- Remplacez *AppId* par l'identifiant de votre application.
- Remplacez *richard_roe@example.com* par l'adresse e-mail du destinataire.
- *https://example.com/unsub*Remplacez-le par votre lien de désinscription.
- Remplacez *example123456* par un identifiant unique pour le destinataire.

```
aws pinpoint send-messages --application-id AppId --message-request '{
 "Addresses": {
   "richard_roe@example.com": {
     "ChannelType": "EMAIL"
   }
 },
 "MessageConfiguration": {
   "EmailMessage": {
     "Substitutions": {
       "url": [
         "https://example.com/unsub"
       ],
        "id1": [
          "/example123456"
        ]
     },
     "SimpleEmail": {
       "TextPart": {
         "Data": "Sample email message with an subscribe header",
         "Charset": "UTF-8"
       },
       "Subject": {
```

```
          "Data": "Hello",
          "Charset": "UTF-8"
        },
        "Headers": [
          {
            "Name": "List-Unsubscribe",
            "Value": "{{url}}{{id1}}"
          },
          {
            "Name": "List-Unsubscribe-Post",
            "Value": "List-Unsubscribe=One-Click"
          }
        ]
      }
    }
  }
}'
```

Envoyez des SMS transactionnels à l'aide d'Amazon Pinpoint

Vous pouvez utiliser l'API Amazon Pinpoint pour envoyer des SMS (messages texte) à des numéros de téléphone ou à des terminaux spécifiques. IDs Cette section contient des exemples de code complets que vous pouvez utiliser pour envoyer des SMS via l'API Amazon Pinpoint à l'aide d'un AWS SDK. Votre compte doit être en cours de production et vous devez disposer d'une identité d'origine active capable d'envoyer des SMS.

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

C#

Utilisez cet exemple pour envoyer un SMS à l'aide du kit AWS SDK pour .NET. Cette procédure suppose que vous avez déjà installé et configuré le kit SDK pour .NET. Pour de plus amples informations, veuillez consulter Mise en route dans le Manuel du développeur AWS SDK pour .NET .

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 pour un utilisateur IAM existant. Pour de plus amples informations, veuillez consulter la section Configurer les informations d'identification AWS du Guide du développeur pour le kit AWS SDK pour .NET .

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone number
        // or short code that you specify has to be associated with your Amazon
 Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number.  For best results, you should specify the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;

        // The Pinpoint project/ application ID to use when you send this message.
        // Make sure that the SMS channel is enabled for the project or application
        // that you choose.
        string appId = configuration["AppId"]!;

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;
```

```
        // The registered keyword associated with the originating short code.
        string? registeredKeyword = configuration["RegisteredKeyword"];

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        string? senderId = configuration["SenderId"];

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
 {response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
 {messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
 originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";


        var client = new
 AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));
```

```
        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
 ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

Java

Utilisez cet exemple pour envoyer un SMS à l'aide du kit AWS SDK pour Java. Cette procédure suppose que vous avez déjà installé et configuré le kit SDK pour Java. Pour de plus amples informations, veuillez consulter Mise en route dans le Manuel du développeur AWS SDK pour Java .

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 pour un utilisateur IAM existant. Pour plus d'informations, consultez Définir les informations d'identification et la région par défaut dans le Guide du développeur AWS SDK pour Java .

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendMessage {

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";
```

```
        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application ID
 to use when you send this message.
                                  originationNumber - The phone number or short code
 that you specify has to be associated with your Amazon Pinpoint account. For best
 results, specify long codes in E.164 format (for example, +1-555-555-5654).
                                  destinationNumber - The recipient's phone number.
 For best results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).\s
                                  """;

                if (args.length != 4) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                sendSMSMessage(pinpoint, message, appId, originationNumber,
 destinationNumber);
                pinpoint.close();
```

```
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request = SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();

                        SendMessagesResponse response =
pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();
```

```
                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" + v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

Pour l'exemple complet du SDK, consultez SendMessage.java on. GitHub

JavaScript (Node.js)

Utilisez cet exemple pour envoyer un message SMS 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. Pour plus d'informations, consultez Getting started in the AWS SDK for JavaScript in Node.js Developer Guide.

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 pour un utilisateur IAM 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";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number.  For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";
```

```
// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-countries.html
var senderId = "MySenderID";

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

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
```

```
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

Python

Utilisez cet exemple pour envoyer un SMS 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. Pour plus d'informations, consultez Quickstart in AWS SDK for Python (Boto3) Getting Started.

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

logger = logging.getLogger(__name__)


def send_sms_message(
```

```
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you send
                   this message. The SMS channel must be enabled for the project or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This phone
                               number must be associated with your Amazon Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you send
                         time-sensitive content, specify TRANSACTIONAL. If you send
                         marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]
```

```
def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK for
 "
        "Python (Boto 3)."
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

Vous pouvez également utiliser des modèles de SMS pour envoyer des e-mails, comme illustré dans l'exemple suivant :

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

logger = logging.getLogger(__name__)


def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
```

```
    origination_number,
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
    :param message_type: The type of SMS message (promotional or transactional).
    :param origination_number: The phone number that the message is sent from.
    :param template_name: The name of the SMS template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
                "TemplateConfiguration": {
                    "SMSTemplate": {"Name": template_name, "Version":
 template_version}
                },
            },
        )

    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]
```

```
def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),
        project_id,
        destination_number,
        message_type,
        origination_number,
        template_name,
        template_version,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

Ces exemples supposent 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 IAM existant. Pour plus d'informations, consultez Informations d'identité dans la Référence des API du kit SDK AWS pour Python (Boto).

Envoyer des messages vocaux à l'aide d'Amazon Pinpoint

Vous pouvez utiliser l'API Amazon Pinpoint pour envoyer des messages vocaux à des numéros de téléphone spécifiques. Cette section contient des exemples de code complets que vous pouvez utiliser pour envoyer des messages vocaux via l'API Amazon Pinpoint SMS and Voice à l'aide d'un AWS SDK. Votre compte doit être en cours de production et vous devez disposer d'une identité d'origine active capable d'envoyer des messages vocaux.

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

Java

Utilisez cet exemple pour envoyer un message vocal à l'aide du kit AWS SDK pour Java. Cette procédure suppose que vous avez déjà installé et configuré le kit SDK pour Java. Pour plus d'informations, consultez Mise en route dans le Manuel du développeur AWS SDK pour Java .

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, voir Configurer les AWS informations d'identification et la région pour le développement dans le Guide du AWS SDK pour Java développeur.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";

    // The language to use when sending the message. For a list of supported
    // languages, see
    // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
    static final String languageCode = "en-US";

    // The content of the message. This example uses SSML to customize and control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
    static final String ssmlMessage = "<speak>This is a test message sent from "
            + "<emphasis>Amazon Pinpoint</emphasis> "
            + "using the <break strength='weak'/>AWS "
            + "SDK for Java. "
            + "<amazon:effect phonation='soft'>Thank "
            + "you for listening.</amazon:effect></speak>";

    public static void main(String[] args) {

        final String usage = """
                Usage:   <originationNumber> <destinationNumber>\s

                Where:
                  originationNumber - The phone number or short code that you
 specify has to be associated with your Amazon Pinpoint account. For best results,
 specify long codes in E.164 format (for example, +1-555-555-5654).
                  destinationNumber - The recipient's phone number.  For best
 results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).\s
                """;
```

```
        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
        String originationNumber = args[0];
        String destinationNumber = args[1];
        System.out.println("Sending a voice message");

        // Set the content type to application/json.
        List<String> listVal = new ArrayList<>();
        listVal.add("application/json");
        Map<String, List<String>> values = new HashMap<>();
        values.put("Content-Type", listVal);

        ClientOverrideConfiguration config2 = ClientOverrideConfiguration.builder()
                .headers(values)
                .build();

        PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
                .overrideConfiguration(config2)
                .region(Region.US_EAST_1)
                .build();

        sendVoiceMsg(client, originationNumber, destinationNumber);
        client.close();
    }

    public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                                    String destinationNumber) {
        try {
            SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                    .languageCode(languageCode)
                    .text(ssmlMessage)
                    .voiceId(voiceName)
                    .build();

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

            SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()
                    .destinationPhoneNumber(destinationNumber)
```

```
                    .originationPhoneNumber(originationNumber)
                    .content(content)
                    .build();

            client.sendVoiceMessage(voiceMessageRequest);
            System.out.println("The message was sent successfully.");

        } catch (PinpointSmsVoiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Pour l'exemple complet du SDK, consultez SendVoiceMessage.java on. GitHub

JavaScript (Node.js)

Utilisez cet exemple pour envoyer un message vocal 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 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";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best results,
 you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";
```

```
// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

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

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointsmsvoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
```

```
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointsmsvoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

Python

Utilisez cet exemple pour envoyer un message vocal 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 logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
```

```
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
 message.
    :param ssml_message: The content of the message. This example uses SSML to
 control
                         certain aspects of the message, such as the volume and the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
```

```
            origination_number,
            destination_number,
        )
        raise
    else:
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


if __name__ == "__main__":
    main()
```

Utiliser l'API de messagerie vocale et SMS pour les utilisateurs AWS finaux, version 2

Amazon Pinpoint inclut une API, appelée API SMS and Voice, version 2, conçue pour envoyer des SMS et des messages vocaux. Alors que l'API Amazon Pinpoint est axée sur l'envoi de messages par le biais de campagnes et de parcours planifiés et axés sur des événements, l'API SMS et voix fournit de nouvelles fonctionnalités et capacités pour envoyer des SMS et des messages vocaux directement à des destinataires individuels. Vous pouvez utiliser les API SMS et Voice indépendamment des fonctionnalités de campagne et de parcours Amazon Pinpoint, ou vous pouvez utiliser les deux en même temps pour répondre à différents cas d'utilisation. Si vous utilisez déjà Amazon Pinpoint pour envoyer des SMS ou des messages vocaux, votre compte est déjà configuré pour utiliser cette API.

Cette API est une bonne solution pour les utilisateurs dotés d'une architecture multi-locataires, tels que les fournisseurs de logiciels indépendants (ISVs). Cette API permet de garantir plus facilement que les données relatives aux événements, les numéros de téléphone d'origine et les listes de désabonnement sont séparés pour les différents locataires.

Lorsque vous utilisez l'API SMS et voix, nous vous recommandons de configurer des jeux de configuration et des destinations d'événements. L'API SMS et voix n'émet pas automatiquement de données d'événement pour les messages que vous envoyez. La configuration des destinations des événements garantit la saisie des données importantes relatives aux événements, telles que la livraison des messages et les défaillances.

La version 2 de cette API a été précédée par la version 1. Si vous utilisez actuellement la version 1 de cette API, elle restera disponible et vous pourrez continuer à l'utiliser. Si vous migrez vers la version 2, vous bénéficierez de fonctionnalités supplémentaires, telles que la possibilité de créer des groupes de numéros de téléphone, de demander de nouveaux numéros de téléphone par programmation et d'activer ou de désactiver certaines fonctionnalités des numéros de téléphone.

> **Note**
>
> Certaines tâches ne peuvent être effectuées qu'à l'aide de la console Amazon Pinpoint. Par exemple, vérifier un numéro de téléphone à utiliser lorsque votre compte est dans le sandbox SMS et vous inscrire pour utiliser 10DLC.

Pour plus d'informations sur l'API Amazon Pinpoint SMS and Voice version 2, consultez le manuel de référence de l'API SMS and Voice, version 2. Pour plus d'informations sur la création, la configuration et la gestion de vos ressources vocales et de messagerie pour les utilisateurs AWS finaux, consultez le guide de l'utilisateur de messagerie SMS pour les utilisateurs AWS finaux

Générez des mots de passe à usage unique (OTPs) avec Amazon Pinpoint

Amazon Pinpoint inclut une fonctionnalité de gestion des mots de passe à usage unique (OTP) que vous pouvez utiliser pour générer de nouveaux mots de passe à usage unique et les envoyer à vos destinataires sous forme de SMS.

> **Important**
>
> Pour utiliser cette fonctionnalité, votre compte doit disposer d'un accès à la production et d'une identité d'origine active. Pour plus d'informations, voir À propos du sandbox SMS/MMS and Voice et Demander un numéro de téléphone dans le Guide de l'utilisateur pour la messagerie SMS destinée aux utilisateurs AWS finaux.

Dans certains pays et régions, vous devez obtenir un numéro de téléphone ou un identifiant d'origine dédié avant de pouvoir envoyer des SMS. Par exemple, lorsque vous envoyez des messages à des destinataires aux États-Unis, vous devez disposer d'un numéro gratuit dédié, d'un numéro 10DLC ou d'un code abrégé. Lorsque vous envoyez des messages à des destinataires en Inde, vous devez disposer d'un ID d'expéditeur enregistré, qui comprend un ID d'entité principale (PEID) et un ID de modèle. Ces exigences s'appliquent toujours lorsque vous utilisez la fonctionnalité OTP.

Pour utiliser cette fonctionnalité, vous devez disposer d'autorisations pour envoyer et vérifier des messages OTP. Consultez One-time mots de passe. Si vous avez besoin d'aide pour déterminer les autorisations, consultezDépannage de la gestion des identités et des accès Amazon Pinpoint.

Vous pouvez utiliser l'opération `SendOtpMessages` dans l'API Amazon Pinpoint pour envoyer un code OTP à un utilisateur de votre application. Lorsque vous utilisez cette API, Amazon Pinpoint génère un code aléatoire et l'envoie à votre utilisateur sous forme de SMS. Votre demande peut inclure les paramètres suivants :

- `Channel` : le canal de communication par lequel le code OTP est envoyé. Actuellement, seuls les SMS sont pris en charge. La seule valeur acceptable est donc SMS.
- `BrandName` : le nom de la marque, de l'entreprise ou du produit associé au code OTP. Ce nom peut contenir jusqu'à 20 caractères.

> **Note**
>
> Lorsqu'Amazon Pinpoint envoie le message OTP, le nom de la marque est automatiquement inséré dans le modèle de message suivant :
>
> ```
> This is your One Time Password: {{otp}} from {{brand}}
> ```
>
> Ainsi, si vous spécifiez ExampleCorp comme nom de marque et qu'Amazon Pinpoint génère un mot de passe à usage unique 123456, le message suivant est envoyé à votre utilisateur :
>
> ```
> This is your One Time Password: 123456 from ExampleCorp
> ```

- `CodeLength` : le nombre de chiffres qui figureront dans le code OTP envoyé au destinataire. Les codes OTP peuvent contenir entre cinq et huit chiffres inclus.
- `ValidityPeriod` : la durée, en minutes, pendant laquelle le code OTP sera valide. La période de validité peut être comprise entre 5 et 60 minutes incluses.
- `AllowedAttempts` : le nombre de fois où le destinataire peut tenter en vain de vérifier l'OTP. Si le nombre de tentatives dépasse cette valeur, l'OTP devient automatiquement non valide. Le nombre maximal de tentatives autorisées est de cinq.
- `Language` : la langue, au format IETF BCP-47, à utiliser lors de l'envoi du message. Les valeurs acceptables sont :
  - `de-DE` : allemand
  - `en-GB` : anglais (Royaume-Uni)
  - `en-US` : anglais (États-Unis)
  - `es-419` : espagnol (Amérique latine)
  - `es-ES` : espagnol
  - `fr-CA` : français (Canada)
  - `fr-FR` : français
  - `it-IT` : italien
  - `ja-JP` : japonais
  - `ko-KR` : coréen
  - `pt-BR` : portugais (Brésil)

  - `zh-CN` : chinois (simplifié)
  - `zh-TW` : chinois (traditionnel)
- `OriginationIdentity` : l'identité d'origine (telle qu'un code long, un code court ou un ID d'expéditeur) utilisée pour envoyer le code OTP. Si vous utilisez un code long ou un numéro gratuit pour envoyer l'OTP, le numéro de téléphone doit être au format E.164.
- `DestinationIdentity` : le numéro de téléphone au format E.164 auquel le code OTP a été envoyé.
- `ReferenceId` : un ID de référence unique pour la demande. L'ID de référence correspond exactement à l'ID de référence que vous fournissez lorsque vous vérifiez l'OTP. L'ID de référence peut contenir entre 1 et 48 caractères inclus.
- `EntityId` : un ID d'entité enregistré auprès d'une agence de réglementation. Ce paramètre est actuellement utilisé uniquement lors de l'envoi de messages à des destinataires en Inde. Si vous n'envoyez pas de messages à des destinataires en Inde, vous pouvez omettre ce paramètre.
- `TemplateId` : un ID de modèle enregistré auprès d'une agence de réglementation. Ce paramètre est actuellement utilisé uniquement lors de l'envoi de messages à des destinataires en Inde. Si vous n'envoyez pas de messages à des destinataires en Inde, vous pouvez omettre ce paramètre.

> **Note**
>
> Pour plus d'informations sur les exigences relatives à l'envoi de messages à des destinataires en Inde, consultez le processus d'enregistrement de l'identifiant d'expéditeur indien dans le guide de l'utilisateur Amazon Pinpoint.

Pour vous assurer que votre compte Amazon Pinpoint est correctement configuré pour envoyer des messages OTP, vous pouvez utiliser le AWS CLI pour envoyer un message test. Pour plus d'informations à ce sujet AWS CLI, consultez le guide de AWS Command Line Interface l'utilisateur.

Pour envoyer un message OTP de test à l'aide du AWS CLI, exécutez la send-otp-messagecommande dans le terminal :

```
aws pinpoint send-otp-message --application-id 7353f53e6885409fa32d07cedexample --send-
otp-message-request-parameters
 Channel=SMS,BrandName=ExampleCorp,CodeLength=5,ValidityPeriod=20,AllowedAttempts=5,Origination
```

Dans la commande précédente, procédez comme suit :

- Remplacez *7353f53e6885409fa32d07cedexample* par l'identifiant de votre application.
- Remplacez *ExampleCorp* par le nom de votre entreprise.
- Remplacez `CodeLegth` par *5* le nombre de chiffres qui figureront dans le code OTP envoyé au destinataire.
- Remplacez par *20* la durée, en minutes, pendant laquelle le code OTP sera valide. `ValidityPeriod`
- Remplacez `AllowedAttempts` par *5* le nombre de fois où le destinataire peut tenter sans succès de vérifier l'OTP.
- *+18555550142*`OriginationIdentity`Remplacez-le par l'identité d'origine utilisée pour envoyer le code OTP.
- Remplacez `DestinationIdentity` par *+12065550007* le numéro de téléphone auquel envoyer le code OTP.
- *SampleReferenceId*`ReferenceId`Remplacez-le par un numéro de référence unique pour la demande.

Réponse de **`SendOtpMessage`**

Lorsque vous parvenez à envoyer un message OTP, vous recevez une réponse similaire à l'exemple suivant :

```
{
    "MessageResponse": {
        "ApplicationId": "7353f53e6885409fa32d07cedexample",
        "RequestId": "255d15ea-75fe-4040-b919-096f2example",
        "Result": {
            "+12065550007": {
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "nvrmgq9kq4en96qgp0tlqli3og1at6aexample",
                "StatusCode": 200,
                "StatusMessage": "MessageId: nvrmgq9kq4en96qgp0tlqli3og1at6aexample"
            }
        }
    }
}
```

Valider les messages OTP dans Amazon Pinpoint

Après avoir envoyé un one-time-password, votre application peut appeler l'API Amazon Pinpoint pour le vérifier. Pour vérifier un code OTP, appelez l'API `VerifyOtpMessages`. Votre demande doit inclure les paramètres suivants :

- `DestinationIdentity` : le numéro de téléphone au format E.164 auquel le code OTP a été envoyé.
- `ReferenceId` : l'ID de référence que vous avez utilisé lorsque vous avez envoyé le code OTP au destinataire. L'ID de référence doit correspondre exactement.
- `Otp` : le code OTP que vous êtes en train de valider.

Vous pouvez utiliser le AWS CLI pour tester le processus de validation. Pour plus d'informations sur l'installation et la configuration du AWS CLI, consultez le guide de AWS Command Line Interface l'utilisateur.

Pour vérifier un OTP à l'aide du AWS CLI, exécutez la verify-otp-messagecommande dans le terminal :

```
aws pinpoint verify-otp-message --application-id 7353f53e6885409fa32d07cedexample --
verify-otp-message-request-parameters
 DestinationIdentity=+12065550007,ReferenceId=SampleReferenceId,Otp=01234
```

Dans la commande précédente, procédez comme suit :

- Remplacez *7353f53e6885409fa32d07cedexample* par l'identifiant de votre application.
- *+12065550007*Remplacez-le `DestinationIdentity` par le numéro de téléphone auquel le code OTP a été envoyé.
- *SampleReferenceId*`ReferenceId`Remplacez-le par un numéro de référence unique pour la demande. Cette valeur doit correspondre à celle `ReferenceID` qui a été utilisée pour envoyer la demande.
- *01234*`Otp`Remplacez-le par un Otp envoyé au`DestinationIdentity`.

Réponse de **`VerifyOtpMessage`**

Lorsque vous envoyez une demande à l'API `VerifyOTPMessage`, celle-ci renvoie un objet `VerificationResponse` contenant une seule propriété, `Valid`. Si l'ID de référence, le numéro de

téléphone et l'OTP correspondent tous aux valeurs attendues par Amazon Pinpoint, et si l'OTP n'a pas expiré, la valeur de `Valid` est `true` ; sinon, elle est `false`. Voici un exemple de réponse à une vérification OTP réussie :

```
{
    "VerificationResponse": {
        "Valid": true
    }
}
```

Exemples de code OTP pour l'utilisation du SDK pour Python (Boto3) dans Amazon Pinpoint

Cette section contient des exemples de code qui montrent comment utiliser le kit SDK for Python (Boto3) pour envoyer et vérifier des codes OTP.

Générer un ID de référence

La fonction suivante génère un ID de référence unique pour chaque destinataire, en fonction du numéro de téléphone du destinataire, du produit ou de la marque pour lequel le destinataire reçoit un OTP et de la source de la demande (qui peut être le nom d'une page d'un site ou d'une application, par exemple). Lorsque vous vérifiez le code OTP, vous devez transmettre un ID de référence identique pour que la validation réussisse. Les exemples de code d'envoi et de validation utilisent cette fonction utilitaire.

Cette fonction n'est pas obligatoire, mais elle constitue un moyen utile d'étendre le processus d'envoi et de vérification OTP à une transaction spécifique de manière à pouvoir la soumettre à nouveau facilement lors de l'étape de vérification. Vous pouvez utiliser l'ID de référence de votre choix. Il ne s'agit que d'un exemple de base. Cependant, les autres exemples de code présentés dans cette section s'appuient sur cette fonction.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import hashlib

def generate_ref_id(destinationNumber,brandName,source):
    refId = brandName + source + destinationNumber
```

```
    return hashlib.md5(refId.encode()).hexdigest()
```

Envoyer des codes OTP

L'exemple de code suivant montre comment utiliser le kit SDK pour Python (Boto3) pour l'envoi d'un code OTP.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

### Some variables that are unlikely to change from request to request. ###

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The phone number or short code to send the message from.
originationNumber = "+18555550142"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# The number of times the user can unsuccessfully enter the OTP code before it becomes
 invalid.
allowedAttempts = 3

# Function that sends the OTP as an SMS message.
def send_otp(destinationNumber,codeLength,validityPeriod,brandName,source,language):
    client = boto3.client('pinpoint',region_name=region)
    try:
        response = client.send_otp_message(
            ApplicationId=appId,
            SendOTPMessageRequestParameters={
                'Channel': 'SMS',
                'BrandName': brandName,
                'CodeLength': codeLength,
                'ValidityPeriod': validityPeriod,
                'AllowedAttempts': allowedAttempts,
                'Language': language,
                'OriginationIdentity': originationNumber,
```

```
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source)
            }
        )

    except ClientError as e:
        print(e.response)
    else:
        print(response)

# Send a message to +14255550142 that contains a 6-digit OTP that is valid for 15
 minutes. The
# message will include the brand name "ExampleCorp", and the request originated from a
 part of your
# site or application called "CreateAccount". The US English message template should be
 used to
# send the message.
send_otp("+14255550142",6,15,"ExampleCorp","CreateAccount","en-US")
```

Valider les codes OTP

L'exemple de code suivant vous montre comment utiliser le kit SDK pour Python (Boto3) pour vérifier un code OTP que vous avez déjà envoyé. Pour que l'étape de validation réussisse, votre demande doit inclure un numéro de référence qui correspond exactement à l'ID de référence utilisé pour envoyer le message.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# Function that verifies the OTP code.
def verify_otp(destinationNumber,otp,brandName,source):
    client = boto3.client('pinpoint',region_name=region)
    try:
```

```
        response = client.verify_otp_message(
            ApplicationId=appId,
            VerifyOTPMessageRequestParameters={
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source),
                'Otp': otp
            }
        )

    except ClientError as e:
        print(e.response)
    else:
        print(response)

# Verify the OTP 012345, which was sent to +14255550142. The brand name ("ExampleCorp")
 and the
# source name ("CreateAccount") are used to generate the correct reference ID.
verify_otp("+14255550142","012345","ExampleCorp","CreateAccount")
```

Personnalisez les messages intégrés à l'application avec Amazon Pinpoint et Amplify

Vous pouvez utiliser les messages intégrés à l’application pour envoyer des messages ciblés aux utilisateurs de vos applications. Les messages intégrés à l’application sont hautement personnalisables. Ils peuvent inclure des boutons qui ouvrent des sites Web ou redirigent les utilisateurs vers des parties spécifiques de votre application. Vous pouvez configurer les couleurs d’arrière-plan et de texte, positionner le texte et ajouter des boutons et des images à la notification. Vous pouvez envoyer un seul message ou créer un carrousel contenant jusqu'à cinq messages uniques. Pour obtenir un aperçu des messages intégrés à l'application, y compris les instructions relatives à la création de modèles de messages intégrés, consultez la section Création de modèles dans l'application dans le Guide de l'utilisateur Amazon Pinpoint.

Vous pouvez l'utiliser AWS Amplify pour intégrer facilement les fonctionnalités de messagerie intégrées à l'application d'Amazon Pinpoint dans votre application. Amplify peut gérer automatiquement les processus de récupération des messages, de rendu des messages et d’envoi de données analytiques à Amazon Pinpoint. Cette intégration est actuellement prise en charge pour les applications React Native. Pour plus d’informations, consultez Messagerie dans l’application dans la documentation du Cadre Amplify.

Récupérez les messages intégrés à l'application pour un point de terminaison par programmation à l'aide d'Amazon Pinpoint

Vos applications peuvent appeler l'GetInAppMessagesAPI pour récupérer tous les messages intégrés auxquels un point de terminaison donné a droit. Lorsque vous appelez l'API `GetInAppMessages`, vous fournissez les paramètres suivants :

- `ApplicationId` : ID du projet Amazon Pinpoint unique à laquelle la campagne dans l'application est associée.
- `EndpointId` : ID unique du point de terminaison pour lequel vous récupérez des messages.

Lorsque vous appelez l'API avec ces valeurs, elle renvoie une liste de messages. Pour plus d'informations sur la réponse produite par cette opération, consultez `GetInAppMessages`Exemple JSON de réponse à l'API Amazon Pinpoint.

Vous pouvez utiliser le AWS SDKs pour appeler l'`GetInAppMessages`opération. Les exemples de code suivants incluent des fonctions permettant de récupérer des messages intégrés à l'application.

JavaScript

Créez le client dans un module séparé et exportez-le :

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
const REGION = "us-east-1";
const pinClient = new PinpointClient({ region: REGION });
export { pinClient };
```

Récupérez les messages dans l'application pour un point de terminaison :

```
// Import required AWS SDK clients and commands for Node.js
import { PinpointClient, GetInAppMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./lib/pinClient.js";

("use strict");

//The Amazon Pinpoint application ID.
const projectId = "4c545b28d21a490cb51b0b364example";

//The ID of the endpoint to retrieve messages for.
const endpointId = "c5ac671ef67ee3ad164cf7706example";

const params = {
  ApplicationId: projectId,
  EndpointId: endpointId
};

const run = async () => {
  try {
    const data = await pinClient.send(new GetInAppMessagesCommand(params));
    console.log(JSON.stringify(data, null, 4));
    return data;
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Python

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

logger = logging.getLogger(__name__)

def retrieve_inapp_messages(
            pinpoint_client, project_id, endpoint_id):
    """
    Retrieves the in-app messages that a given endpoint is entitled to.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project ID.
    :param endpoint_id: The ID of the endpoint to retrieve messages for.
    :return: A JSON object that contains information about the in-app message.
    """

    try:
        response = pinpoint_client.get_in_app_messages(
            ApplicationId=project_id,
            EndpointId=endpoint_id)
    except ClientError:
        logger.exception("Couldn't retrieve messages.")
        raise
    else:
        return response

def main():
    project_id = "4c545b28d21a490cb51b0b364example"
    endpoint_id = "c5ac671ef67ee3ad164cf7706example"
    inapp_response = retrieve_inapp_messages(
        boto3.client('pinpoint'), project_id, endpoint_id)
    print(inapp_response)

if __name__ == '__main__':
    main()
```

GetInAppMessagesExemple JSON de réponse à l'API Amazon Pinpoint

Lorsque vous appelez l'opération GetInAppMessagesAPI, elle renvoie une liste de messages auxquels le point de terminaison spécifié a droit. Votre application peut ensuite afficher le message en fonction des valeurs de la réponse.

Voici un exemple d'objet JSON renvoyé lorsque vous appelez l'API `GetInAppMessages` :

```
{
  "InAppMessagesResponse":{
    "InAppMessageCampaigns":[
      {
        "CampaignId":"inAppTestCampaign-4c545b28d21a490cb51b0b364example",
        "DailyCap":0,
        "InAppMessage":{
          "Content":[
            {
              "BackgroundColor":"#f8e71c",
              "BodyConfig":{
                "Alignment":"CENTER",
                "Body":"This is a sample in-app message sent using Amazon Pinpoint.",
                "TextColor":"#d0021b"
              },
              "HeaderConfig":{
                "Alignment":"CENTER",
                "Header":"Sample In-App Message",
                "TextColor":"#d0021b"
              },
              "ImageUrl":"https://example.com/images/thumbnail.png",
              "PrimaryBtn":{
                "DefaultConfig":{
                  "BackgroundColor":"#d0021b",
                  "BorderRadius":50,
                  "ButtonAction":"CLOSE",
                  "Text":"Dismiss",
                  "TextColor":"#f8e71c"
                }
              }
            }
          ],
          "Layout":"MIDDLE_BANNER"
```

```
        },
        "Priority":3,
        "Schedule":{
          "EndDate":"2021-11-06T00:08:05Z",
          "EventFilter":{
            "Dimensions":{
              "Attributes":{

              },
              "EventType":{
                "DimensionType":"INCLUSIVE",
                "Values":[
                  "_session.start"
                ]
              },
              "Metrics":{

              }
            }
          }
        },
        "SessionCap":0,
        "TotalCap":0,
        "TreatmentId":"0"
      }
    ]
  }
}
```

Les sections suivantes fournissent des informations sur les composants de cette réponse et leurs attributs.

Objet **InAppMessageCampaigns**

L'objet `InAppMessageCampaigns` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`CampaignId`	Chaîne contenant le nom et l'ID de campagne unique de la campagne Amazon Pinpoint à partir de laquelle le message	Créé automatiquement par Amazon Pinpoint lorsque vous créez la campagne.

Attribut	Description	Emplacement d'installation
	a été envoyé. Nom précède l'ID de campagne. Les deux valeurs sont séparées par un tiret (-).	
`TreatmentId`	Nombre entier qui représente l'ID du traitement de campagne pour ce message. Si la campagne ne comporte qu'un seul traitement, la valeur est `0`.	
`Priority`	La priorité du message intégré, exprimée sous la forme d'un nombre entier compris entre 1 et 5, inclus, où 1 indique la priorité la plus élevée et 5 la priorité la plus faible.	Étape 1 du processus de création de campagne.
`InAppMessage`	Un Objet `InAppMessage` qui contient des informations sur la façon dont le message est rendu.	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.
`Schedule`	Un objet Schedule qui contient des informations sur la date d'envoi du message.	Étape 4 du processus de création de campagne (si la campagne a été créée dans la console) ou de l'objet `Schedule` (si la campagne a été créée à l'aide de l'API ou d'un kit SDK).

Attribut	Description	Emplacement d'installation
`DailyCap`	Le nombre de fois, sous forme de nombre entier, qu'un message dans l'application peut être affiché à l'utilisateur sur une période de 24 heures.	Hérité des paramètres au niveau du projet. Si la campagne inclut des paramètres qui remplacent ceux du projet, ceux-ci sont utilisés à la place.
`SessionCap`	Le nombre de fois, exprimé sous forme de nombre entier, qu'un message intégré à l'application peut être affiché à l'utilisateur au cours d'une session d'application.	
`TotalCap`	Le nombre total de fois, exprimé sous forme de nombre entier, qu'un message intégré à l'application peut être affiché à un point de terminaison par campagne.	

Objet **`InAppMessage`**

L'objet `InAppMessage` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`Content`	Tableau contenant un InAppMessageContentobjet qui décrit le contenu du message.	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.
`Layout`	Une chaîne qui décrit la façon dont le message intégré à l'application s'affichera sur	

Attribut	Description	Emplacement d'installation
	l'appareil du destinataire. Les valeurs possibles sont : • `BOTTOM_BANNER` : un message qui s'affiche sous la forme d'une bannière en bas de la page. • `TOP_BANNER` : un message qui s'affiche sous la forme d'une bannière en haut de la page. • `OVERLAYS` : un message qui couvre la totalité de l'écran. • `MOBILE_FEED` : un message qui s'affiche dans une fenêtre en face de la page. • `MIDDLE_BANNER` : un message qui s'affiche sous forme de bannière au milieu de la page. • `CAROUSEL` : une mise en page déroulante contenant jusqu'à cinq messages uniques.	

Objet **`HeaderConfig`**

L'objet `HeaderConfig` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`Alignment`	Chaîne qui indique l'alignement du texte de l'en-tête. Les	Sur la base du contenu du modèle de message dans

Attribut	Description	Emplacement d'installation
	valeurs possibles sont `LEFT`, `CENTER` et `RIGHT`.	l'application spécifié pour la campagne.
`Header`	Texte de l'en-tête du message.	
`TextColor`	La couleur du texte de l'en-tête, exprimée sous la forme d'une chaîne décrivant le code couleur hexadécimal (tel que « #000000 » pour le noir).	

Objet **`BodyConfig`**

L'objet `BodyConfig` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`Alignment`	Chaîne qui spécifie l'alignement du texte du corps du message. Les valeurs possibles sont `LEFT`, `CENTER` et `RIGHT`.	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.
`Body`	Le corps du texte principal du message.	
`TextColor`	La couleur du corps du texte, exprimée sous la forme d'une chaîne composée d'un code couleur hexadécimal (par exemple : « #000000 » pour le noir).	

Objet **InAppMessageContent**

L'objet `InAppMessageContent` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`BackgroundColor`	La couleur d'arrière-plan du message dans l'application, exprimée sous la forme d'une chaîne contenant un code de couleur hexadécimal (tel que « #000000 » pour le noir).	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.
`BodyConfig`	BodyConfigObjet contenant des informations relatives au contenu principal du message.	
`HeaderConfig`	HeaderConfigObjet contenant des informations relatives à l'en-tête ou au titre du message.	
`ImageUrl`	L'URL de l'image qui apparaît dans le message.	
`PrimaryBtn`	InAppMessageButtonObjet contenant des informations sur le bouton principal du message.	
`SecondaryBtn`	InAppMessageButtonObjet contenant des informations sur le bouton secondaire du message. Absent si le modèle de message intégré à l'application ne spécifie pas de bouton secondaire.	

Objet **Schedule**

L'objet `Schedule` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`EndDate`	Heure planifiée, au format ISO 8601, à laquelle la campagne se terminera.	Étape 4 du processus de création de campagne (si la campagne a été créée dans la console) ou de l'objet `Schedule` (si la campagne a été créée à l'aide de l'API ou d'un kit SDK).
`EventFilter`	Informations sur l'événement à l'origine de l'affichage du message dans l'application. Lorsque vous générez un événement correspondant à une campagne intégrée à l'application Amazon Pinpoint, le message s'affiche.	

Objet **InAppMessageButton**

Un objet `InAppMessageButton` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`DefaultConfig`	DefaultButtonConfigObjet contenant des informations sur les paramètres par défaut d'un bouton dans un message intégré à l'application.	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.
`Android`	OverrideButtonConfigObjet qui indique le comportement du bouton sur les appareils Android. Cette option remplace la configuration par défaut des boutons décrite	

Attribut	Description	Emplacement d'installation
	dans l'objet `DefaultConfig` .	
`IOS`	OverrideButtonConfigObjet qui indique le comportement du bouton sur les appareils iOS. Cette option remplace la configuration par défaut des boutons décrite dans l'objet `DefaultConfig` .	
`Web`	OverrideButtonConfigObjet qui indique le comportement du bouton dans les applications Web. Cette option remplace la configuration par défaut des boutons décrite dans l'objet `DefaultConfig` .	

Objet **`DefaultButtonConfig`**

Un objet `DefaultButtonConfig` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`BackgroundColor`	La couleur d'arrière-plan d'un bouton, exprimée sous la forme d'une chaîne contenant un code couleur hexadécimal (par exemple : « #000000 » pour le noir).	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.
`BorderRadius`	Rayon de la bordure du bouton en pixels, exprimé sous forme de nombre entier.	

Attribut	Description	Emplacement d'installation
	Plus le nombre est élevé, plus les coins sont arrondis.	
`ButtonAction`	Chaîne décrivant l'action qui se produit lorsque le destinataire choisit un bouton dans le message intégré à l'application. Les valeurs possibles sont : • `LINK` : un lien vers une destination web. • `DEEP_LINK` : un lien vers une page spécifique d'une application. • `CLOSE` : ignore le message.	
`Link`	L'URL de destination d'un bouton. Non présent pour les boutons où se ButtonAction trouve le`CLOSE`.	
`Text`	Le texte qui apparaît sur le bouton.	
`TextColor`	La couleur du texte du bouton, exprimée sous la forme d'une chaîne contenant un code couleur hexadécimal (par exemple : « #000000 » pour le noir).	

Objet `OverrideButtonConfig`

L'objet `OverrideButtonConfig` n'est présent que si le modèle de message intégré à l'application utilise des boutons de remplacement. Un bouton de remplacement est un bouton dont la configuration est spécifique pour un type d'appareil particulier, tel qu'un appareil iOS, un appareil Android ou un navigateur Web.

Un objet `OverrideButtonConfig` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
`ButtonAction`	L'action qui se produit lorsqu'un destinataire choisit un bouton dans le message intégré à l'application. Les valeurs possibles sont : • `LINK` : un lien vers une destination web. • `DEEP_LINK` : un lien vers une page spécifique d'une application. • `CLOSE` : ignore le message.	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.
`Link`	L'URL de destination d'un bouton. Non présent pour les boutons dont la valeur `ButtonAction` est `CLOSE`.	
`Text`	Le texte qui apparaît sur le bouton.	
`TextColor`	La couleur du texte du bouton, exprimée sous la forme d'une chaîne contenant un code couleur hexadécimal (par exemple : « #000000 » pour le noir).	

Utilisez le service de validation du numéro de téléphone Amazon Pinpoint

Amazon Pinpoint intègre un service de validation de numéro de téléphone, que vous pouvez utiliser pour déterminer la validité d'un numéro, ainsi que pour obtenir des informations supplémentaires sur le numéro de téléphone lui-même. Par exemple, lorsque vous utilisez le service de validation de numéro de téléphone, il renvoie les informations suivantes :

- Le numéro de téléphone au format E.164.
- Le type de numéro de téléphone (mobile, fixe, VoIP, etc.).
- La ville et le pays d'origine du numéro de téléphone.
- Le fournisseur de services associé au numéro de téléphone.

Des frais supplémentaires sont facturés pour l'utilisation du service de validation de numéro de téléphone. Pour plus d'informations, consultez Tarification d'Amazon Pinpoint.

> **Important**
>
> Pour les numéros de téléphone provenant des États-Unis et du Canada, l'API de validation des numéros de téléphone ne renverra plus de données pour `City`, `County`, `Timezone` et `ZipCode`.

Cas d'utilisation de la validation des numéros de téléphone Amazon Pinpoint

Vous pouvez utiliser le service de validation de numéro de téléphone dans plusieurs cas d'utilisation, parmi lesquels :

- Vérification de numéros de téléphone fournis sur un formulaire web : si vous utilisez des formulaires web pour collecter des informations de contact pour vos clients, vous validez les numéros de téléphone fournis par les clients avant de soumettre le formulaire. Utilisez le backend de votre site web pour valider le numéro à l'aide de l'API Amazon Pinpoint. La réponse de l'API indique si le numéro n'est pas valide, par exemple si le numéro de téléphone est mal formaté. S'il

s'avère que le numéro de téléphone fourni par le client n'est pas valide, votre formulaire web peut demander au client de fournir un autre numéro.

- Nettoyage de votre base de contact existante : si vous disposez d'une base de données répertoriant les numéros de téléphone des clients, vous pouvez valider chaque numéro de téléphone, puis mettre à jour votre base de données en fonction des résultats. Par exemple, si vous détectez des points de terminaison dont les numéros de téléphone ne sont pas en mesure de recevoir des SMS, vous pouvez modifier la propriété `ChannelType` pour le point de terminaison de `SMS` sur `VOICE`. Vous pouvez d'abord valider le numéro de téléphone, puis mettre à jour la propriété `ChannelType` pour les points de terminaison nouveaux ou existants en suivant les instructions dans Ajouter des points de terminaison à Amazon Pinpoint pour un seul point de terminaison ou Ajouter un lot de points de terminaison à Amazon Pinpoint pour plusieurs points de terminaison.
- Choix du canal adapté avant d'envoyer un message : si vous prévoyez d'envoyer un SMS, mais que vous déterminez que le numéro de destination n'est pas valide, vous pouvez envoyer le message au destinataire via un autre canal. Par exemple, si le point de terminaison n'est pas en mesure de recevoir des SMS, vous pouvez envoyer un message vocal.

Validez un numéro de téléphone à l'aide du AWS CLI

L'exemple suivant montre comment valider un numéro de téléphone à l'aide du AWS CLI. Pour plus d'informations, consultez phone-number-validate dans la Référence des commandes de l'AWS CLI . Pour des exemples de réponses de validation, voirRéponse de validation du numéro de téléphone. Pour plus d'informations sur la configuration du AWS CLI, voir Configurer le AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser le service de validation des numéros de téléphone à l'aide du AWS CLI

- Sur la ligne de commande, entrez la commande suivante :

```
aws pinpoint phone-number-validate --number-validate-request
 PhoneNumber=+442079460881,IsoCountryCode=GB
```

Dans la commande précédente, remplacez *+442079460881* par le numéro de téléphone que vous souhaitez valider et *GB* par le code ISO à deux chiffres du pays ou de la région.

> **Note**
>
> Lorsque vous fournissez un numéro de téléphone au service de validation de numéro de téléphone, vous devez toujours inclure le code pays. Si vous n'incluez pas le code pays, le service peut renvoyer des informations relatives à un numéro de téléphone basé dans un autre pays. Vous pouvez avoir des tirets dans le numéro de téléphone, par exemple*+44-207-946-0881*.

Réponse de validation du numéro de téléphone

Les informations renvoyées par le service de validation des numéros de téléphone varient légèrement en fonction des données disponibles pour le numéro de téléphone fourni. Cette section contient des exemples de réponses renvoyées par le service de validation des numéros de téléphone.

> **Note**
>
> Les données fournies par le service de validation de numéro de téléphone reposent sur des informations fournies par des opérateurs de télécommunications et d'autres entités dans le monde entier. Dans certains pays, les fournisseurs peuvent mettre à jour ces informations moins fréquemment que dans d'autres pays. Par exemple, si vous envoyez une requête pour valider un numéro de téléphone mobile, et que le numéro que vous avez fourni a été transmis d'un opérateur mobile à un autre, la réponse du service de validation de numéro de téléphone peut inclure le nom de l'opérateur d'origine, plutôt que celui de l'opérateur actuel.

Numéros de téléphone mobile valides

Lorsque vous envoyez une requête au service de validation des numéros de téléphone et que le numéro est un numéro de téléphone mobile valide, il renvoie des informations semblables à l'exemple suivant :

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp Mobile",
        "City": "Seattle",
```

```
        "CleansedPhoneNumberE164": "+12065550142",
        "CleansedPhoneNumberNational": "2065550142",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+12065550142",
        "PhoneType": "MOBILE",
        "PhoneTypeCode": 0,
        "Timezone": "America/Los_Angeles",
        "ZipCode": "98101"
    }
}
```

Numéros de téléphone fixe valides

Si votre requête contient un numéro de téléphone fixe valide, le service de validation des numéros de téléphone renvoie des informations semblables à l'exemple suivant :

```
{
    "CountryCodeIso2": "US",
    "CountryCodeNumeric": "1",
    "Country": "United States",
    "City": "Santa Clara",
    "ZipCode": "95037",
    "Timezone": "America/Los_Angeles",
    "CleansedPhoneNumberNational": "4085550101",
    "CleansedPhoneNumberE164": "14085550101",
    "Carrier": "AnyCompany",
    "PhoneTypeCode": 1,
    "PhoneType": "LANDLINE",
    "OriginalPhoneNumber": "+14085550101"
}
```

Numéros de téléphone VoIP valides

Si votre requête contient un numéro de téléphone VoIP (voix sur IP) valide, le service de validation des numéros de téléphone renvoie des informations semblables à l'exemple suivant :

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
```

```
        "CleansedPhoneNumberE164": "+441514960001",
        "CleansedPhoneNumberNational": "1514960001",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+441514960001",
        "PhoneType": "VOIP",
        "PhoneTypeCode": 2
    }
}
```

Numéros de téléphone non valides

Si votre requête contient un numéro de téléphone mobile non valide, le service de validation des numéros de téléphone renvoie des informations semblables à l'exemple suivant :

```
{
    "NumberValidateResponse": {
        "CleansedPhoneNumberE164": "+44163296076",
        "CleansedPhoneNumberNational": "163296076",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+440163296076",
        "PhoneType": "INVALID",
        "PhoneTypeCode": 3
    }
}
```

Notez que dans cette réponse, la propriété `PhoneType`indique que ce numéro de téléphone est `INVALID`. En outre, elle n'inclut pas d'informations sur l'opérateur ou l'emplacement associé au numéro de téléphone. Vous devez éviter d'envoyer des SMS ou des messages vocaux aux numéros de téléphone dont le `PhoneType` est `INVALID`, car ces numéros n'appartiennent probablement pas aux destinataires réels.

Autres numéros de téléphone

Parfois, la réponse du service de validation de numéro de téléphone comprend une propriété `PhoneType` dont la valeur est `OTHER`. Le service peut retourner ce type de réponse dans les situations suivantes :

- Le numéro de téléphone est un numéro d'appel gratuit.

- Le numéro de téléphone est réservé à une utilisation dans des émissions télévisées et des films, comme les numéros de téléphone nord-américains qui commencent par 555.
- Le numéro de téléphone contient un indicatif qui n'est pas actuellement utilisé (par exemple, l'indicatif 999 en Amérique du Nord).
- Le numéro de téléphone est réservé à d'autres fins.

L'exemple suivant illustre la réponse renvoyée par le service de validation de numéro de téléphone lorsque votre requête contient un numéro de téléphone nord-américain fictif :

```
{
    "NumberValidateResponse": {
        "Carrier": "Multiple OCN Listing",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "OTHER",
        "PhoneTypeCode": 4,
        "Timezone": "America/Los_Angeles"
    }
}
```

Numéros de téléphone prépayés

Si votre requête contient un numéro de téléphone prépayé valide, le service de validation des numéros de téléphone renvoie des informations semblables à l'exemple suivant :

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "PREPAID",
        "PhoneTypeCode": 5
```

```
    }
}
```

Pour plus d'informations sur les informations contenues dans ces réponses, consultez Validation des numéros de téléphone dans la Référence de l'API Amazon Pinpoint.

Créez une chaîne personnalisée dans Amazon Pinpoint à l'aide d'un webhook ou d'une fonction Lambda

Amazon Pinpoint inclut la prise en charge intégrée de l'envoi de messages via les canaux de notification push, e-mail, SMS et voix. Vous pouvez également configurer Amazon Pinpoint pour l'envoi de messages à travers d'autres canaux en créant des canaux personnalisés. Les canaux personnalisés dans Amazon Pinpoint vous permettent d'envoyer des messages à travers n'importe quel service possédant une API, y compris des services tiers. Vous pouvez interagir avec APIs en utilisant un webhook ou en appelant une AWS Lambda fonction.

Les segments vers lesquels vous envoyez des campagnes de canaux personnalisées peuvent contenir des points de terminaison de tous les types (c'est-à-dire des points de terminaison dont la valeur de l'attribut `ChannelType` est EMAIL, VOICE, SMS, CUSTOM ou l'un des différents types de point de terminaison de notification push).

Utiliser un webhook

Si vous utilisez un webhook pour envoyer des messages de chaîne personnalisés, l'URL du webhook doit commencer par « https://". The webhook URL can only contain alphanumeric characters, plus the following symbols: hyphen (-), period (.), underscore (_), tilde (~), question mark (?), slash or solidus (/), pound or hash sign (#), and semicolon (:). The URL has to com ply with ». RFC3986

Lorsque vous créez une campagne qui spécifie une URL de webhook, Amazon Pinpoint émet un HTTP `HEAD` vers cette URL. La réponse à la demande `HEAD` doit contenir un en-tête appelé `X-Amz-Pinpoint-AccountId`. La valeur de cet en-tête doit être égale à votre identifiant de AWS compte.

Utiliser une fonction Lambda

Si vous préférez envoyer des messages de chaîne personnalisés en créant une fonction Lambda, il est préférable de vous familiariser d'abord avec les données émises par Amazon Pinpoint. Lorsqu'une campagne Amazon Pinpoint envoie des messages à travers un canal personnalisé, elle envoie une charge utile à la fonction Lambda cible semblable à l'exemple suivant :

```
{
  "Message":{},
  "Data":"The payload that's provided in the CustomMessage object in
 MessageConfiguration",
```

```
  "ApplicationId":"3a9b1f4e6c764ba7b031e7183example",
  "CampaignId":"13978104ce5d6017c72552257example",
  "TreatmentId":"0",
  "ActivityId":"575cb1929d5ba43e87e2478eeexample",
  "ScheduledTime":"2020-04-08T19:00:16.843Z",
  "Endpoints":{
    "1dbcd396df28ac6cf8c1c2b7fexample":{
      "ChannelType":"EMAIL",
      "Address":"mary.major@example.com",
      "EndpointStatus":"ACTIVE",
      "OptOut":"NONE",
      "Location":{
        "City":"Seattle",
        "Country":"USA"
      },
      "Demographic":{
        "Make":"OnePlus",
        "Platform":"android"
      },
      "EffectiveDate":"2020-04-01T01:05:17.267Z",
      "Attributes":{
        "CohortId":[
          "42"
        ]
      },
      "CreationDate":"2020-04-01T01:05:17.267Z"
    }
  }
}
```

Les données d'événements fournissent les attributs suivants :

- `ApplicationId` : ID du projet Amazon Pinpoint auquel la campagne appartient.
- `CampaignId` : ID de la campagne Amazon Pinpoint pour laquelle la fonction Lambda est invoquée.
- `TreatmentId` : ID de la variante de campagne. Si vous avez créé une campagne standard, cette valeur est toujours 0. Si vous avez créé une campagne de A/B test, cette valeur est un entier compris entre 0 et 4.
- `ActivityId` : ID de l'activité effectuée par la campagne.
- `ScheduledTime` : heure à laquelle Amazon Pinpoint exécute la campagne, affichée au format ISO 8601.

- `Endpoints` : liste des points de terminaison ciblés par la campagne. Chaque charge utile peut contenir jusqu'à 50 points de terminaison. Si le segment vers lequel la campagne a été envoyée contient plus de 50 points de terminaison, Amazon Pinpoint appelle la fonction à plusieurs reprises, avec un maximum de 50 points de terminaison à la fois, jusqu'à ce que tous les points de terminaison aient été traités.

Vous pouvez utiliser cet échantillon de données lors de la création et du test de votre fonction Lambda de canal personnalisé.

Attribuez une fonction Lambda ou un webhook à une campagne individuelle à l'aide de l'API Amazon Pinpoint

Pour attribuer une fonction Lambda ou un webhook à une campagne individuelle, utilisez l'API Amazon Pinpoint pour créer ou mettre à jour un objet de campagne.

L'objet `MessageConfiguration` de la campagne doit également contenir un objet `CustomMessage`. Cet objet possède un membre :`Data`. La valeur de `Data` est une chaîne JSON qui contient la charge utile du message que vous souhaitez envoyer au canal personnalisé.

La campagne doit contenir un objet `CustomDeliveryConfiguration`. Dans l'objet `CustomDeliveryConfiguration`, spécifiez ce qui suit :

- `EndpointTypes` : tableau contenant tous les types de point de terminaison auxquels la campagne de canal personnalisé doit être envoyée. Il peut contenir un tous les types de canaux suivants :
  - `ADM`
  - `APNS`
  - `APNS_SANDBOX`
  - `APNS_VOIP`
  - `APNS_VOIP_SANDBOX`
  - `BAIDU`
  - `CUSTOM`
  - `EMAIL`
  - `GCM`
  - `SMS`
  - `VOICE`

- `DeliveryUri` : destination vers laquelle les points de terminaison sont envoyés. Vous pouvez spécifier uniquement l'une des options suivantes :
  - URL du webhook auquel vous souhaitez envoyer des données de point de terminaison lors de l'exécution de la campagne.
  - L'Amazon Resource Name (ARN) d'une fonction Lambda que vous souhaitez exécuter lors de l'exécution de la campagne.

> **Note**
>
> L'objet `Campaign` peut également contenir un objet `Hook`. Cet objet est uniquement utilisé pour créer des segments qui sont personnalisés par une fonction Lambda lors de l'exécution d'une campagne. Pour de plus amples informations, veuillez consulter Personnalisez les segments Amazon Pinpoint à l'aide d'une fonction AWS Lambda.

Création et configuration d'une fonction Lambda pour une campagne Amazon Pinpoint

Cette section fournit une vue d'ensemble des étapes de création d'une fonction Lambda qui envoie des messages via un canal personnalisé. Tout d'abord, vous créez la fonction. Vous ajoutez ensuite une politique d'exécution à la fonction. Cette stratégie permet à Amazon Pinpoint d'exécuter la stratégie lors de l'exécution d'une campagne.

Pour une introduction à la création de fonctions Lambda, consultez Création de fonctions Lambda dans le Guide du développeur AWS Lambda .

Exemple de fonction Lambda

L'exemple de code suivant traite la charge utile et enregistre le nombre de points de terminaison de chaque type de point de terminaison. CloudWatch

```
import boto3
import random
import pprint
import json
import time

cloudwatch = boto3.client('cloudwatch')
```

```
def lambda_handler(event, context):
    customEndpoints = 0
    smsEndpoints = 0
    pushEndpoints = 0
    emailEndpoints = 0
    voiceEndpoints = 0
    numEndpoints = len(event['Endpoints'])

    print("Payload:\n", event)
    print("Endpoints in payload: " + str(numEndpoints))

    for key in event['Endpoints'].keys():
        if event['Endpoints'][key]['ChannelType'] == "CUSTOM":
            customEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "SMS":
            smsEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "EMAIL":
            emailEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "VOICE":
            voiceEndpoints += 1
        else:
            pushEndpoints += 1

    response = cloudwatch.put_metric_data(
        MetricData = [
            {
                'MetricName': 'EndpointCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': len(event['Endpoints'])
            },
            {
                'MetricName': 'CustomCount',
                'Dimensions': [
```

```
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': customEndpoints
        },
        {
            'MetricName': 'SMSCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': smsEndpoints
        },
        {
            'MetricName': 'EmailCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': emailEndpoints
        },
        {
            'MetricName': 'VoiceCount',
```

```
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': voiceEndpoints
            },
            {
                'MetricName': 'PushCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': pushEndpoints
            },
            {
                'MetricName': 'EndpointCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': len(event['Endpoints'])
            },
            {
                'MetricName': 'CustomCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': customEndpoints
            },
            {
                'MetricName': 'SMSCount',
```

```
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': smsEndpoints
            },
            {
                'MetricName': 'EmailCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': emailEndpoints
            },
            {
                'MetricName': 'VoiceCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': voiceEndpoints
            },
            {
                'MetricName': 'PushCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': pushEndpoints
            }
        ],
        Namespace = 'PinpointCustomChannelExecution'
    )
    print("cloudwatchResponse:\n",response)
```

Lorsqu'une campagne Amazon Pinpoint exécute cette fonction Lambda, Amazon Pinpoint envoie à la fonction une liste de membres du segment. La fonction compte le nombre de points de terminaison de chacun `ChannelType`. Il envoie ensuite ces données à Amazon CloudWatch. Vous pouvez consulter ces statistiques dans la section Métriques de la CloudWatch console. Les métriques sont disponibles dans l'espace de PinpointCustomChannelExecutionnoms.

Vous pouvez modifier cet exemple de code pour qu'il se connecte également à l'API d'un service externe, afin d'envoyer des messages à travers ce service.

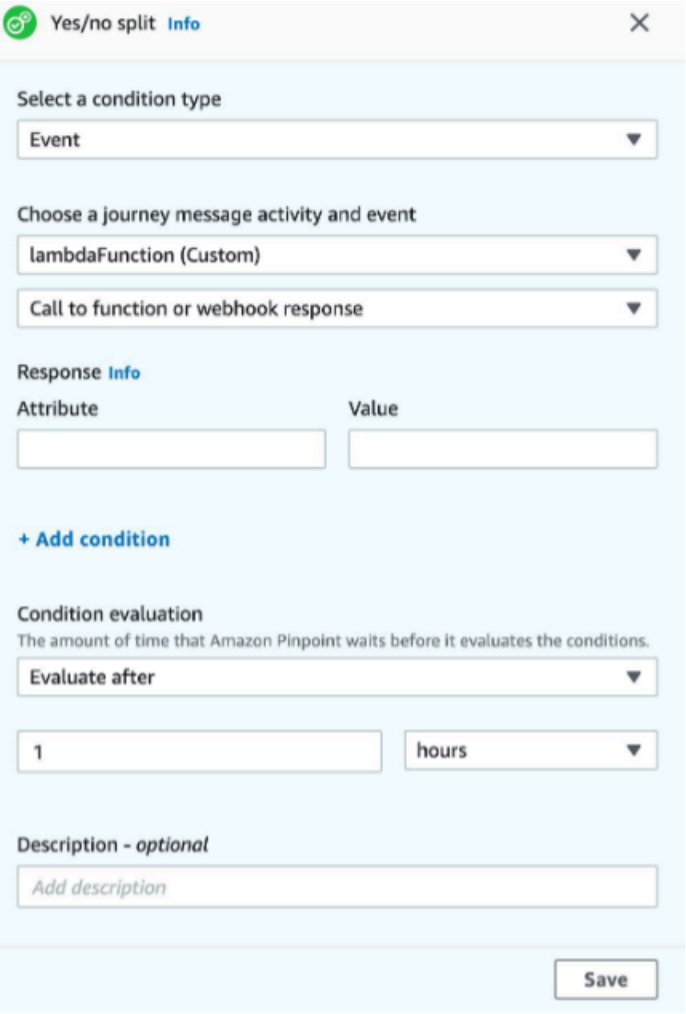
Format de réponse de fonction Lambda pour Amazon Pinpoint

Si vous souhaitez utiliser le parcours multivarié ou yes/no fractionné pour déterminer le chemin du point de terminaison après une activité de canal personnalisée, vous devez structurer la réponse de votre fonction Lambda dans un format compréhensible par Amazon Pinpoint, puis envoyer les points de terminaison sur le bon chemin.

La structure de la réponse doit être au format suivant :

```
{
    <Endpoint ID 1>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
    <Endpoint ID 2>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
...
}
```

Cela vous permettra ensuite de sélectionner une clé et une valeur que vous souhaitez utiliser pour déterminer le chemin du point de terminaison.



Accordez à Amazon Pinpoint l'autorisation d'appeler la fonction Lambda

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour ajouter des autorisations à la politique de fonction Lambda attribuée à votre fonction Lambda. Pour autoriser Amazon Pinpoint à invoquer une fonction, utilisez la commande Lambda ajouter autorisation, comme illustré dans l'exemple suivant :

```
aws lambda add-permission \
--function-name myFunction \
--statement-id sid0 \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:111122223333:apps/* 
--source-account 111122223333
```

Dans la commande précédente, procédez comme suit :

- Remplacez *myFunction* par le nom de la fonction Lambda.
- Remplacez *us-east-1* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
- Remplacez *111122223333* par votre ID de compte AWS .

Lorsque vous exécutez la commande `add-permission` de Lambda, celui-ci renvoie le résultat suivant :

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:myFunction\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/*\"}},
      {\"StringEquals\":
        {\"AWS:SourceAccount\":
          \"111122223333\"}}}
}
```

La valeur `Statement` est une version de chaîne JSON correspondant à l'instruction ajoutée à la stratégie de la fonction Lambda.

Restrictions supplémentaires concernant la stratégie d'exécution

Vous pouvez modifier la stratégie d'exécution en la limitant à un projet Amazon Pinpoint spécifique. Pour ce faire, remplacez le `*` dans l'exemple précédent par l'ID unique du projet. Vous pouvez restreindre davantage la stratégie en la limitant à une campagne spécifique. Par exemple, pour restreindre la stratégie de manière à autoriser uniquement une campagne avec l'ID de campagne `95fee4cd1d7f5cd67987c1436example` dans un projet avec l'ID de projet `dbaf6ec2226f0a9a8615e3ea5example`, utilisez la valeur suivante pour l'attribut `source-arn` :

```
arn:aws:mobiletargeting:us-east-1:111122223333:apps/dbaf6ec2226f0a9a8615e3ea5example/
campaigns/95fee4cd1d7f5cd67987c1436example
```

> **Note**
>
> Si vous limitez l'exécution de la fonction Lambda à une campagne spécifique, vous devez d'abord créer la fonction avec une stratégie moins restrictive. Ensuite, vous devez créer la campagne dans Amazon Pinpoint et choisir la fonction. Enfin, vous devez mettre à jour la stratégie d'exécution pour faire référence à la campagne spécifiée.

Diffusez les données d'événements des applications via Kinesis et Firehose à l'aide d'Amazon Pinpoint

Dans Amazon Pinpoint, un événement est une action qui se produit lorsqu'un utilisateur interagit avec l'une de vos applications, vous envoyez un message à partir d'une campagne ou d'un parcours, ou que vous envoyez un SMS ou un message électronique transactionnel. Par exemple, si vous envoyez un message électronique, plusieurs événements se produisent :

- Lorsque vous envoyez le message, un événement envoyé se produit.
- Lorsque le message arrive dans la boîte de réception du destinataire, un événement remis se produit.
- Lorsque le destinataire ouvre le message, un événement ouvert se produit.

Vous pouvez configurer Amazon Pinpoint pour envoyer des informations sur les événements vers Amazon Kinesis. La plateforme Kinesis propose des services que vous pouvez utiliser pour collecter, traiter et analyser les données des AWS services en temps réel. Amazon Pinpoint peut envoyer des données d'événements à Firehose, qui diffuse ces données vers des magasins de AWS données tels qu'Amazon S3 ou Amazon Redshift. Amazon Pinpoint peut également diffuser des données vers Kinesis Data Streams, qui ingère et stocke plusieurs flux de données pour les traiter par des applications analytiques.

Le flux d'événements Amazon Pinpoint inclut les informations sur les interactions utilisateur avec les applications que vous connectez à Amazon Pinpoint. Il inclut également des informations sur tous les messages que vous envoyez à partir de campagnes, via n'importe quel canal et depuis les parcours. Cela peut également inclure tous les événements personnalisés que vous avez définis. Enfin, il inclut des informations sur les e-mails transactionnels et les SMS que vous envoyez.

> **Note**
>
> Amazon Pinpoint ne diffuse pas d'informations sur les notifications push transactionnelles ou les messages vocaux.

Ce chapitre fournit des informations sur la configuration d'Amazon Pinpoint pour diffuser des données d'événement vers Kinesis. Il contient également des exemples de données d'événement diffusées par Amazon Pinpoint.

Rubriques

- Configurer Amazon Pinpoint pour diffuser les données d'événements des applications via Amazon Kinesis ou Amazon Data Firehose
- Flux de données d'événements d'applications provenant d'Amazon Pinpoint
- Flux de données sur les événements liés à la campagne depuis Amazon Pinpoint
- Données sur les événements du parcours provenant d'Amazon Pinpoint
- Flux de données d'événements par e-mail provenant d'Amazon Pinpoint
- Flux de données d'événements par SMS depuis Amazon Pinpoint
- Supprimer un flux d'événements d'Amazon Pinpoint

Configurer Amazon Pinpoint pour diffuser les données d'événements des applications via Amazon Kinesis ou Amazon Data Firehose

Vous pouvez configurer Amazon Pinpoint pour envoyer les données des événements vers un flux Amazon Kinesis ou un flux de diffusion Amazon Data Firehose. Amazon Pinpoint peut envoyer des données d'événements pour les campagnes, les parcours et les e-mails et SMS transactionnels.

Cette section inclut des informations sur la configuration de la diffusion en continu d'événements par programmation. Vous pouvez également utiliser la console Amazon Pinpoint pour configurer la diffusion en continu des événements. Pour plus d'informations sur la configuration de la diffusion en continu d'événements à l'aide de la console Amazon Pinpoint, consultez Paramètres de flux d'événements dans le Guide de l'utilisateur Amazon Pinpoint.

Conditions préalables

Les exemples de cette section nécessitent l'entrée suivante :

- ID d'application d'une application intégrée à Amazon Pinpoint et qui génère des rapports sur les événements. Pour plus d'informations sur l'intégration, consultez Intégrez Amazon Pinpoint à votre application.
- Le nom de ressource Amazon (ARN) d'un flux Kinesis ou d'un flux de diffusion Firehose dans votre compte. AWS Pour plus d'informations sur la création de ces ressources, consultez Creating and Managing Streams dans le guide du développeur Amazon Kinesis Data Streams ou Creating an Amazon Data Firehose delivery stream dans le manuel du développeur Amazon Data Firehose.

- L'ARN d'un rôle Gestion des identités et des accès AWS (IAM) qui autorise Amazon Pinpoint à envoyer des données au flux. Pour plus d'informations sur la création d'un rôle de , consultez la page Rôle IAM pour la diffusion des événements vers Kinesis.

AWS CLI

L' AWS CLI exemple suivant utilise la put-event-streamcommande. Cette commande configure Amazon Pinpoint pour envoyer des événements vers un flux Kinesis :

```
aws pinpoint put-event-stream \
--application-id projectId \
--write-event-stream DestinationStreamArn=streamArn,RoleArn=roleArn
```

AWS SDK pour Java

L'exemple Java suivant configure Amazon Pinpoint pour envoyer des événements vers un flux Kinesis :

```
public PutEventStreamResult createEventStream(AmazonPinpoint pinClient,
        String appId, String streamArn, String roleArn) {

    WriteEventStream stream = new WriteEventStream()
            .withDestinationStreamArn(streamArn)
            .withRoleArn(roleArn);

    PutEventStreamRequest request = new PutEventStreamRequest()
            .withApplicationId(appId)
            .withWriteEventStream(stream);

    return pinClient.putEventStream(request);
}
```

Cet exemple construit un `WriteEventStream` objet qui stocke le ARNs flux Kinesis et le rôle IAM. L'objet `WriteEventStream` est transmis à un objet `PutEventStreamRequest` afin de configurer Amazon Pinpoint pour diffuser en continu les événements d'une application spécifique. L'objet `PutEventStreamRequest` est transmis à la méthode `putEventStream` du client Amazon Pinpoint.

Vous pouvez attribuer un flux Kinesis à plusieurs applications. Dans ce cas, Amazon Pinpoint envoie les données d'événement de chaque application au flux, ce qui vous permet d'analyser les

données sous forme de collection. L'exemple de méthode suivant accepte une liste d'applications (app) IDs et utilise l'exemple de méthode précédent pour attribuer un flux à chaque application : `createEventStream`

```
public List<PutEventStreamResult> createEventStreamFromAppList(
        AmazonPinpoint pinClient, List<String> appIDs,
        String streamArn, String roleArn) {

    return appIDs.stream()
            .map(appId -> createEventStream(pinClient, appId, streamArn,
                    roleArn))
            .collect(Collectors.toList());
}
```

Même si vous pouvez attribuer un même flux à plusieurs applications, vous ne pouvez pas attribuer plusieurs flux à une même application.

Flux de données d'événements d'applications provenant d'Amazon Pinpoint

Après avoir intégré votre application (application) à Amazon Pinpoint et configuré le streaming d'événements, Amazon Pinpoint récupère l'activité des utilisateurs, les événements personnalisés et les données de livraison des messages de votre application depuis la destination que vous avez spécifiée lors de la configuration pour que vous puissiez les consulter. Pour plus d'informations sur la façon de configurer le streaming d'événements afin de consulter les données de vos événements, consultezConfigurer Amazon Pinpoint pour diffuser les données d'événements des applications via Amazon Kinesis ou Amazon Data Firehose.

Exemple d'événement d'application

L'objet JSON d'un événement d'application contient les données illustrées dans l'exemple suivant.

```
{
  "event_type": "_session.stop",
  "event_timestamp": 1487973802507,
  "arrival_timestamp": 1487973803515,
  "event_version": "3.0",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
```

```
    "cognito_identity_pool_id": "us-east-1:a1b2c3d4-e5f6-g7h8-i9j0-k1l2m3n4o5p6",
    "package_name": "main.page",
    "sdk": {
      "name": "aws-sdk-mobile-analytics-js",
      "version": "0.9.1:2.4.8"
    },
    "title": "title",
    "version_name": "1.0",
    "version_code": "1"
  },
  "client": {
    "client_id": "m3n4o5p6-a1b2-c3d4-e5f6-g7h8i9j0k1l2",
    "cognito_id": "us-east-1:i9j0k1l2-m3n4-o5p6-a1b2-c3d4e5f6g7h8"
  },
  "device": {
    "locale": {
      "code": "en_US",
      "country": "US",
      "language": "en"
    },
    "make": "generic web browser",
    "model": "Unknown",
    "platform": {
      "name": "android",
      "version": "10.10"
    }
  },
  "session": {
    "session_id": "f549dea9-1090-945d-c3d1-e4967example",
    "start_timestamp": 1487973202531,
    "stop_timestamp": 1487973802507
  },
  "attributes": {},
  "metrics": {}
}
```

Attributs d'événement d'application

Cette section définit les attributs inclus dans l'exemple précédent du flux d'événements de l'application.

Attribut	Description
`event_type`	Type d'événement. Les valeurs possibles sont : • _session.start : le point de terminaison a commencé une nouvelle session. • _session.stop : le point de terminaison a mis fin à une session. • _userauth.sign_in : point de terminaison connecté à votre application. • _userauth.sign_up : le nouveau point de terminaison a terminé le processus d'inscription dans votre application. • _userauth.auth_fail : le point de terminaison a tenté de se connecter à votre application, mais n'a pas pu terminer le processus. • _monetization.purchase : le point de terminaison a effectué un achat dans votre application. • _session.pause : la session du point de terminaison a été suspendue. Les sessions suspendues peuvent être reprises afin que vous puissiez continuer à collecter les métriques sans démarrer une toute nouvelle session. • _session.resume : la session du point de terminaison a repris.
`event_timestamp`	Heure à laquelle l'événement a été signalé, affichée sous la forme Heure Unix en millisecondes.
`arrival_timestamp`	Heure à laquelle l'événement a été reçu par Amazon Pinpoint, affichée sous la forme Heure Unix en millisecondes.

Attribut	Description
`event_version`	Version du schéma JSON de l'événement. **Tip** Vérifiez la version dans votre application de traitement des événements afin de savoir à quel moment mettre à jour l'application suite à une mise à jour du schéma.
`application`	Informations sur le projet Amazon Pinpoint associé à l'événement. Pour plus d'informations, consultez le tableau Application.
`client`	Informations sur le point de terminaison qui a signalé l'événement. Pour plus d'informations, consultez le tableau Client.
`device`	Informations sur l'appareil qui a signalé l'événement. Pour plus d'informations, consultez le tableau Appareil.
`session`	Informations sur la session qui a généré l'événement. Pour plus d'informations, consultez le tableau Session.
`attributes`	Attributs associés à l'événement. Pour les événements rapportés par vos applications, cet objet inclut les attributs personnalisés que vous définissez.
`metrics`	Métriques liées à l'événement. Si vous le souhaitez, vous pouvez configurer vos applications pour envoyer des métriques personnalisées à Amazon Pinpoint.

Application

Inclut les informations sur le projet Amazon Pinpoint auquel l'événement est associé.

Attribut	Description
`app_id`	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
`cognito_identity_pool_id`	ID de la réserve d'identités Amazon Cognito auquel le point de terminaison est associé.
`package_name`	Nom du package d'application, par exemple `com.example.my_app` .
`sdk`	Informations sur le kit SDK qui a été utilisé pour signaler l'événement. Pour plus d'informations, consultez le tableau kit SDK.
`title`	Nom de l'application.
`version_name`	Nom de la version de l'application (par exemple, `V2.5`).
`version_code`	Numéro de version de l'application, tel que `3`.

SDK

Informations sur le kit SDK qui a été utilisé pour signaler l'événement.

Attribut	Description
`name`	Nom du kit SDK qui a été utilisé pour signaler l'événement.
`version`	Version du kit SDK.

Client

Inclut des informations sur le point de terminaison qui a généré l'événement.

Attribut	Description
`client_id`	ID du point de terminaison.
`cognito_id`	Jeton d'ID Amazon Cognito associé au point de terminaison.

Appareil

Inclut des informations sur l'appareil du point de terminaison qui a généré l'événement.

Attribut	Description
`locale`	Informations sur les paramètres de langue et de région de l'appareil du point de terminaison. Pour plus d'informations, consultez le tableau Locale.
`make`	Fabricant de l'appareil du point de terminaison.
`model`	Identifiant du modèle de l'appareil du point de terminaison.
`platform`	Informations sur le système d'exploitation sur le périphérique du point de terminaison. Pour plus d'informations, consultez le tableau Plateforme.

Paramètre régional

Informations sur les paramètres de langue et de région de l'appareil du point de terminaison.

Attribut	Description
`code`	Identifiant local associé à l'appareil.
`country`	Pays ou région associé aux paramètres régionaux de l'appareil.
`language`	Langue associée aux paramètres régionaux de l'appareil.

Plateforme

Informations sur le système d'exploitation de l'appareil du point de terminaison.

Attribut	Description
`name`	Nom du système d'exploitation sur l'appareil.
`version`	Version du système d'exploitation de l'appareil.

Session

Inclut des informations sur la session qui a généré l'événement.

Attribut	Description
`session_id`	ID unique qui identifie la session.
`start_timestamp`	Date et heure de début de la session, affichées sous la forme Heure Unix en millisecondes.
`stop_timestamp`	Date et heure auxquelles la session s'est terminée, affichées sous la forme Heure Unix en millisecondes.

Flux de données sur les événements liés à la campagne depuis Amazon Pinpoint

Si vous utilisez Amazon Pinpoint pour envoyer des campagnes via un canal, Amazon Pinpoint peut diffuser les données d'événements relatives à ces campagnes. Une fois que vous avez configuré le streaming d'événements, Amazon Pinpoint récupère les données relatives aux événements de votre application pour les e-mails ou les SMS que vous envoyez depuis une campagne depuis la destination que vous avez spécifiée lors de la configuration pour que vous puissiez les consulter. Pour obtenir des informations détaillées sur les données diffusées par Amazon Pinpoint pour les e-mails et les SMS, consultez the section called “Flux de données d'événements par e-mail provenant d'Amazon Pinpoint” et. the section called “Flux de données d'événements par SMS depuis Amazon Pinpoint” Pour plus d'informations sur la configuration de la diffusion d'événements, consultezConfigurer Amazon Pinpoint pour diffuser les données d'événements des applications via Amazon Kinesis ou Amazon Data Firehose.

Exemple d'événement de campagne

L'objet JSON d'un événement de campagne contient les données présentées dans l'exemple suivant.

```
{
  "event_type": "_campaign.send",
  "event_timestamp": 1562109497426,
  "arrival_timestamp": 1562109497494,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "d8dcf7c5-e81a-48ae-8313-f540cexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "treatment_id": "0",
    "campaign_activity_id": "5473285727f04865bc673e527example",
    "delivery_type": "GCM",
    "campaign_id": "4f8d6097c2e8400fa3081d875example",
```

```
    "campaign_send_status": "SUCCESS"
  },
  "client_context": {
    "custom": {
      "endpoint": "{\"ChannelType\":\"GCM\",\"EndpointStatus\":\"ACTIVE\",
          #\"OptOut\":\"NONE\",\"RequestId\":\"ec229696-9d1e-11e9-8bf1-85d0aexample\",
          #\"EffectiveDate\":\"2019-07-02T23:12:54.836Z\",\"User\":{}}"
    }
  },
  "awsAccountId": "123456789012"
}
```

Attributs d'événement de campagne

Cette section définit les attributs qui sont inclus dans le flux d'événements de la campagne.

Attribut	Description
`event_type`	Type d’événement. Les valeurs possibles sont : • _campaign.send : Amazon Pinpoint a exécuté la campagne. • _campaign.opened_notification : pour les campagnes de notification push, ce type d'événement indique que le destinataire a utilisé la notification pour l'ouvrir. • _campaign.received_foreground : pour les campagnes de notification push, ce type d'événement indique que le destinataire a reçu le message en tant que notification au premier plan. • _campaign.received_background : pour les campagnes de notification push, ce type d'événement indique que le destinataire a reçu le message en tant que notification en arrière plan.

Attribut	Description
	Note _campaign.opened_notification, _campaign.received_foreground et _campaign.received_background sont uniquement renvoyés si vous utilisez AWS Amplify. Pour plus d'informations sur l'intégration de votre application à AWS Amplify. Consultez Connectez votre application frontale à Amazon AWS Pinpoint à l'aide d'Amplify.
`event_timestamp`	Heure à laquelle l'événement a été signalé, affichée sous la forme Heure Unix en millisecondes.
`arrival_timestamp`	Heure à laquelle l'événement a été reçu par Amazon Pinpoint, affichée sous la forme Heure Unix en millisecondes.
`event_version`	Version du schéma JSON de l'événement. **Tip** Vérifiez la version dans votre application de traitement des événements afin de savoir à quel moment mettre à jour l'application suite à une mise à jour du schéma.
`application`	Informations sur le projet Amazon Pinpoint associé à l'événement. Pour plus d'informations, consultez le tableau Application.

Attribut	Description
`client`	Informations sur le point de terminaison auquel l'événement est associé. Pour plus d'informations, consultez le tableau Client.
`device`	Informations sur l'appareil qui a signalé l'événement. Pour les messages de campagne et transactionnels, cet objet est vide.
`session`	Informations sur la session qui a généré l'événement. Pour les campagnes, cet objet est vide.
`attributes`	Attributs associés à l'événement. Pour les événements rapportés par l'une de vos applications, cet objet peut inclure des attributs personnalisés qui sont définis par l'application. Pour les événements qui sont créés lorsque vous envoyez une campagne, cet objet contient les attributs associés à la campagne. Pour les événements qui sont générés lorsque vous envoyez des messages transactionnels, cet objet contient des informations liées au message lui-même. Pour plus d'informations, consultez le tableau Attributs.
`client_context`	Contient un objet `custom`, qui contient une propriété `endpoint`. La propriété `endpoint` propriété contient le contenu de l'enregistrement de point de terminaison pour le point de terminaison auquel la campagne a été envoyée.
`awsAccountId`	ID du AWS compte utilisé pour envoyer le message.

Application

Inclut les informations sur le projet Amazon Pinpoint auquel l'événement est associé.

Attribut	Description
`app_id`	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
`sdk`	Le kit SDK qui a été utilisé pour signaler l'événement.

Attributes

Inclut des informations sur la campagne qui a généré l'événement.

Attribut	Description
`treatment_id`	Si le message a été envoyé à l'aide d'une campagne de A/B test, cette valeur représente le numéro de traitement du message. Pour les campagnes standard, cette valeur est `0`.
`campaign_activity_id`	ID unique généré par Amazon Pinpoint lorsque l'événement se produit.
`delivery_type`	Mode de livraison de la campagne. Ne confondez pas cet attribut avec le champ `ChannelType` spécifié sous la propriété `endpoint` de `client_context` . Le champ `ChannelType` est généralement basé sur le point de terminaison auquel le message est envoyé. Pour les canaux qui ne prennent en charge qu'un seul type de point de terminaison, les champs `delivery_type` et `ChannelType` ont la même valeur. Par exemple, pour le

Attribut	Description
	canal e-mail, les champs `delivery_type` et `ChannelType` ont la même valeur que EMAIL. Toutefois, cette condition n'est pas toujours vraie pour les canaux qui prennent en charge différents types de points de terminaison, tels que les canaux personnalisés. Vous pouvez utiliser un canal personnalisé pour différents points de terminaison, tels que EMAIL, SMS, CUSTOM, etc. Dans ce cas, le `delivery_type` identifie un événement de distribution personnalisé, CUSTOM, et `ChannelType` spécifie le type de point de terminaison auquel la campagne a été envoyée, tel que EMAIL, SMS, CUSTOM, etc. Pour en savoir plus sur la création de canaux personnalisés, consultez la section Créez une chaîne personnalisée. Les valeurs possibles sont : • EMAIL • SMS • ADM • APNS • APNS_SANDBOX • APNS_VOIP • APNS_VOIP_SANDBOX • VOICE • GCM • BAIDU • PUSH • CUSTOM

Attribut	Description
`campaign_id`	ID unique de la campagne à partir de laquelle le message a été envoyé.

Attribut	Description
`campaign_send_status`	Indique le statut de la campagne pour le point de terminaison cible. Les valeurs possibles incluent : • SUCCESS : la campagne a été envoyée avec succès au point de terminaison. • FAILURE : la campagne n'a pas été envoyée au point de terminaison. • DAILY_CAP : la campagne n'a pas été envoyée au point de terminaison, car le nombre maximal de messages quotidiens a déjà été envoyé au point de terminaison. • EXPIRED : la campagne n'a pas été envoyée au point de terminaison, car son envoi dépasserait la durée maximale ou les paramètres de taux d'envoi de la campagne. • QUIET_TIME : la campagne n'a pas été envoyée au point de terminaison en raison de restrictions de temps de pause. • HOLDOUT : la campagne n'a pas été envoyée au point de terminaison, car ce dernier était membre du groupe d'éléments d'arrêt. • DUPLICATE_ADDRESS : le segment contient des adresses de point de terminais on dupliquées. La campagne a été envoyée une fois à l'adresse du point de terminaison. • QUIET_TIME : la campagne n'a pas été envoyée au point de terminaison en raison de restrictions de temps de pause. • CAMPAIGN_CAP : la campagne n'a pas été envoyée au point de terminaison, car le nombre maximal de messages a déjà été

Attribut	Description
	envoyé au point de terminaison à partir de cette campagne. • FAILURE_PERMANENT : une défaillance permanente s'est produite lors de l'envoi vers le point de terminaison. • TRANSIENT_FAILURE : une défaillance transitoire s'est produite lors de l'envoi vers le point de terminaison. • THROTTLED : l'envoi a été limité. • UNKNOWN : défaillance inconnue. • HOOK_FAILURE : le hook de campagne a échoué. • CUSTOM_DELIVERY_FAILURE : la distribut ion personnalisée a échoué. • RECOMMENDATION_FAILURE : le moteur de recommendation a échoué. • UNSUPPORTED_CHANNEL : le canal n'est pas pris en charge.

Client

Inclut des informations sur le point de terminaison qui a été ciblé par la campagne.

Attribut	Description
`client_id`	ID du point de terminaison auquel la campagne a été envoyée.

Données sur les événements du parcours provenant d'Amazon Pinpoint

Lorsque vous publiez un parcours, Amazon Pinpoint peut diffuser les données des événements pour les e-mails, les SMS, les notifications push et les messages personnalisés que vous envoyez dans le cadre du parcours. Après avoir configuré le streaming d'événements, Amazon Pinpoint extrait les données de la destination que vous avez spécifiée lors de la configuration pour que vous puissiez les consulter. Pour obtenir des informations détaillées sur les données diffusées par Amazon Pinpoint pour les e-mails et les SMS, consultez the section called “Flux de données d'événements par e-mail provenant d'Amazon Pinpoint” et. the section called “Flux de données d'événements par SMS depuis Amazon Pinpoint” Pour plus d'informations sur la configuration de la diffusion d'événements, consultezConfigurer Amazon Pinpoint pour diffuser les données d'événements des applications via Amazon Kinesis ou Amazon Data Firehose.

Exemple d'événement Journey

L'objet JSON d'un événement de parcours contient les données illustrées dans l'exemple suivant.

```
{
   "event_type":"_journey.send",
   "event_timestamp":1572989078843,
   "arrival_timestamp":1572989078843,
   "event_version":"3.1",
   "application":{
      "app_id":"a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
      "sdk":{

      }
   },
   "client":{
      "client_id":"d8dcf7c5-e81a-48ae-8313-f540cexample"
   },
   "device":{
      "platform":{

      }
   },
   "session":{

   },
```

```
   "attributes":{
      "journey_run_id":"edc9a0b577164d1daf72ebd15example",
      "journey_send_status":"SUCCESS",
      "journey_id":"546401670c5547b08811ac6a9example",
      "journey_activity_id":"0yKexample",
      "journey_activity_type": "EMAIL",
      "journey_send_status_message": "200",
      "journey_send_status_code": "200"
   },
   "client_context":{
      "custom":{
         "endpoint":"{\"ChannelType\":\"EMAIL\",\"EndpointStatus\":\"ACTIVE\",\"OptOut
\":\"NONE\",\"Demographic\":{\"Timezone\":\"America/Los_Angeles\"}}"
      }
   },
   "awsAccountId":"123456789012"
}
```

Attributs de l'événement de parcours

Cette section définit les attributs qui sont inclus dans les données de flux d'événements générés par Amazon Pinpoint pour un parcours.

Attribut	Description
`event_type`	Type d'événement. Pour les événements de parcours, la valeur de cet attribut est toujours `_journey.send` , ce qui indique que le parcours a été exécuté par Amazon Pinpoint.
`event_timestamp`	Heure à laquelle l'événement a été signalé, affichée sous la forme Heure Unix en millisecondes.
`arrival_timestamp`	Heure à laquelle l'événement a été reçu par Amazon Pinpoint, affichée sous la forme Heure Unix en millisecondes.
`event_version`	Version du schéma JSON de l'événement.

Attribut	Description
	Tip Vérifiez la version dans votre application de traitement des événements afin de savoir à quel moment mettre à jour l'application suite à une mise à jour du schéma.
`application`	Informations sur le projet Amazon Pinpoint associé à l'événement. Pour plus d'informations, consultez le tableau Application.
`client`	Informations sur le point de terminaison auquel l'événement est associé. Pour plus d'informations, consultez le tableau Client.
`device`	Informations sur l'appareil qui a signalé l'événement. Pour les parcours, cet objet est vide.
`session`	Informations sur la session qui a généré l'événement. Pour les parcours, cet objet est vide.
`attributes`	Attributs associés au parcours et à l'activité de parcours et de parcours qui a généré l'événement. Pour plus d'informations, consultez le tableau Attributs.
`client_context`	Contient un objet `custom`, qui contient une propriété `endpoint`. La propriété `endpoint` contient les contenus de l'enregistrement de point de terminaison du point de terminaison associé à l'événement.

Attribut	Description
`awsAccountId`	L'identifiant du AWS compte utilisé pour exécuter le trajet.

Application

Inclut les informations sur le projet Amazon Pinpoint auquel l'événement est associé.

Attribut	Description
`app_id`	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
`sdk`	Le kit SDK qui a été utilisé pour signaler l'événement.

Client

Inclut des informations sur le point de terminaison associé à l'événement.

Attribut	Description
`client_id`	ID du point de terminaison.

Attributes

Inclut des informations sur le parcours qui a généré l'événement.

Attribut	Description
`journey_run_id`	ID unique de l'exécution du parcours qui a généré l'événement. Amazon Pinpoint génère et attribue automatiquement cet identifiant à chaque nouvelle étape d'un parcours.

Attribut	Description
`journey_send_status`	Indique l'état de remise du message qui est associé à l'événement. Les valeurs possibles incluent : • SUCCESS : le message a été envoyée avec succès au point de terminaison. • FAILURE : le message n'a pas été envoyé au point de terminaison car une erreur s'est produite. • CUSTOM_DELIVERY_FAILURE : la distribut ion personnalisée a échoué. • FAILURE_PERMANENT : une défaillance permanente s'est produite lors de l'envoi vers le point de terminaison. **Tip** Vous pouvez filtrer les événement s dont le statut FAILURE_P ERMANENT est `journey_s end_status_code` défini sur 403 pour déterminer s'il existe une politique d'accès et une violation de rôle. Pour les campagnes sortantes avec commande vocale, ces exceptions sont typiques des cas où le rôle d'exécution de campagnes Connect liant les campagnes Amazon Pinpoint Journeys à Connect Customer est supprimé par inadvertance pour les exécutions en vol. • THROTTLED : l'envoi a été limité.

Attribut	Description
	• UNSUPPORTED_CHANNEL : le canal n'est pas pris en charge. • DAILY_CAP : le message n'a pas été envoyé au point de terminaison car l'envoi du message dépasserait le nombre maximum de messages que le parcours ou le projet peut envoyer à un point de terminaison unique sur une période de 24 heures. • QUIET_TIME : le message n'a pas été envoyé en raison des restrictions relatives au temps de pause pendant le parcours ou le projet. • QUIET_TIME_MISSING_TIMEZONE : le message n'a pas été envoyé car l'estimat ion du fuseau horaire n'a pas pu estimer le fuseau horaire du point de terminaison et le temps de pause est activé.
`journey_id`	ID unique du parcours qui a généré l'événeme nt.
`journey_activity_id`	ID unique de l'activité de parcours qui a généré l'événement.

Attribut	Description
`journey_activity_type`	Type d'activité lié au parcours de l'événement. Cela peut être EMAIL, SMS, PUSH, CONTACT_CENTER ou CUSTOM. **Note** VOICE n'est pas un type d'activité de parcours pris en charge. Le `journey_activity_type` champ n'est pas présent lorsqu'il `journey_send_status` est défini sur QUIET_TIME_WAIT_FINISHED.
`journey_send_status_message`	Description de l'état de l'événement d'envoi.
`journey_send_status_code`	Le code d'état HTTP de la demande.

Flux de données d'événements par e-mail provenant d'Amazon Pinpoint

Si vous utilisez Amazon Pinpoint pour envoyer des e-mails, Amazon Pinpoint peut diffuser des données d'événements concernant ces e-mails. Une fois que vous avez configuré le streaming d'événements, Amazon Pinpoint extrait les données de vos événements depuis la destination que vous avez spécifiée lors de la configuration pour que vous puissiez les consulter. Pour plus d'informations sur la configuration de la diffusion d'événements, consultezConfigurer Amazon Pinpoint pour diffuser les données d'événements des applications via Amazon Kinesis ou Amazon Data Firehose. Amazon Pinpoint diffuse des données relatives aux types d'événements suivants pour les e-mails :

- Envois
- Messages délivrés
- Retours à l'expéditeur
- Réclamations

- Messages ouverts
- Clics
- Rejets
- Désabonnement
- Échec du rendu

Ces types d'événements sont expliqués en détail dansAttributs d'événement d'e-mail.

Selon l'API et les paramètres que vous utilisez pour envoyer des messages électroniques, vous pouvez voir des types d'événements supplémentaires ou des données différentes. Par exemple, si vous envoyez des messages à l'aide de jeux de configuration qui publient des données d'événement dans Amazon Kinesis, tels que ceux fournis par Amazon Simple Email Service (Amazon SES), les données peuvent également inclure des événements pour des échecs de rendu de modèle. Pour plus d'informations sur ces données, consultez Surveillance à l'aide de la publication d'événements Amazon SES dans le Guide du développeur Amazon Simple Email Service. Avant de pouvoir visionner vos événements, vous devez configurer le streaming des événements, voirConfigurer Amazon Pinpoint pour diffuser les données d'événements des applications via Amazon Kinesis ou Amazon Data Firehose. Lorsque vous configurez le streaming d'événements, vous spécifiez une destination pour l'enregistrement des données de votre événement, puis vous pouvez utiliser cette destination pour récupérer les données de vos événements afin de les consulter.

Exemples d'événements par e-mail

Envoi d'e-mails

L'objet JSON d'un événement email send contient les données illustrées dans l'exemple suivant.

```
{
  "event_type": "_email.send",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622025,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "9a311b17-6f8e-4093-be61-4d0bbexample"
  },
```

```
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "received"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "send": {}
      }
    }
  }
```

```
}
```

Livraison d'e-mail

L'objet JSON d'un événement email delivered contient les données illustrées dans l'exemple suivant.

```
{
  "event_type": "_email.delivered",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622690,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "e9a3000d-daa2-40dc-ac47-1cd34example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "delivered"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
```

```
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "delivery": {
          "smtp_response": "250 ok:  Message 82080542 accepted",
          "reporting_mta": "a8-53.smtp-out.amazonses.com",
          "recipients": ["recipient@example.com"],
          "processing_time_millis": 1310
        }
      }
    }
  }
}
```

Clic sur e-mail

L'objet JSON d'un événement email click contient les données illustrées dans l'exemple suivant.

```
{
  "event_type": "_email.click",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618713751,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "49c1413e-a69c-46dc-b1c4-6470eexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
```

```
  "attributes": {
    "feedback": "https://aws.amazon.com/pinpoint/"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "click": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15",
          "link": "https://aws.amazon.com/pinpoint/"
        }
```

```
      }
    }
  }
}
```

Ouvertures d'e-mails

L'objet JSON d'un événement email open contient les données illustrées dans l'exemple suivant.

```
{
  "event_type": "_email.open",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618712316,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "8dc1f651-b3ec-46fc-9b67-2a050example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "opened"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
```

```
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "open": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko)"
        }
      }
    }
  }
}
```

Attributs d'événement d'e-mail

Cette section définit les attributs inclus dans l'exemple précédent des données de flux d'événements générées par Amazon Pinpoint lorsque vous envoyez des e-mails.

Attribut	Description
`event_type`	Type d'événement. Les valeurs possibles sont : • _email.send : Amazon Pinpoint a accepté le message et a tenté de le remettre au destinataire.

Attribut	Description
	• _email.delivered : le message a été remis au destinataire. • _email.rejected : Amazon Pinpoint a déterminé que le message contenait un programme malveillant et n'a pas tenté de l'envoyer. • _email.hardbounce : un problème permanent a empêché Amazon Pinpoint de délivrer le message. Amazon Pinpoint ne tentera pas de remettre le message à nouveau. • _email.softbounce : un problème temporair e a empêché Amazon Pinpoint de délivrer le message. Amazon Pinpoint tentera de remettre le message pendant un certain laps de temps. Si le message ne peut toujours pas être remis, aucune autre tentative ne sera effectuée. L'état final de l'e-mail sera alors SOFTBOUNCE. • _email.complaint : le destinataire a reçu le message, puis l'a signalé comme courrier indésirable pour son fournisseur de messagerie (par exemple, en utilisant la fonctionnalité « Signaler le courrier indésirab le » de son client de messagerie). • _email.open : le destinataire a reçu le message et l'a ouvert. • _email.click : le destinataire a reçu le message et a cliqué sur un lien qu'il contient. • _email.unsubscribe : le destinataire a reçu le message et a cliqué sur le lien de désabonne ment qu'il contient. • _email.rendering_failure : l'e-mail n'a pas été envoyé en raison d'un échec de rendu.

Attribut	Description
	Cela peut se produire lorsqu'il manque des données du modèle ou lorsqu'il n'y a pas concordance entre les paramètres du modèle et les données.
`event_timestamp`	Heure à laquelle le message a été envoyé, indiquée exprimée en millisecondes de temps Unix. Cette valeur est généralement la même pour tous les événements générés pour un message.
`arrival_timestamp`	Heure à laquelle l'événement a été reçu par Amazon Pinpoint, affichée sous la forme Heure Unix en millisecondes.
`event_version`	Version du schéma JSON de l'événement. **Tip** Vérifiez la version dans votre application de traitement des événements afin de savoir à quel moment mettre à jour l'application suite à une mise à jour du schéma.
`application`	Informations sur le projet Amazon Pinpoint associé à l'événement. Pour plus d'informations, consultez le tableau Application.
`client`	Informations sur le client d'application installé sur l'appareil qui a signalé l'événement. Pour plus d'informations, consultez le tableau Client.

Attribut	Description
`device`	Informations sur l'appareil qui a signalé l'événement. Pour plus d'informations, consultez le tableau Appareil. Pour les événements d'e-mail, cet objet est vide.
`session`	Pour les événements d'e-mail, cet objet est vide.
`attributes`	Attributs associés à l'événement. Pour plus d'informations, consultez le tableau Attributs. Pour les événements rapportés par l'une de vos applications, cet objet peut inclure des attributs personnalisés qui sont définis par l'applica tion. Pour les événements qui sont créés lorsque vous envoyez un message à partir d'une campagne ou d'un parcours, cet objet contient les attributs associés à la campagne ou au parcours. Pour les événements qui sont générés lorsque vous envoyez des messages transactionnels, cet objet contient des informati ons liées au message lui-même.
`client_context`	Pour les événements de messagerie, cet objet contient un objet `custom`, qui contient un attribut `legacy_identifier` . La valeur de l'attribut `legacy_identifier` est l'ID du projet à partir de laquelle le message a été envoyé.
`facets`	Informations supplémentaires sur le message, telles que les en-têtes d'e-mail. Consultez le tableau Facettes pour plus d'informations.

Attribut	Description
`awsAccountId`	L'ID du AWS compte qui a été utilisé pour envoyer le message.

Application

Inclut les informations sur le projet Amazon Pinpoint auquel l'événement est associé.

Attribut	Description
`app_id`	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
`sdk`	Le kit SDK qui a été utilisé pour signaler l'événement. Si vous envoyez un e-mail transactionnel en appelant directement l'API Amazon Pinpoint ou en utilisant la console Amazon Pinpoint, cet objet est vide.

Attributs

Inclut des informations sur la campagne ou le parcours qui a généré l'événement.

Campagne

Inclut des informations sur la campagne qui a généré l'événement.

Attribut	Description
`feedback`	Pour les événements `_email.click` , la valeur de cet attribut est l'URL du lien sur lequel le destinataire a cliqué dans le message pour générer l'événement. Pour les autres événements, cette valeur représente le type d'événement, tel que `received`, `opened` ou `clicked`.

Attribut	Description
`treatment_id`	Si le message a été envoyé à l'aide d'une campagne de test A/B, cette valeur représente le numéro de traitement du message. Pour les campagnes standard et les e-mails transactionnels, cette valeur est `0`.
`campaign_activity_id`	ID unique généré par Amazon Pinpoint lorsque l'événement se produit.
`campaign_id`	ID unique de la campagne qui a envoyé le message.

Parcours

Inclut des informations sur le parcours qui a généré l'événement.

Attribut	Description
`journey_run_id`	ID unique de l'exécution du parcours qui a envoyé le message. Amazon Pinpoint génère et attribue automatiquement cet identifiant à chaque nouvelle étape d'un parcours.
`feedback`	Pour les événements `_email.click` , la valeur de cet attribut est l'URL du lien sur lequel le destinataire a cliqué dans le message pour générer l'événement. Pour les autres événements, cette valeur représente le type d'événement, tel que `received`, `delivered` ou `opened`.
`journey_id`	ID unique du parcours qui a envoyé le message.
`journey_activity_id`	ID unique de l'activité de parcours qui a envoyé le message.

Client

L'identifiant unique du client ciblé par la campagne ou le parcours.

Attribut	Description
`client_id`	ID du client. La valeur est l'identifiant du point de terminaison pour les campagnes et les parcours, et pour l'envoi transactionnel, il s'agit d'un UUID.

Facettes

Inclut des informations sur le message et le type d'événement.

Attribut	Description
`email_channel`	Contient un objet `mail_event` , qui contient deux objets : `mail` et un objet correspondant au type d'événement.

Mail

Inclut des informations sur le contenu de l'e-mail, ainsi que des métadonnées liées au message.

Attribut	Description
`message_id`	ID unique du message. Amazon Pinpoint génère automatiquement cet ID lorsqu'il accepte le message.
`message_send_timestamp`	Date et heure auxquelles le message a été envoyé au format spécifié dans RFC 822.
`from_address`	Adresse e-mail à partir de laquelle le message a été envoyé.

Attribut	Description
`destination`	Tableau contenant les adresses e-mail auxquelles le message a été envoyé.
`headers_truncated`	Valeur booléenne qui indique si les en-têtes d'e-mail ont été tronqués.
`headers`	Objet qui contient plusieurs paires nom-valeur correspondant aux en-têtes du message. Cet objet contient généralement des informations sur les en-têtes suivants : • `From` : adresse e-mail de l'expéditeur. • `To` : adresse e-mail du destinataire. • `Subject` : ligne Objet de l'e-mail. **Tip** L'en-tête du sujet n'est pas inclus pour les événements Campaign _email.send. • `MIME-Version` : indique que le message est au format MIME. Si cet en-tête est présent, la valeur est toujours `1.0`. • `Content-Type` : type de média MIME du contenu du message.
`common_headers`	Contient des informations sur plusieurs en-têtes communs pour les messages électroniques. Les informations peuvent inclure la date à laquelle le message a été envoyé, ainsi que les lignes de destination, d'expéditeur et d'objet du message.

Flux de données d'événements par SMS depuis Amazon Pinpoint

Si le canal SMS est activé pour un projet, Amazon Pinpoint peut diffuser des données d'événement sur les remises de messages SMS pour le projet. Une fois que vous avez configuré le streaming d'événements, Amazon Pinpoint extrait les données de vos événements depuis la destination que vous avez spécifiée lors de la configuration pour que vous puissiez les consulter. Pour plus d'informations sur la configuration de la diffusion d'événements, consultezConfigurer Amazon Pinpoint pour diffuser les données d'événements des applications via Amazon Kinesis ou Amazon Data Firehose.

> **Note**
>
> Les événements SMS générés par les opérateurs peuvent prendre jusqu'à 72 heures pour être reçus et ne doivent pas être utilisés pour déterminer s'il y a un retard dans la livraison des messages sortants. Au bout de 72 heures, si Amazon Pinpoint n'a pas reçu d'événement final de la part d'un transporteur, le service renvoie automatiquement un record_status INCONNU, car Amazon Pinpoint ne sait pas ce qu'il est advenu de ce message.

Exemple d'événement SMS

L'objet JSON d'un événement SMS contient les données illustrées dans l'exemple suivant.

```
{
  "event_type": "_SMS.SUCCESS",
  "event_timestamp": 1553104954322,
  "arrival_timestamp": 1553104954064,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "123456789012"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
```

```
    "sender_request_id": "565d4425-4b3a-11e9-b0a5-example",
    "campaign_activity_id": "cbcfc3c5e3bd48a8ae2b9cb41example",
    "origination_phone_number": "+12065550142",
    "destination_phone_number": "+14255550199",
    "record_status": "DELIVERED",
    "iso_country_code": "US",
    "treatment_id": "0",
    "number_of_message_parts": "1",
    "message_id": "1111-2222-3333",
    "message_type": "Transactional",
    "campaign_id": "52dc44b35c4742c98c5935269example"
    "customer_context": "{\"userId\":\"user-id-4\"}"
  },
  "metrics": {
    "price_in_millicents_usd": 645.0
  },
  "awsAccountId": "123456789012"
}
```

Attributs d'événement SMS

Cette section définit les attributs inclus dans l'exemple précédent des données de flux d'événements générées par Amazon Pinpoint lorsque vous envoyez des SMS.

Événement

Attribut	Description
`event_type`	Type d'événement. Les valeurs possibles sont : • _SMS.BUFFERED : le message est toujours en cours de remise au destinataire. • _SMS.SUCCESS : le message a été accepté par l'opérateur/remis au destinataire. • _SMS.FAILURE : Amazon Pinpoint n'a pas été en mesure de transmettre le message au destinataire. Pour en savoir plus sur l'erreur qui a empêché la remise du message, consultez `attributes.record_status` .

Attribut	Description
	• _SMS.OPTOUT : le client a reçu le message et a répondu en envoyant le mot clé de refus (généralement « STOP »).
`event_timestamp`	Heure à laquelle l'événement a été signalé, affichée sous la forme Heure Unix en millisecondes.
`arrival_timestamp`	Heure à laquelle l'événement a été reçu par Amazon Pinpoint, affichée sous la forme Heure Unix en millisecondes.
`event_version`	Version du schéma JSON de l'événement. **Tip** Vérifiez la version dans votre application de traitement des événements afin de savoir à quel moment mettre à jour l'application suite à une mise à jour du schéma.
`application`	Informations sur le projet Amazon Pinpoint associé à l'événement. Pour plus d'informations, consultez le tableau Application.
`client`	Informations sur le client d'application installé sur le périphérique qui a signalé l'événement. Pour plus d'informations, consultez le tableau Client.
`device`	Informations sur l'appareil qui a signalé l'événement. Pour plus d'informations, consultez le tableau Appareil. Pour les événements SMS, cet objet est vide.

Attribut	Description
`session`	Pour les événements SMS, cet objet est vide.
`attributes`	Attributs associés à l'événement. Pour les événements rapportés par l'une de vos applications, cet objet peut inclure des attributs personnalisés qui sont définis par l'application. Pour les événements qui sont créés lorsque vous envoyez une campagne, cet objet contient les attributs associés à la campagne. Pour les événements qui sont générés lorsque vous envoyez des messages transactionnels, cet objet contient des informations liées au message lui-même. Pour plus d'informations, consultez le tableau Attributs.
`metrics`	Métriques supplémentaires associées à l'événement. Pour plus d'informations, consultez le tableau Métriques.
`awsAccountId`	L'ID du AWS compte qui a été utilisé pour envoyer le message.

Application

Comporte des informations sur le projet Amazon Pinpoint auquel l'événement est associé et, le cas échéant, sur le kit SDK utilisé pour signaler l'événement.

Attribut	Description
`app_id`	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
`sdk`	Le kit SDK qui a été utilisé pour signaler l'événement. Si vous envoyez un SMS transacti

Attribut	Description
	onnel en appelant directement l'API Amazon Pinpoint ou en utilisant la console Amazon Pinpoint, cet objet est vide.

Attributs

Comporte des informations sur les attributs associés à l'événement.

Attribut	Description
`sender_request_id`	ID unique associé à la demande d'envoi du SMS.
`campaign_activity_id`	ID unique de l'activité au sein de la campagne.
`origination_phone_number`	Numéro de téléphone à partir duquel le message a été envoyé.
`destination_phone_number`	Numéro de téléphone auquel vous avez tenté d'envoyer le message.
`record_status`	Informations supplémentaires sur l'état du message. Les valeurs possibles incluent : • SUCCESSFUL/DELIVERED : le message a été correctement remis. • PENDING : le message n'a pas encore été remis à l'appareil du destinataire. • INVALID : le numéro de téléphone de destination n'est pas valide. • UNREACHABLE : l'appareil du destinataire est actuellement inaccessible ou indisponi ble. Par exemple, l'appareil peut être éteint ou déconnecté du réseau. Vous pouvez

Attribut	Description
	réessayer d'envoyer le message ultérieur ement. • UNKNOWN : une erreur s'est produite qui a empêché la remise du message. Cette erreur est généralement transitoire et vous pouvez essayer d'envoyer le message à nouveau ultérieurement. • BLOCKED : l'appareil du destinataire bloque les SMS provenant du numéro d'origine. • CARRIER_UNREACHABLE : un problème lié au réseau mobile du destinataire a empêché la remise du message. Cette erreur est généralement transitoire et vous pouvez essayer d'envoyer le message à nouveau ultérieurement. • SPAM : l'opérateur mobile du destinataire a identifié le contenu du message comme courrier indésirable et a bloqué la distribution du message. • INVALID_MESSAGE : le corps du message SMS n'est pas valide et ne peut pas être remis. • CARRIER_BLOCKED : l'opérateur du destinataire a bloqué la distribution de ce message. Cela se produit souvent lorsque le transporteur identifie le contenu du message comme indésirable ou malveillant. • TTL_EXPIRED : le SMS n'a pas pu être distribué dans un délai donné. Cette erreur est généralement transitoire et vous pouvez essayer d'envoyer le message à nouveau ultérieurement.

Attribut	Description
	• MAX_PRICE_EXCEEDED : l'envoi du message aurait entraîné des frais supérieurs au quota de dépenses mensuelles pour l'envoi de SMS pour votre compte. Vous pouvez demander une augmentation de ce quota en suivant la procédure décrite dans la section Demander une augmentation de votre quota mensuel de dépenses en SMS du Guide de l'utilisateur Amazon Pinpoint. • OPTED_OUT : le SMS n'a pas été envoyé car le destinataire a choisi de ne pas recevoir de messages de votre part. • NO_QUOTA_LEFT_ON_ACCOUNT : le quota de dépenses restant sur votre compte est insuffisant pour envoyer le message. Vous pouvez demander une augmentation de ce quota en suivant la procédure décrite dans la section Demande d'augmentation de votre quota mensuel de dépenses par SMS dans le Guide de l'utilisateur pour la messagerie SMS destinée aux utilisateurs AWS finaux. • NO_ORIGINATION_IDENTITY_AVAILABLE_TO_SEND : votre compte ne contient aucun numéro de téléphone pouvant être utilisé pour envoyer le message à destination. • DESTINATION_COUNTRY_NOT_SUPPORTED : le pays de destination est bloqué. Pour tous les pays pris en charge, consultez la section Pays et régions pris en charge (canal SMS) dans le Guide de l'utilisateur pour la messagerie SMS destinée aux utilisateurs AWS finaux.

Attribut	Description
	• ACCOUNT_IN_SANDBOX : votre compte est dans l'environnement de test (sandbox) et il peut uniquement envoyer des messages vers des numéros de destination vérifiés. Vous pouvez vérifier le numéro de destination dans la console Amazon Pinpoint ou lancer le processus pour déplacer le compte hors du sandbox. Consultez la section À propos du sandbox SMS/MMS et vocal dans le guide de l'utilisateur final pour la AWS messagerie SMS. • RATE_EXCEEDED : vous avez tenté d'envoyer un message trop rapidement et vous avez été limité. Vous devez ralentir votre débit d'appels. Pour en savoir plus sur nos limites, consultez la section Limites du nombre de parties de message par seconde (MPS) dans le Guide de l'utilisateur pour la messagerie SMS destinée aux utilisateurs AWS finaux. • INVALID_ORIGINATION_IDENTITY : l'identité d'origine fournie n'est pas valide. • ORIGINATION_IDENTITY_DOES_NOT_EXIST : l'identité d'origine fournie n'existe pas. • INVALID_DLT_PARAMETERS : des paramètres DLT non valides (obligatoires pour les destinations en Inde) ont été fournis. • INVALID_PARAMETERS : des paramètres non valides ont été fournis. • ACCESS_DENIED : l'envoi de messages est bloqué sur votre compte. Contactez le service client pour en déterminer la cause et résoudre le problème.

Attribut	Description
	• INVALID_KEYWORD : le mot clé fourni n'est pas valide. Il se peut que le format du mot clé soit incorrect ou qu'il ne soit pas défini dans votre compte. • INVALID_SENDER_ID : l'ID d'expéditeur fourni n'est pas valide. Le format ou la longueur de l'ID de l'expéditeur sont peut-être incorrects. • INVALID_POOL_ID : l'ID de pool fourni n'est pas valide. Il se peut que le format de l'ID du pool soit incorrect ou qu'il n'appartienne pas à votre compte. • SENDER_ID_NOT_SUPPORTED_FOR _DESTINATION : le pays de destination ne prend pas en charge l'ID de l'expéditeur. Vous devez utiliser un numéro de téléphone ou une autre identité d'origine pour l'envoi. • INVALID_PHONE_NUMBER : le numéro de téléphone d'origine fourni n'est pas valide. Le format ou la longueur du numéro de téléphone peuvent être incorrects.
`iso_country_code`	Pays associé au numéro de téléphone du destinataire, affiché au format ISO 3166-1 alpha-2.
`treatment_id`	ID du traitement du message, si le message a été envoyé dans une campagne A/B.
`treatment_id`	Si le message a été envoyé à l'aide d'une campagne de test A/B, cette valeur représente le numéro de traitement du message. Pour les SMS transactionnels, cette valeur est 0.

Attribut	Description
`number_of_message_parts`	Nombre de parties de message créées par Amazon Pinpoint pour envoyer le message. En général, les SMS ne peuvent contenir que 160 caractères GSM-7 ou 67 caractères autres que GSM, même si ces limites peuvent varier selon le pays . Si vous envoyez un message qui dépasse ces limites, Amazon Pinpoint fractionne automatiquement les messages en plusieurs parties. Nous vous facturons en fonction du nombre de parties du message que vous envoyez.
`message_id`	ID unique généré par Amazon Pinpoint lorsqu'il accepte le message.
`message_type`	Type du message. Les valeurs possibles sont Promotional et Transactional. Vous spécifiez cette valeur lorsque vous créez une campagne ou lorsque vous envoyez des messages transactionnels à l'aide de l'SendMessagesopération de l'API Amazon Pinpoint.
`campaign_id`	ID unique de la campagne Amazon Pinpoint qui a envoyé le message.
`customer_context`	Chaîne JSON du contenu de la `Context` carte envoyée dans le cadre d'une opération Amazon Pinpoint SendMessages.

Client

Présente des informations sur le client d'application installé sur l'appareil qui a signalé l'événement.

Attribut	Description
`client_id`	Pour les événements générés par les applications, cette valeur est l'ID unique du client d'application installé sur l'appareil. Cet identifiant est automatiquement généré par le Applications mobiles AWS SDK for iOS et le AWS Mobile SDK pour Android. Pour les événements qui sont générés lorsque vous envoyez des campagnes et des messages transactionnels, cette valeur est égale à l'ID du point de terminaison auquel vous avez envoyé le message.
`cognito_id`	ID unique attribué au client d'application dans la réserve d'identités Amazon Cognito utilisée par votre application.

Appareil

Présente des informations sur l'appareil qui a signalé l'événement.

Attribut	Description
`locale`	Paramètres régionaux du périphérique.
`make`	Fabricant du périphérique, tel qu'`Apple` ou `Samsung`.
`model`	Modèle du périphérique, tel qu'`iPhone`.
`platform`	Plateforme du périphérique, telle qu'`ios` ou `android`.

Métriques

Comprend des informations sur les métriques associées à l'événement.

Attribut	Description
`price_in_millicents_usd`	Montant facturé pour l'envoi du message. Ce prix est indiqué en millièmes d'un cent des États-Unis. Par exemple, si la valeur de cet attribut est 645, nous vous avons facturé 0,645 ¢ pour envoyer le message (645 / 1000 = 0,645 ¢ = 0,00645 $). **Note** Cette propriété n'apparaît pas pour les messages ayant un `event_type` _SMS.BUFFERED.

Supprimer un flux d'événements d'Amazon Pinpoint

Si vous avez attribué un flux Kinesis à une application, vous pouvez désactiver la diffusion en continu des événements pour cette application. Amazon Pinpoint arrête de diffuser les événements vers Kinesis, mais vous pouvez consulter les analyses des événements à l'aide de la console Amazon Pinpoint.

AWS CLI

Utilisez la commande `delete-event-stream` :

```
aws pinpoint delete-event-stream --application-id application-id
```

AWS SDK pour Java

Utilisez la `deleteEventStream`méthode du client Amazon Pinpoint :

```
pinClient.deleteEventStream(new DeleteEventStreamRequest().withApplicationId(appId));
```

Interrogez les données d'analyse Amazon Pinpoint

Outre les pages d'analyse de la console Amazon Pinpoint, vous pouvez utiliser Amazon Pinpoint APIs Analytics pour interroger les données d'analyse d'un sous-ensemble de mesures standard qui fournissent des informations sur les tendances liées à l'engagement des utilisateurs, à la portée des campagnes, etc. Ces indicateurs, également appelés indicateurs de performance clés (KPIs), sont des valeurs mesurables qui peuvent vous aider à surveiller et à évaluer les performances de vos projets, campagnes et parcours.

Si vous utilisez le APIs pour interroger des données d'analyse, vous pouvez analyser les données à l'aide de l'outil de reporting de votre choix, sans avoir à vous connecter à la console Amazon Pinpoint ou à analyser les données d'événements brutes provenant de sources telles que les flux Amazon Kinesis. Par exemple, vous pouvez créer un tableau de bord personnalisé qui affiche les résultats hebdomadaires des campagnes ou fournit des analyses approfondies des taux de diffusion de vos campagnes.

Vous pouvez interroger les données à l'aide de l'API REST Amazon Pinpoint, du AWS Command Line Interface (AWS CLI) ou d'un AWS SDK. Pour interroger les données, vous envoyez une demande à l'API Amazon Pinpoint et utilisez les paramètres pris en charge pour spécifier les données que vous voulez et les filtres que vous souhaitez appliquer. Une fois que vous avez soumis votre requête, Amazon Pinpoint renvoie les résultats de la requête dans une réponse JSON. Vous pouvez alors transmettre les résultats à un autre service ou une autre application pour une analyse plus approfondie, ou encore à des fins de stockage ou de génération de rapports.

Amazon Pinpoint collecte et regroupe automatiquement les données pour tous les types de métriques pris en charge ainsi que pour l'ensemble de vos projets, campagnes et parcours. En outre, les données sont mises à jour de façon continue, ce qui se traduit par une période de latence des données, limitée à deux heures environ. Notez toutefois qu'il peut y avoir une latence supplémentaire des données pour certaines métriques. En effet, les données de certaines métriques sont basées sur des informations reçues de la part des fournisseurs de messagerie des destinataires. Certains fournisseurs nous envoient ces informations immédiatement, tandis que d’autres les envoient plus rarement.

Amazon Pinpoint stocke les données pendant 90 jours. Pour stocker les données pendant plus de 90 jours ou pour accéder aux données analytiques brutes en temps réel, vous pouvez configurer un projet Amazon Pinpoint afin de diffuser les données d'événements vers Amazon Kinesis Data Streams ou Amazon Data Firehose. Pour plus d'informations sur la configuration des flux

d'événements, consultez Diffusez les données d'événements des applications via Kinesis et Firehose à l'aide d'Amazon Pinpoint.

Composants et paramètres de requête pour les métriques dans Amazon Pinpoint

Pour interroger les données d'une métrique, vous envoyez une demande `get` à la ressource de métrique appropriée de l'API Amazon Pinpoint. Dans votre demande, vous définissez votre requête en utilisant les paramètres pris en charge pour les composants de requête suivants :

- Projet : spécifiez un projet en fournissant l'ID du projet comme valeur du paramètre `application-id`. Ce paramètre est requis pour toutes les métriques.
- Campagne : spécifiez une campagne en fournissant l'ID de campagne comme valeur du paramètre `campaign-id`. Ce paramètre n'est requis que pour les métriques de campagne.
- Parcours : spécifiez un parcours en fournissant l'ID du parcours en tant que valeur du paramètre `journey-id`. Ce paramètre est obligatoire uniquement pour les métriques d'engagement et d'exécution du parcours, ainsi que pour les mesures d'exécution de l'activité du parcours.
- Activité du parcours : spécifiez une activité du parcours en fournissant l'ID d'activité du parcours en tant que valeur du paramètre `journey-activity-id`. Ce paramètre est obligatoire uniquement pour les métriques d'exécution de l'activité du parcours.
- Plage de dates : pour filtrer éventuellement les données par plage de dates, indiquez la première et la dernière date et heure de la plage de dates à l'aide des paramètres d'heure de début et de fin pris en charge. Les valeurs doivent être au format ISO 8601 étendu et utiliser le temps universel coordonné (UTC) : par exemple, `2019-07-19T20:00:00Z` pour 20h00 UTC le 19 juillet 2019.

  Notez que les plages de dates sont inclusives et doivent être limitées à 31 jours calendaires au plus. En outre, les premières date et heure doivent être inférieures à 90 jours à compter du jour actuel. Si vous ne spécifiez pas de plage de dates, Amazon Pinpoint renvoie les données des 31 jours civils précédents. Les paramètres de plage de dates sont pris en charge par toutes les métriques, à l'exception des métriques d'exécution du parcours et des mesures d'exécution de l'activité du parcours.
- Métrique : spécifiez la métrique en fournissant son nom comme valeur du paramètre `kpi-name`. Cette valeur décrit la métrique associée et se compose de deux termes ou plus, composés de caractères alphanumériques minuscules, séparés par un trait d'union. Par exemple, `email-open-rate` et `successful-delivery-rate`. Ce paramètre est obligatoire uniquement pour toutes les métriques, à l'exception des métriques d'exécution du parcours et des mesures d'exécution

de l'activité du parcours. Pour obtenir la liste complète des métriques prises en charge et la valeur `kpi-name` à utiliser pour chacune d'elles, consultez Mesures standard pour les projets, les campagnes et les parcours.

Une fois que vous avez envoyé votre requête, Amazon Pinpoint renvoie les résultats de la requête dans une réponse JSON. Dans la réponse, la structure des résultats varie en fonction de la métrique que vous avez interrogée.

Certaines métriques ne fournissent qu'une seule valeur : par exemple, le nombre de messages envoyés par une campagne. D'autres métriques fournissent plusieurs valeurs et ces valeurs sont généralement regroupées selon un champ pertinent : par exemple, le nombre de messages remis par chaque exécution d'une campagne, regroupés par exécution de campagne. Si une métrique fournit et regroupe plusieurs valeurs, la réponse JSON inclut un champ qui indique celui utilisé pour regrouper les données. Pour en savoir plus sur la structure des résultats de requête, reportez-vous à la section Utiliser les résultats des requêtes JSON.

Politiques IAM pour l'interrogation de données analytiques Amazon Pinpoint

À l'aide de l'API Amazon Pinpoint, vous pouvez interroger les données d'analyse pour un sous-ensemble de mesures standard, également appelées indicateurs de performance clés (KPIs) qui s'appliquent aux projets, campagnes et parcours Amazon Pinpoint. Ces métriques peuvent vous aider à surveiller et à évaluer les performances des projets, campagnes et parcours.

Pour gérer l'accès à ces données, vous pouvez créer des politiques Gestion des identités et des accès AWS (IAM) qui définissent les autorisations pour les rôles IAM ou les utilisateurs autorisés à accéder aux données. Pour prendre en charge le contrôle granulaire de l'accès à ces données, Amazon Pinpoint fournit plusieurs actions distinctes que vous pouvez spécifier dans les politiques IAM. Il existe une action distincte pour afficher les données analytiques sur la console Amazon Pinpoint (`mobiletargeting:GetReports`), et il existe d'autres actions pour accéder aux données analytiques par programmation à l'aide de l'API Amazon Pinpoint.

Pour créer des politiques IAM qui gèrent l'accès aux données d'analyse, vous pouvez utiliser l' AWS Management Console API AWS CLI, ou l'API IAM. Notez que l'onglet Éditeur visuel de la AWS Management Console n'inclut pas actuellement les actions permettant d'afficher ou d'interroger les données analytiques Amazon Pinpoint. Toutefois, vous pouvez ajouter manuellement les actions nécessaires aux stratégies IAM à l'aide de l'onglet JSON de la console.

Par exemple, la politique suivante autorise l'accès programmatique à toutes les données d'analyse pour tous vos projets, campagnes et voyages dans toutes les AWS régions :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryAllAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:*:111122223333:apps/*/kpis/*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/campaigns/*/kpis/
*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/kpis/
*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/
execution-metrics",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/
activities/*/execution-metrics"
            ]
        }
    ]
}
```

Où se *accountId* trouve votre identifiant de AWS compte ?

Cependant, une bonne pratique consiste à créer des stratégies qui suivent le principe du moindre privilège. En d'autres termes, vous devez créer des stratégies qui incluent uniquement les autorisations requises pour effectuer une tâche spécifique. Pour soutenir cette pratique et mettre en œuvre un contrôle plus précis, vous pouvez restreindre l'accès programmatique aux données analytiques uniquement pour un projet spécifique dans une AWS région spécifique, par exemple :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryProjectAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/kpis/
*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/campaigns/*/
kpis/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
kpis/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
execution-metrics",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
activities/*/execution-metrics"
            ]
        }
    ]
}
```

Où :

- *region*est le nom de la AWS région qui héberge le projet.
- *accountId*est votre identifiant de AWS compte.
- *projectId*est l'identifiant du projet auquel vous souhaitez donner accès.

De même, l'exemple de stratégie suivant autorise l'accès par programmation aux données analytiques pour une campagne donnée uniquement :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryCampaignAnalytics",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetCampaignDateRangeKpi",
            "Resource": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/projectId/campaigns/campaignId/kpis/*"
        }
    ]
}
```

Où :

- *region*est le nom de la AWS région qui héberge le projet.
- *accountId*est votre Compte AWS identifiant.
- *projectId*est l'identifiant du projet associé à la campagne.
- *campaignId*est l'identifiant de la campagne à laquelle vous souhaitez donner accès.

Et l'exemple de stratégie suivant permet l'accès programmatique à toutes les données d'analyse, les données d'engagement et d'exécution, pour un parcours particulier et les activités qui composent ce parcours :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryJourneyAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
```

```
            ],
            "Resource": [
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/kpis/*",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/execution-metrics",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/activities/*/execution-metrics"
            ]
        }
    ]
}
```

Où :

- *region*est le nom de la AWS région qui héberge le projet.
- *accountId*est votre identifiant de AWS compte.
- *projectId*est l'identifiant du projet associé au voyage.
- *journeyId*est l'identifiant du trajet auquel vous souhaitez donner accès.

Pour obtenir la liste complète des actions d'API Amazon Pinpoint que vous pouvez utiliser dans les politiques IAM, reportez-vous à la section Actions Amazon Pinpoint pour les politiques IAM. Pour plus d'informations sur la création et la gestion de politiques IAM, consultez le Guide de l'utilisateur IAM.

Mesures standard qui s'appliquent aux projets, campagnes et parcours Amazon Pinpoint

Vous pouvez utiliser les API analytiques Amazon Pinpoint pour interroger les données analytiques pour un sous-ensemble de métriques standard qui s'appliquent aux projets, campagnes et parcours Amazon Pinpoint. Ces métriques, également appelées indicateurs de performance clés (KPI), sont des valeurs mesurables qui peuvent vous aider à surveiller et évaluer les performances des projets, campagnes et parcours.

Amazon Pinpoint fournit un accès programmatique aux données analytiques pour plusieurs types de métriques standard :

- Métriques d'application : ces métriques fournissent des informations sur les tendances pour les campagnes et les messages transactionnels associés à un projet, également appelé application.

Par exemple, vous pouvez utiliser une métrique d'application pour voir le détail du nombre de messages de campagne ayant été ouverts par les destinataires pour chaque campagne associée à un projet.

- Métriques de campagne : ces métriques fournissent des informations sur les performances de chaque campagne. Par exemple, vous pouvez utiliser une métrique de campagne pour consulter les données sur le nombre de points de terminaison auxquels un message de campagne a été envoyé ou le nombre de ces messages qui ont été remis aux points de terminaison ciblés.
- Métriques d'engagement du parcours : ces métriques offrent un aperçu des performances des parcours individuels. Par exemple, vous pouvez utiliser une métrique d'engagement du parcours pour obtenir le détail du nombre de messages ouverts par les participants dans chaque activité d'un parcours.
- Métriques d'exécution des parcours : ces métriques fournissent un aperçu des tendances de participation pour les différents parcours. Par exemple, vous pouvez utiliser une métrique d'exécution du parcours pour déterminer le nombre de participants commençant une activité.
- Métriques d'exécution des activités de parcours : ces métriques fournissent un aperçu des tendances de participation aux activités individuelles d'un parcours. Par exemple, vous pouvez utiliser une métrique d'exécution du parcours pour déterminer le nombre de participants ayant commencé une activité et le nombre ayant terminé chaque chemin d'une activité.

Les rubriques de cette section répertorient et décrivent les mesures individuelles que vous pouvez interroger pour chaque type de métrique.

Rubriques

- Statistiques relatives aux applications Amazon Pinpoint pour les campagnes
- Statistiques de l'application Amazon Pinpoint pour les e-mails transactionnels
- Statistiques de l'application Amazon Pinpoint pour les SMS transactionnels
- Statistiques de campagne Amazon Pinpoint
- Indicateurs d'engagement liés au parcours Amazon Pinpoint
- Indicateurs d'exécution du parcours Amazon Pinpoint
- Mesures d'exécution des activités liées au parcours Amazon Pinpoint
- Statistiques relatives au parcours et à l'exécution des campagnes Amazon Pinpoint

Statistiques relatives aux applications Amazon Pinpoint pour les campagnes

Le tableau suivant répertorie et décrit les métriques d'application standard que vous pouvez interroger pour évaluer les tendances de toutes les campagnes associées à un projet Amazon Pinpoint. Afin d'interroger les données pour ces métriques, utilisez la ressource Métriques d'application de l'API Amazon Pinpoint. La colonne kpi-name de la table indique la valeur à utiliser pour le paramètre `kpi-name` de votre requête.

Métrique	Kpi-name	Description
Delivery rate (Taux de remise)	`successful-delivery-rate`	Pour toutes les campagnes associées à un projet, pourcentage de messages ayant été remis aux destinataires. Cette métrique est calculée de la façon suivante : le nombre de messages qui ont été envoyés par l'ensemble des campagnes pour un projet et remis aux destinataires, divisé par le nombre de messages qui ont été envoyés par l'ensemble de ces campagnes.
Delivery rate, grouped by date (Taux de remise, regroupement par date)	`successful-delivery-rate-grouped-by-date`	Pour toutes les campagnes associées à un projet, pourcentage de messages ayant été remis aux destinataires pour chaque jour de la plage de dates spécifiée. Cette métrique est calculée de la façon suivante : le nombre de messages envoyés par toutes les campagnes pour un projet et remis aux destinata

Métrique	Kpi-name	Description
		ires, divisé par le nombre de messages qui ont été envoyés par l'ensemble de ces campagnes, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Email open rate (Taux d’ouverture d’e-mail)	`email-open-rate`	Pour toutes les campagnes associées à un projet, pourcentage d'e-mails ayant été ouverts par les destinata ires. Cette métrique est calculée de la façon suivante : le nombre d'e-mails envoyés par toutes les campagnes pour un projet et ouverts par les destinata ires, divisé par le nombre d'e-mails qui ont été envoyés par l'ensemble de ces campagnes et remis aux destinataires.

Métrique	Kpi-name	Description
Email open rate, grouped by campaign (Taux d'ouverture d'e-mail, regroupement par campagne)	`email-open-rate-grouped-by-campaign`	Pour chaque campagne associée à un projet, pourcentage d'e-mails ayant été ouverts par les destinataires. Cette métrique est calculée de la façon suivante : le nombre d'e-mails envoyés par une campagne et ouverts par les destinataires, divisé par le nombre d'e-mails qui ont été envoyés par la campagne et remis aux destinataires. Les résultats de la requête pour cette métrique sont regroupés par ID de campagne (`CampaignId`), une chaîne qui identifie de façon unique une campagne.
Endpoint deliveries (Remises au point de terminaison)	`unique-deliveries`	Pour toutes les campagnes associées à un projet, nombre de points de terminaison uniques auxquels les messages ont été envoyés.

Métrique	Kpi-name	Description
Endpoint deliveries, grouped by campaign (Remises au point de terminaison, regroupement par campagne)	`unique-deliveries-grouped-by-campaign`	Pour chaque campagne associée à un projet, nombre de points de terminaison uniques auxquels les messages ont été envoyés. Les résultats de la requête pour cette métrique sont regroupés par ID de campagne (`CampaignId`), une chaîne qui identifie de façon unique une campagne.
Endpoint deliveries, grouped by date (Remises au point de terminaison, regroupement par date)	`unique-deliveries-grouped-by-date`	Pour toutes les campagnes associées à un projet, nombre de points de terminaison uniques auxquels les messages ont été envoyés, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Métrique	Kpi-name	Description
Messages delivered, grouped by campaign (Messages remis, regroupement par campagne)	`successful-deliveries-grouped-by-campaign`	Pour chaque campagne associée à un projet, nombre de messages ayant été remis aux destinataires. Cette métrique est calculée de la façon suivante : le nombre de messages envoyés par une campagne, moins le nombre de messages ayant été envoyés par la campagne et qui n'ont pas été remis aux destinataires du fait d'un message d'erreur définitif. Les résultats de la requête pour cette métrique sont regroupés par ID de campagne (`CampaignId`), une chaîne qui identifie de façon unique une campagne.

Métrique	Kpi-name	Description
Push open rate (Taux d’ouverture des notifications push)	`push-open-rate`	Pour toutes les campagnes associées à un projet, pourcentage de notifications push ayant été ouvertes par les destinataires. Cette métrique est calculée de la façon suivante : le nombre de notifications push envoyées par toutes les campagnes pour un projet et ouvertes par les destinataires, divisé par le nombre de notifications push qui ont été envoyées par l'ensemble de ces campagnes et remises aux destinataires.

Métrique	Kpi-name	Description
Push open rate, grouped by campaign (Taux d'ouverture de notifications push, regroupement par campagne)	`push-open-rate-grouped-by-campaign`	Pour chaque campagne associée à un projet, pourcentage de notifications push ayant été ouvertes par les destinataires. Cette métrique est calculée de la façon suivante : le nombre de notifications push envoyées par une campagne et ouvertes par les destinataires, divisé par le nombre de notifications push qui ont été envoyées par la campagne remises aux destinataires. Les résultats de la requête pour cette métrique sont regroupés par ID de campagne (`CampaignId`), une chaîne qui identifie de façon unique une campagne.

Statistiques de l'application Amazon Pinpoint pour les e-mails transactionnels

Le tableau suivant répertorie et décrit les métriques d'application standard que vous pouvez interroger pour en savoir plus sur les tendances de tous les e-mails transactionnels associés à un projet Amazon Pinpoint. Afin d'interroger les données pour ces métriques, utilisez la ressource Métriques d'application de l'API Amazon Pinpoint. La colonne kpi-name de la table indique la valeur à utiliser pour le paramètre `kpi-name` de votre requête.

Notez que ces métriques ne fournissent pas de données sur les e-mails envoyés par les campagnes. Elles fournissent des données sur les e-mails transactionnels uniquement. Pour interroger les

données pour les messages envoyés par une ou plusieurs campagnes, utilisez une métrique de campagne ou une métrique d'application pour les campagnes.

Métrique	Kpi-name	Description
Clics	`txn-emails-clicked`	Nombre de fois où les destinataires ont cliqué sur les liens dans les messages. Si un seul destinataire a cliqué sur plusieurs liens dans un message, ou s'il a cliqué sur le même lien plusieurs fois, chaque clic est inclus dans le total.
Clics, regroupés par date	`txn-emails-clicked-grouped-by-date`	Nombre de fois où les destinataires ont cliqué sur des liens dans les messages, pour chaque jour de la plage de dates spécifiée. Si un seul destinataire a cliqué sur plusieurs liens dans un message, ou s'il a cliqué sur le même lien plusieurs fois, chaque clic est inclus dans le total. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Complaint rate (Taux de réclamations)	`txn-emails-complaint-rate`	Pourcentage de messages signalés par les destinataires comme des e-mails non souhaités ou indésirables.

Métrique	Kpi-name	Description
		Cette métrique est calculée en fonction du nombre de messages signalés par les destinataires comme étant des e-mails non souhaités ou indésirables, divisé par le nombre de messages envoyés.
Taux de réclamations, regroupé par date	`txn-emails-complaint-rate-grouped-by-date`	Pourcentage de messages signalés par les destinataires comme des e-mails non souhaités ou indésirables, pour chaque jour de la plage de dates spécifiée. Cette métrique est calculée en fonction du nombre de messages signalés par les destinataires comme étant des e-mails non souhaités ou indésirables divisé par le nombre de messages envoyés, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Réclamations	`txn-emails-with-complaints`	Nombre de messages signalés par les destinataires comme des e-mails non souhaités ou indésirables.

Métrique	Kpi-name	Description
Réclamations, regroupées par date	`txn-emails-with-complaints-grouped-by-date`	Nombre de messages signalés par les destinataires comme des e-mails non souhaités ou indésirables, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Messages délivrés	`txn-emails-delivered`	Nombre de messages remis à leurs destinataires prévus. Cette métrique est calculée en fonction du nombre de messages envoyés moins le nombre de messages qui n'ont pas pu être remis en raison d'une non-remise temporaire ou définitive, ou parce qu'ils ont été rejetés. Un message est rejeté si Amazon Pinpoint détermine qu'il contient un logiciel malveillant. Amazon Pinpoint n'essaie pas d'envoyer des messages rejetés.

Métrique	Kpi-name	Description
Livraisons, regroupées par date	`txn-emails-delivered-grouped-by-date`	Nombre de messages remis aux destinataires, pour chaque jour de la plage de dates spécifiée. Cette métrique est calculée en fonction du nombre de messages envoyés moins le nombre de messages qui n'ont pas pu être remis en raison d'une non-remise temporaire ou définitive, ou parce qu'ils ont été rejetés, pour chaque jour de la plage de dates spécifiée. Un message est rejeté si Amazon Pinpoint détermine qu'il contient un logiciel malveillant. Amazon Pinpoint n'essaie pas d'envoyer des messages rejetés. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Métrique	Kpi-name	Description
Delivery rate (Taux de remise)	`txn-emails-delivery-rate`	Pourcentage de messages remis aux destinataires. Cette métrique est calculée en fonction du nombre de messages envoyés et remis aux destinataires, divisé par le nombre de messages envoyés.
Delivery rate, grouped by date (Taux de remise, regroupement par date)	`txn-emails-delivery-rate-grouped-by-date`	Pourcentage de messages remis aux destinataires, pour chaque jour de la plage de dates spécifiée. Cette métrique est calculée en fonction du nombre de messages envoyés et remis aux destinataires, divisé par le nombre de messages envoyés, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Métrique	Kpi-name	Description
Messages d'erreur définitifs	`txn-emails-hard-bounced`	Nombre de messages qui n'ont pas pu être remis aux destinataires en raison d'une non-remise définitive. Un message d'erreur définitif est renvoyé si un problème persistant empêche la remise d'un message, par exemple, si l'adresse e-mail d'un destinataire n'existe pas.
Non-remises définitives, regroupées par date	`txn-emails-hard-bounced-grouped-by-date`	Nombre de messages qui n'ont pas pu être remis aux destinataires en raison d'une non-remise définitive, pour chaque jour de la plage de dates spécifiée. Un message d'erreur définitif est renvoyé si un problème persistant empêche la remise d'un message, par exemple, si l'adresse e-mail d'un destinataire n'existe pas. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Messages ouverts	`txn-emails-opened`	Nombre de messages ouverts par les destinataires.

Métrique	Kpi-name	Description
Messages ouverts, regroupés par date	`txn-emails-opened-grouped-by-date`	Nombre de messages ouverts par les destinataires, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Envois	`txn-emails-sent`	Nombre de messages envoyés.
Envois, regroupés par date	`txn-emails-sent-grouped-by-date`	Nombre de messages envoyés, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Erreurs de remise temporaire	`txn-emails-soft-bounced`	Nombre de messages qui n'ont pas pu être remis aux destinataires en raison d'une erreur de remise temporaire. Un message d'erreur temporaire est renvoyé lorsqu'un problème temporaire empêche la remise d'un message, par exemple si la boîte de réception du destinataire est pleine ou si le serveur de réception est temporairement indisponible.

Métrique	Kpi-name	Description
Erreurs de remise temporaire, regroupées par date	`txn-emails-soft-bounced-grouped-by-date`	Nombre de messages qui n'ont pas pu être remis aux destinataires en raison d'une erreur de remise temporaire, pour chaque jour de la plage de dates spécifiée. Un message d'erreur temporaire est renvoyé lorsqu'un problème temporaire empêche la remise d'un message, par exemple si la boîte de réception du destinataire est pleine ou si le serveur de réception est temporairement indisponible. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Métrique	Kpi-name	Description
Événements de clics utilisateur uniques	`txn-emails-unique-clicks`	Nombre de destinataires uniques (points de terminaison) ayant cliqué sur des liens dans des messages. Contrairement à la métrique Clicks (Clics), cette métrique indique le nombre de destinataires uniques ayant cliqué sur des liens, et non le nombre d'événements de clics qui se sont produits. Par exemple, si un seul destinataire a cliqué sur plusieurs liens dans le même message ou cliqué sur le même lien plusieurs fois, cette métrique ne rapporte qu'un seul événement de clic pour ce destinataire.

Métrique	Kpi-name	Description
Événements de clics utilisateur uniques, regroupés par date	`txn-emails-unique-clicks-grouped-by-date`	Nombre de destinataires uniques (points de terminaison) ayant cliqué sur des liens dans des messages, pour chaque jour de la plage de dates spécifiée. Contrairement à la métrique Clicks, grouped by date (Clics, regroupés par date), cette métrique indique le nombre de destinataires uniques ayant cliqué sur des liens, et non le nombre d'événements de clic qui se sont produits. Par exemple, si un seul destinataire a cliqué sur plusieurs liens dans le même message ou cliqué sur le même lien plusieurs fois, cette métrique ne rapporte qu'un seul événement de clic pour ce destinataire. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Métrique	Kpi-name	Description
Événements d'ouverture utilisateur uniques	`txn-emails-unique-opens`	Nombre de destinataires uniques (points de terminaison) ayant ouvert des messages. Contrairement à la métrique Ouvertures, cette métrique indique le nombre de destinataires uniques ayant ouvert des messages, et non le nombre d'événements d'ouverture qui se sont produits. Par exemple, si un seul destinataire ouvre le même message plusieurs fois, cette métrique indique un seul événement d'ouverture pour ce destinataire.

Métrique	Kpi-name	Description
Événements d'ouverture utilisateur uniques, regroupés par date	`txn-emails-unique-opens-grouped-by-date`	Nombre de destinataires uniques (points de terminaison) ayant ouvert des messages, pour chaque jour de la plage de dates spécifiée. Contrairement à la métrique Ouvertures, regroupées par date, cette métrique indique le nombre de destinataires uniques ayant ouvert des messages, et non le nombre d'événements d'ouverture qui se sont produits. Par exemple, si un seul destinataire ouvre le même message plusieurs fois, cette métrique indique un seul événement d'ouverture pour ce destinataire. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Statistiques de l'application Amazon Pinpoint pour les SMS transactionnels

Le tableau suivant répertorie et décrit les métriques d’application standard que vous pouvez interroger pour surveiller les tendances de tous les SMS transactionnels associés à un projet Amazon Pinpoint. Afin d'interroger les données pour ces métriques, utilisez la ressource Métriques d'application de l'API Amazon Pinpoint. La colonne kpi-name de la table indique la valeur à utiliser pour le paramètre `kpi-name` de votre requête.

Notez que ces métriques ne fournissent pas de données sur les SMS envoyés par les campagnes. Elles fournissent des données sur les SMS transactionnels uniquement. Pour interroger les données

pour les messages envoyés par une ou plusieurs campagnes, utilisez une métrique de campagne ou une métrique d'application pour les campagnes.

Métrique	Kpi-name	Description
Prix moyen par message, regroupé par pays	`txn-sms-average-price-grouped-by-country`	Coût moyen de l'envoi de chaque message, pour chaque pays ou région où les messages ont été envoyés. Le prix est indiqué en millièmes d'un cent des États-Unis. Par exemple, si la valeur de cet attribut est 645, nous vous avons facturé 0,645 ¢ pour envoyer le message (645 / 1000 = 0,645 ¢ = 0,00645 $). Cette métrique est calculée comme le coût total de tous les messages envoyés aux destinataires dans chaque pays ou région, divisé par le nombre de messages envoyés aux destinataires dans chacun de ces pays et régions. Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.
Prix moyen par partie de message, regroupé par pays	`txn-sms-average-price-by-parts-grouped-by-country`	Coût moyen de l'envoi de chaque partie d'un message, pour chaque pays ou région où les messages ont été envoyés. Une partie de message représente une partie d'un message SMS. Le

Métrique	Kpi-name	Description
		prix est indiqué en millièmes d'un cent des États-Unis. Par exemple, si la valeur de cet attribut est 645, nous vous avons facturé 0,645 ¢ pour envoyer le message (645 / 1000 = 0,645 ¢ = 0,00645 $). Cette métrique est calculée comme le coût total de toutes les parties de message envoyées aux destinataires dans chaque pays ou région, divisé par le nombre de parties de message envoyées aux destinataires dans chacun de ces pays et régions. Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.
Messages délivrés	`txn-sms-delivered`	Nombre de messages remis à leurs destinataires prévus.
Livraisons, regroupées par pays	`txn-sms-delivered-grouped-by-country`	Nombre de messages remis aux destinataires, pour chaque pays ou région où les messages ont été envoyés. Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.

Métrique	Kpi-name	Description
Livraisons, regroupées par date	`txn-sms-delivered-grouped-by-date`	Nombre de messages remis aux destinataires, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Erreurs de livraison	`txn-sms-error-distribution`	Nombre de fois où une erreur s'est produite lors des tentatives de livraison des messages, pour chaque type d'erreur qui s'est produite. Les résultats de la requête pour cette métrique sont regroupés par code d'erreur, pour chaque type d'erreur qui s'est produite.
Delivery rate (Taux de remise)	`txn-sms-delivery-rate`	Pourcentage de messages remis aux destinataires. Cette métrique est calculée en fonction du nombre de messages envoyés et remis aux destinataires, divisé par le nombre de messages envoyés.

Métrique	Kpi-name	Description
Delivery rate, grouped by date (Taux de remise, regroupement par date)	`txn-sms-delivery-rate-grouped-by-date`	Pourcentage de messages remis aux destinataires, pour chaque jour de la plage de dates spécifiée. Cette métrique est calculée en fonction du nombre de messages envoyés et remis aux destinataires, divisé par le nombre de messages envoyés, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.
Parties de message remises	`txn-sms-delivered-by-parts`	Nombre de parties de message remises. Une partie de message représente une partie d'un message SMS. Si un SMS contient plus de caractères que le protocole SMS ne le permet, Amazon Pinpoint divise le message en autant de parties que nécessaire pour l'envoyer à un destinataire.

Métrique	Kpi-name	Description
Parties de message remises, regroupées par pays	`txn-sms-delivered-by-parts-grouped-by-country`	Nombre de parties de message remises, pour chaque pays ou région où les messages ont été envoyés. Une partie de message représente une partie d'un message SMS. Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.
Parties de message envoyées	`txn-sms-sent-by-parts`	Nombre de parties de message envoyées. Une partie de message représente une partie d'un message SMS. Si un SMS contient plus de caractères que le protocole SMS ne le permet, Amazon Pinpoint divise le message en autant de parties que nécessaire pour l'envoyer à un destinataire.

Métrique	Kpi-name	Description
Parties de message envoyées, regroupées par pays	`txn-sms-sent-by-parts-grouped-by-country`	Nombre de parties de message envoyées, pour chaque pays ou région où les messages ont été envoyés. Une partie de message représente une partie d'un message SMS. Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.
Messages sent (Messages envoyés)	`txn-sms-sent`	Nombre de messages envoyés.
Messages envoyés, regroupés par pays	`txn-sms-sent-grouped-by-country`	Nombre de messages envoyés, pour chaque pays ou région où les messages ont été envoyés. Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.
Messages envoyés, regroupés par date	`txn-sms-sent-grouped-by-date`	Nombre de messages envoyés, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Métrique	Kpi-name	Description
Prix total, regroupé par pays	`txn-sms-total-price-grouped-by-country`	Coût total de l'envoi des messages, pour chaque pays ou région où les messages ont été envoyés. Le prix est indiqué en millièmes d'un cent des États-Unis. Par exemple, si la valeur de cet attribut est 645, nous vous avons facturé 0,645 ¢ pour envoyer le message (645 / 1000 = 0,645 ¢ = 0,00645 $). Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.

Statistiques de campagne Amazon Pinpoint

Le tableau suivant répertorie et décrit les métriques de campagne standard que vous pouvez interroger pour en savoir plus sur les tendances d'une campagne individuelle. Pour interroger les données pour ces métriques, vous utilisez la ressource Métriques de campagne de l'API Amazon Pinpoint. Dans la table, la colonne kpi-name indique la valeur à utiliser pour le paramètre `kpi-name` de votre requête.

Métrique	Kpi-name	Description
Bounce rate (Taux de retours à l’expéditeur)	`hard-bounce-rate`	Pour l'ensemble des exécutions de campagnes, pourcentage d'e-mails n'ayant pas pu être remis aux destinataires. Cette métrique mesure uniquement les messages d'erreur définitifs, c'est-à-dire

Métrique	Kpi-name	Description
		les messages dans lesquels l'adresse e-mail du destinataire a rencontré un problème permanent ayant empêché la remise du message. Cette métrique est calculée de la façon suivante : le nombre d'e-mails non remis qui ont été envoyés par toutes les exécutions de campagnes, divisé par le nombre d'e-mails qui ont été envoyés par l'ensemble de ces exécutions de campagnes.

Métrique	Kpi-name	Description
Bounce rate, grouped by campaign (Taux de retours à l'expéditeur, regroupement par campagne)	`hard-bounce-rate-grouped-by-campaign-activity`	Pour chaque exécution de campagne, pourcentage d'e-mails n'ayant pas pu être remis aux destinataires. Cette métrique mesure uniquement les messages d'erreur définitifs, c'est-à-dire les messages dans lesquels l'adresse e-mail du destinataire a rencontré un problème permanent ayant empêché la remise du message. Cette métrique est calculée de la façon suivante : le nombre d'e-mails non remis qui ont été envoyés par une exécution de campagne, divisé par le nombre d'e-mails qui ont été envoyés par cette exécution de campagne. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.

Métrique	Kpi-name	Description
Delivery rate (Taux de remise)	`successful-delivery-rate`	Pour l'ensemble des exécutions de campagnes, pourcentage de messages ayant été remis aux destinataires. Cette métrique est calculée en fonction du nombre de messages envoyés par toutes les exécutions de campagnes et remis aux destinataires, divisé par le nombre de messages qui ont été envoyés par l'ensemble de ces exécutions de campagnes .

Métrique	Kpi-name	Description
Delivery rate, grouped by campaign run (Taux de remise, regroupement par exécution de campagne)	`successful-delivery-rate-grouped-by-campaign-activity`	Pour chaque exécution de campagne, pourcentage de messages ayant été remis aux destinataires. Cette métrique est calculée de la façon suivante : le nombre de messages envoyés par une exécution de campagne et remis aux destinataires, divisé par le nombre de messages qui ont été envoyés par cette exécution de campagne. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.

Métrique	Kpi-name	Description
Delivery rate, grouped by date (Taux de remise, regroupement par date)	`successful-delivery-rate-grouped-by-date`	Pour l'ensemble des exécutions de campagnes, pourcentage de messages ayant été remis aux destinataires, pour chaque jour de la plage de dates spécifiée. Cette métrique est calculée de la façon suivante : le nombre de messages envoyés par toutes les exécutions de campagnes et remis aux destinataires, divisé par le nombre de messages qui ont été envoyés par l'ensemble de ces exécutions de campagnes, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Métrique	Kpi-name	Description
Email open rate (Taux d’ouverture d’e-mail)	`email-open-rate`	Pour l'ensemble des exécutions de campagnes, pourcentage d'e-mails ayant été ouverts par les destinataires. Cette métrique est calculée de la façon suivante : le nombre d'e-mails envoyés par toutes les exécutions de campagnes pour un projet et ouverts par les destinataires, divisé par le nombre d'e-mails qui ont été envoyés par l'ensemble de ces exécutions de campagnes et remis aux destinataires.

Métrique	Kpi-name	Description
Email open rate, grouped by campaign run (Taux d'ouverture d'e-mail, regroupement par exécution de campagne)	`email-open-rate-grouped-by-campaign-activity`	Pour chaque exécution de campagne, pourcentage d'e-mails ayant été ouverts par les destinataires. Cette métrique est calculée de la façon suivante : le nombre d'e-mails envoyés par une exécution de campagne et ouverts par les destinataires, divisé par le nombre d'e-mails qui ont été envoyés par l'exécution de campagne et remis aux destinataires. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.
Emails opened, grouped by campaign run (E-mails ouverts, regroupement par exécution de campagne)	`direct-email-opens-grouped-by-campaign-activity`	Pour chaque exécution de campagne, nombre d'e-mails ayant été ouverts par les destinataires. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.

Métrique	Kpi-name	Description
Endpoint deliveries (Remises au point de terminaison)	`unique-deliveries`	Pour l'ensemble des exécutions de campagnes, nombre de points de terminaison uniques auxquels les messages ont été envoyés.
Endpoint deliveries, grouped by campaign run (Remises au point de terminaison, regroupement par exécution de campagne)	`unique-deliveries-grouped-by-campaign-activity`	Pour chaque exécution de campagne, nombre de points de terminaison uniques auxquels les messages ont été envoyés. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.
Endpoint deliveries, grouped by date (Remises au point de terminaison, regroupement par date)	`unique-deliveries-grouped-by-date`	Pour l'ensemble des exécutions de campagnes, nombre de points de terminaison uniques auxquels les messages ont été remis, pour chaque jour de la plage de dates spécifiée. Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.

Métrique	Kpi-name	Description
Links clicked, grouped by campaign run (Liens activés, regroupement par exécution de campagne)	`clicks-grouped-by-campaign-activity`	Pour chaque exécution de campagne, nombre de fois où les destinataires ont cliqué sur les liens dans l'e-mail. Si un seul destinataire a cliqué sur plusieurs liens dans le message, ou s'il a cliqué sur le même lien plusieurs fois, chaque clic est inclus dans le total. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.

Métrique	Kpi-name	Description
Messages delivered, grouped by campaign run (Messages remis, regroupement par exécution de campagne)	`successful-deliveries-grouped-by-campaign-activity`	Pour chaque exécution de campagne, nombre de messages ayant été remis aux destinataires. Cette métrique est calculée de la façon suivante : le nombre de messages envoyés par une exécution de campagne, moins le nombre de messages n'ayant pas été remis aux destinataires de l'exécution du fait d'un message d'erreur définitif. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.
Messages sent, grouped by campagne run (Messages envoyés, regroupement par exécution de campagne)	`attempted-deliveries-grouped-by-campaign-activity`	Pour chaque exécution de campagne, nombre de messages ayant été envoyés. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.

Métrique	Kpi-name	Description
Push open rate (Taux d’ouverture des notifications push)	`push-open-rate`	Pour l'ensemble des exécutions de campagnes, pourcentage de notifications push ayant été ouvertes par les destinataires. Cette métrique est calculée de la façon suivante : le nombre de notifications push envoyées par toutes les exécutions de campagnes et ouvertes par les destinataires, divisé par le nombre de notifications push qui ont été envoyées par l'ensemble de ces exécutions de campagnes et remises aux destinataires.

Métrique	Kpi-name	Description
Push open rate, grouped by campaign run (Taux d'ouverture des notifications push, regroupement par exécution de campagne)	`push-open-rate-grouped-by-campaign-activity`	Pour chaque exécution de campagne, pourcentage de notifications push ayant été ouvertes par les destinataires. Cette métrique est calculée de la façon suivante : le nombre de notifications push envoyées par une exécution de campagne et ouvertes par les destinataires, divisé par le nombre de notifications push qui ont été envoyées par l'exécution de campagne. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.
Total push opened, grouped by campaign run (Nombre total d'ouvertures de notifications push, regroupement par exécution de campagne)	`direct-push-opens-grouped-by-campaign-activity`	Pour chaque exécution de campagne, nombre de notifications push ayant été ouvertes par les destinataires. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (`CampaignActivityId`), une chaîne qui identifie de façon unique une campagne.

Métrique	Kpi-name	Description
Total SMS spend (Total des dépenses en SMS)	`sms-spend`	Pour toutes les campagnes, le montant total, en millième de cents, dépensé pour envoyer des SMS.

Indicateurs d'engagement liés au parcours Amazon Pinpoint

Le tableau suivant répertorie et décrit les métriques d'engagement du parcours standard que vous pouvez interroger pour en savoir plus sur les tendances de tous les e-mails envoyés par un parcours Amazon Pinpoint. Pour interroger les données de ces métriques, utilisez la ressource Métriques d'engagement du parcours de l'API Amazon Pinpoint. La colonne kpi-name de la table indique la valeur à utiliser pour le paramètre `kpi-name` de votre requête.

Métrique	Kpi-name	Description
Clics	`journey-emails-clicked`	Nombre de fois où les destinataires ont cliqué sur des liens dans les messages. Si un seul participant a cliqué sur plusieurs liens dans un message, ou s'il a cliqué sur le même lien plusieurs fois, chaque clic est inclus dans le total.
Clics, regroupés par activité	`emails-clicked-grouped-by-journey-activity`	Pour chaque activité du parcours, le nombre de fois où les participants ont cliqué sur des liens dans les messages. Si un seul participant a cliqué sur plusieurs liens dans un message, ou s'il a cliqué sur le même lien plusieurs fois,

Métrique	Kpi-name	Description
		chaque clic est inclus dans le total. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.
Réclamations	`journey-emails-complained`	Nombre de messages signalés par les participants comme des e-mails indésirables.
Réclamations, regroupées par activité	`emails-complained-grouped-by-journey-activity`	Pour chaque activité du parcours, le nombre de messages signalés par les participants comme des e-mails indésirables. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.

Métrique	Kpi-name	Description
Messages délivrés	`journey-emails-delivered`	Nombre de messages remis aux participants. Cette métrique est calculée en fonction du nombre de messages envoyés moins le nombre de messages qui n'ont pas pu être remis en raison d'une non-remise temporaire ou définitive, ou parce qu'ils ont été rejetés.
Messages remis, regroupés par activité	`emails-delivered-grouped-by-journey-activity`	Pour chaque activité du parcours, le nombre de messages ayant été remis aux participants. Cette métrique est calculée en fonction du nombre de messages envoyés moins le nombre de messages qui n'ont pas pu être remis en raison d'une erreur de remise temporaire ou d'un retour à l'expéditeur, ou parce qu'ils ont été rejetés, pour chaque activité du parcours. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.

Métrique	Kpi-name	Description
Messages d'erreur définitifs	`journey-emails-hardbounced`	Nombre de messages n'ayant pas pu être remis aux participants en raison d'un retour à l'expéditeur. Un message d'erreur définitif est renvoyé si un problème persistant empêche la remise d'un message, par exemple, si l'adresse e-mail d'un participant n'existe pas.
Retours à l'expéditeur, regroupés par date	`emails-hardbounced-grouped-by-journey-activity`	Pour chaque activité du parcours, le nombre de messages n'ayant pas pu être remis aux participants en raison d'un retour à l'expéditeur. Un message d'erreur définitif est renvoyé si un problème persistant empêche la remise d'un message, par exemple, si l'adresse e-mail d'un participant n'existe pas. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.
Messages ouverts	`journey-emails-opened`	Nombre de messages ouverts par les participants.

Métrique	Kpi-name	Description
Messages ouverts, regroupés par activité	`emails-opened-grouped-by-journey-activity`	Pour chaque activité du parcours, le nombre de messages ayant été ouverts par les participants. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.
Rejets	`journey-emails-rejected`	Nombre de messages n'ayant pas été envoyés aux participants parce qu'ils ont été rejetés. Un message est rejeté si Amazon Pinpoint détermine qu'il contient un logiciel malveillant. Amazon Pinpoint n'essaie pas d'envoyer des messages rejetés.

Métrique	Kpi-name	Description
Messages rejetés, regroupés par activité	`emails-rejected-grouped-by-journey-activity`	Pour chaque activité, le nombre de messages n'ayant pas été envoyés aux participants parce qu'ils ont été rejetés. Un message est rejeté si Amazon Pinpoint détermine qu'il contient un logiciel malveillant. Amazon Pinpoint n'essaie pas d'envoyer des messages rejetés. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.
Envois	`journey-emails-sent`	Nombre de messages envoyés.
Messages envoyés, regroupés par activité	`emails-sent-grouped-by-journey-activity`	Pour chaque activité du parcours, le nombre de messages envoyés. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.

Métrique	Kpi-name	Description
Erreurs de remise temporaire	`journey-emails-softbounced`	Nombre de messages n'ayant pas pu être remis aux participants en raison d'une erreur de remise temporaire. Un message d'erreur temporaire est renvoyé si un problème temporaire empêche la distribution d'un message, par exemple, si la boîte de réception d'un participant est pleine ou si le serveur de réception est temporairement indisponible.

Métrique	Kpi-name	Description
Erreurs de remise temporaire, regroupées par activité	`emails-softbounced-grouped-by-journey-activity`	Pour chaque activité du parcours, le nombre de messages n'ayant pas pu être remis aux participants en raison d'une erreur de remise temporaire. Un message d'erreur temporaire est renvoyé si un problème temporaire empêche la distribution d'un message, par exemple, si la boîte de réception d'un participant est pleine ou si le serveur de réception est temporairement indisponible. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.
Désabonnement	`journey-emails-unsubscribed`	Nombre de fois où les participants ont cliqué sur des liens de désinscription dans les messages. Si un participant a cliqué plusieurs fois sur le même lien de désabonnement, chaque clic est inclus dans le total.

Métrique	Kpi-name	Description
Désabonnements, regroupés par activité	`emails-unsubscribed-grouped-by-journey-activity`	Pour chaque activité du parcours, le nombre de fois où les participants ont cliqué sur des liens de désabonnement dans les messages. Si un participant a cliqué plusieurs fois sur le même lien de désabonnement, chaque clic est inclus dans le total. Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (`JourneyActivityId`), une chaîne qui identifie de façon unique une activité.

Indicateurs d'exécution du parcours Amazon Pinpoint

Le tableau suivant répertorie et décrit les métriques d'exécution standard que vous pouvez interroger pour évaluer le statut des participants dans un parcours Amazon Pinpoint. Pour interroger les données de ces métriques, utilisez la ressource Métriques d'exécution du parcours de l'API Amazon Pinpoint. La colonne Field (Champ) du tableau identifie le nom du champ qui s'affiche dans les résultats de la requête pour chaque métrique.

Métrique	Champ	Description
Participants actifs	`ENDPOINT_ACTIVE`	Nombre de participants poursuivant activement les activités du parcours. Cette métrique est calculée de la façon suivante : nombre de participants qui ont commencé le parcours, moins nombre

Métrique	Champ	Description
		de participants ayant quitté le parcours et le nombre de participants qui ont été exclus du parcours.
Annulations des participants	`CANCELLED`	Nombre de participants n'ayant pas terminé le parcours en raison de son annulation.
Départs des participants	`ENDPOINT_LEFT`	Nombre de participants ayant quitté le parcours.
Admissions des participants	`ENDPOINT_ENTERED`	Nombre de participants ayant commencé le parcours.
Exceptions des participants, limites de réadmission	`REENTRY_CAP_EXCEEDED`	Nombre de participants n'ayant pas terminé le parcours parce qu'ils auraient dépassé le nombre maximum de fois qu'un seul participant peut accéder au parcours.

Métrique	Champ	Description
Exceptions des participants, rejets	`ACTIVE_ENDPOINT_REJECTED`	Nombre de participants qui ne peuvent pas commencer le parcours parce qu'ils sont déjà des participants actifs au parcours. Un participant est rejeté s'il commence un parcours et que vous mettez ensuite à jour la définition de son point de terminaison d'une façon qui affecte son inclusion dans un segment (en fonction des critères du segment) ou un parcours (en fonction des conditions de l'activité).

Mesures d'exécution des activités liées au parcours Amazon Pinpoint

Le tableau suivant répertorie et décrit les métriques d'exécution standard que vous pouvez interroger pour évaluer le statut des participants dans chaque type d'activité individuelle d'un parcours Amazon Pinpoint. Pour interroger les données de ces métriques, utilisez la ressource Métriques d'exécution des activités du parcours de l'API Amazon Pinpoint. La colonne Metrics (Métriques) du tableau répertorie les champs qui s'affichent dans les résultats de la requête pour chaque type d'activité. Elle fournit également une brève description de chaque champ.

Type d'activité	Métriques
Yes/No scinder (`CONDITIONAL_SPLIT`)	Les métriques sont les suivantes : • `Branch_FALSE` : nombre de participants qui ne répondaient pas aux conditions de l'activité et qui se sont dirigés vers l'activité sur le chemin « Non ».

Type d'activité	Métriques
	• `Branch_TRUE` : nombre de participants qui répondaient aux conditions de l'activité et qui se sont dirigés vers l'activité sur le chemin « Oui ». Des métriques supplémentaires sont disponibles pour chaque chemin de l'activité. Pour de plus amples informations sur ces métriques, veuillez consulter la ligne de ce tableau correspondant à ce type d'activité.
Holdout (Exclusion) (`HOLDOUT`)	Les métriques sont les suivantes : • `HOLDOUT` : nombre de participants ayant été exclus du parcours dans le cadre du pourcentage d'exclusion pour l'activité. • `PASSED` : nombre de participants qui sont passés à l'activité suivante du parcours.

Type d'activité	Métriques
Email (E-mail) (`MESSAGE`)	Les métriques sont les suivantes : • `DAILY_CAP_EXCEEDED` : nombre de messages n'ayant pas été envoyés parce qu'ils auraient dépassé le nombre maximum de messages qu'un seul participa nt peut recevoir au cours d'une période de 24 heures. • `FAILURE_PERMANENT` : nombre de messages n'ayant pas été envoyés en raison d'un problème permanent. • `QUIET_TIME` : nombre de messages n'ayant pas été envoyés, car ils seraient arrivés pendant le temps de pause du participant. • `SERVICE_FAILURE` : nombre de messages n'ayant pas été envoyés en raison d'un problème avec Amazon Pinpoint. • `SUCCESS` : nombre de messages ayant été remis avec succès aux participants. • `THROTTLED` : nombre de messages n'ayant pas été envoyés, car les quotas d'envoi pour votre compte Amazon Pinpoint auraient été dépassés. • `TRANSIENT_FAILURE` : nombre de messages n'ayant pas été envoyés en raison d'un problème temporaire. • `UNKNOWN` : nombre de messages n'ayant pas été envoyés en raison d'un problème inconnu.

Type d'activité	Métriques
Multivariate split (Répartition multivariée) (`MULTI_CONDITIONAL_SPLIT`)	Pour chaque chemin de l'activité, le nombre de participants qui se sont dirigés vers l'activité sur le chemin. Les résultats des requêtes pour cette métrique sont regroupés par chemin, `Branch_#` où `#` est l'identifiant numérique d'un chemin, par exemple, `Branch_1` pour le premier chemin de l'activité. Des métriques supplémentaires sont disponibles pour chaque chemin de l'activité. Pour de plus amples informations sur ces métriques, veuillez consulter la ligne de ce tableau correspondant à ce type d'activité.
Random split (Répartition aléatoire) (`RANDOM_SPLIT`)	Pour chaque chemin de l'activité, le nombre de participants qui se sont dirigés vers l'activité sur le chemin. Les résultats des requêtes pour cette métrique sont regroupés par chemin, `Branch_#` où `#` est l'identifiant numérique d'un chemin, par exemple, `Branch_1` pour le premier chemin de l'activité. Des métriques supplémentaires sont disponibles pour chaque chemin de l'activité. Pour de plus amples informations sur ces métriques, veuillez consulter la ligne de ce tableau correspondant à ce type d'activité.

Type d'activité	Métriques
Wait (Attente) (`WAIT`)	Les métriques sont les suivantes : • `WAIT_FINISHED` : nombre de participants ayant fini d'attendre le temps spécifié. • `WAIT_SKIPPED` : nombre de participants n'ayant pas attendu le temps spécifié, généralement car ils ont commencé l'activité ou le parcours après l'heure de fin prévue de l'activité. • `WAIT_STARTED` : nombre de participants ayant commencé à attendre et n'ayant pas passé ou fini d'attendre le temps spécifié.

Type d'activité	Métriques
Centre de contact (`CONTACT_CENTER`)	Les métriques sont les suivantes : • `CALL_QUEUED` — Le nombre de participants qui ont été appelés et mis en file d'attente pour Connect Customer. Inclut les tentatives de rappel. • `CONTINUE_WAITING` : nombre de participants qui continuent d'attendre que des tentatives d'appel soient effectuées. • `DIAL_FAILURE` : nombre de participants dont les tentatives d'appel ont échoué. • `DROPPED` : nombre de participants qui ne remplissent plus les conditions définies lors des activités de parcours précédentes au moment de l'envoi. • `TIMEOUT`— Le nombre de participants qui n'ont reçu aucun code de disposition Connect Customer après plusieurs tentatives d'appel. • `WAIT_FINISHED` : nombre de participants ayant fini d'attendre le temps spécifié. • `WAIT_FOR_QUIET_HOURS` : nombre de participants qui attendent la fin du temps de pause pour pouvoir être transférés au canal. • `WAIT_STARTED` : nombre de participants ayant commencé à attendre et n'ayant pas passé ou fini d'attendre le temps spécifié.

Statistiques relatives au parcours et à l'exécution des campagnes Amazon Pinpoint

Vous pouvez interroger les mesures d'exécution standard pour évaluer le statut des participants à chaque type d'activité individuelle pour un parcours ou une campagne Amazon Pinpoint. Pour

interroger les données de ces métriques, utilisez la ressource Métriques d'exécution des activités du parcours ou la ressource Métriques de campagnes de l'API Amazon Pinpoint. Le tableau suivant répertorie les champs qui s'affichent dans les résultats de la requête pour chaque type d'activité.

Nom de la métrique	S'applique aux parcours, aux campagnes ou aux deux	Description
ENDPOINT_PRODUCED	les deux	Le nombre de points de terminaison initialement produits à partir du segment ou de l'événement avant tout filtrage.
ENDPOINTS_FROM_USER	les deux	Si le client possède un segment contenant uniquement un ID utilisateur, tous les points de terminais on de ces utilisateurs seront ajoutés. Cette métrique mesure le nombre de points de terminaison ajoutés de cette manière.
ENDPOINT_OPT_OUT	les deux	Le point de terminaison a été désactivé et n'a pas participé à la campagne ou au parcours.
ENDPOINT_INACTIVE	les deux	Le point de terminaison est désactivé et n'a pas participé à la campagne ou au parcours.
FILTERED_OUT_BY_SEGMENT	les deux	Le point de terminaison ne correspondait pas aux filtres de segment et n'a pas participé à la campagne ou au parcours.

Nom de la métrique	S'applique aux parcours, aux campagnes ou aux deux	Description
ENDPOINT_MISSING_ADDRESS	les deux	Il manquait une adresse au point de terminaison et il n'a pas participé à la campagne ou au parcours.
ENDPOINT_MISSING_CHANNEL	les deux	Il manquait un canal au point de terminaison et il n'a pas participé à la campagne ou au parcours.
ENDPOINT_MISSING_TIMEZONE	les deux	Il manquait une valeur de fuseau horaire au point de terminaison et il a été exclu du filtrage. Cela se produit uniquement lorsqu'une valeur de fuseau horaire est requise.
ENDPOINT_TIMEZONE_MISMATCH	les deux	Le point de terminaison se trouvait dans un fuseau horaire qui n'était pas inclus dans l'exécution à ce moment-là.
ENDPOINT_CHANNEL_MISMATCH	Campagnes	Aucun message n'est configuré pour le type de canal de ce point de terminaison.
DUPLICATE_ENDPOINT	les deux	Des points de terminaison dupliqués ont été trouvés et dédupliqués.

Nom de la métrique	S'applique aux parcours, aux campagnes ou aux deux	Description
DUPLICATE_USER	les deux	Des utilisateurs dupliqués ont été trouvés et dédupliqués à partir d'un segment contenant uniquement des ID d'utilisateur. S'ils ont le même ID utilisateur, une métrique de 1 sera émise.
PAUSED	Parcours	Supprimé de l'exécution car le parcours a été suspendu.
ENDED	Parcours	Supprimé de l'exécution car le parcours a été supprimé.
TREATMENT_HOLDOUT	Campagnes	Cela est émis dans les A/B campagnes, pour les paramètres dont les cohortes ne correspondent pas au traitement actuel. Par exemple, dans un 50/50 A/B split, 50 % des points de terminaison émettront cette métrique pour chaque traitement
ENDPOINT_ESTIMATED_TIMEZONE	Parcours	L'estimation du fuseau horaire a permis d'estimer un fuseau horaire pour le point de terminaison.

Interrogez les données d'analyse Amazon Pinpoint pour les campagnes

Outre les pages d'analyse de la console Amazon Pinpoint, vous pouvez utiliser Amazon Pinpoint APIs Analytics pour interroger les données d'analyse relatives à un sous-ensemble de mesures standard qui fournissent des informations sur les tendances en matière de diffusion et d'engagement pour les campagnes.

Chacune de ces métriques est une valeur mesurable, également appelée indicateur de performance clé (KPI), qui peut vous aider à surveiller et à évaluer les performances d'une ou plusieurs campagnes. Par exemple, vous pouvez utiliser une métrique pour connaître le nombre de points de terminaison auxquels un message de campagne a été envoyé ou le nombre de ces messages qui ont été remis aux points de terminaison prévus.

Amazon Pinpoint collecte et regroupe automatiquement ces données pour toutes vos campagnes. Il stocke les données pendant 90 jours. Si vous avez intégré une application mobile à Amazon Pinpoint à l'aide d'un SDK AWS mobile, Amazon Pinpoint étend cette prise en charge pour inclure des statistiques supplémentaires, telles que le pourcentage de notifications push ouvertes par les destinataires. Pour plus d'informations sur l'intégration d'une application mobile, reportez-vous à la section Intégrez Amazon Pinpoint à votre application.

Si vous utilisez Amazon Pinpoint Analytics APIs pour interroger des données, vous pouvez choisir différentes options qui définissent l'étendue, les données, le regroupement et les filtres de votre requête. Pour ce faire, utilisez les paramètres qui spécifient le projet, la campagne et la mesure que vous souhaitez interroger, en plus des filtres basés sur la date que vous souhaitez appliquer.

Cette rubrique explique et fournit des exemples de sélection de ces options et d'interrogation des données pour une ou plusieurs campagnes.

Conditions préalables

Avant d'interroger les données d'analyse d'une ou de plusieurs campagnes, il est utile de recueillir les informations suivantes, que vous utilisez pour définir votre requête :

- ID de projet : identifiant unique du projet associé à la ou aux campagnes. Dans l'API Amazon Pinpoint, cette valeur est stockée dans la propriété `application-id`. Sur la console Amazon Pinpoint, cette valeur s'affiche sous la forme de l'ID de projet sur la page Tous les projets.

- ID de campagne : identifiant unique de la campagne, si vous souhaitez interroger les données d'une seule campagne. Dans l'API Amazon Pinpoint, cette valeur est stockée dans la propriété `campaign-id`. Cette valeur n'est pas affichée sur la console.
- Plage de dates : le cas échéant, premières et dernières date et heure de la plage de dates auxquelles interroger les données. Notez que les plages de dates sont inclusives et doivent être limitées à 31 jours calendaires au plus. En outre, elles doivent commencer dans moins de 90 jours à compter du jour actuel. Si vous ne spécifiez pas de plage de dates, Amazon Pinpoint interroge automatiquement les données des 31 jours civils précédents.
- Type de métrique : type de métrique à interroger. Il existe deux types : les métriques d'application et les métriques de campagne. Une métrique d'application fournit les données de toutes les campagnes associées à un projet, également appelée application. Une métrique de campagne fournit les données pour une seule campagne.
- Métrique : nom de la métrique à interroger, plus précisément la valeur `kpi-name` de la métrique. Pour obtenir la liste complète des métriques prises en charge et la valeur `kpi-name` de chacune d'elles, reportez-vous à la section Mesures standard pour les projets, les campagnes et les parcours.

Permet également de déterminer si vous souhaitez regrouper les données selon un champ pertinent. Si c'est le cas, vous pouvez simplifier votre analyse et vos rapports en choisissant une métrique conçue pour regrouper automatiquement les données. Par exemple, Amazon Pinpoint fournit plusieurs métriques standard qui indiquent le pourcentage de messages remis aux destinataires d'une campagne. L'une de ces métriques regroupe automatiquement les données par date (`successful-delivery-rate-grouped-by-date`). Une autre métrique regroupe automatiquement les données par exécution de campagne (`successful-delivery-rate-grouped-by-campaign-activity`). Une troisième métrique renvoie simplement une valeur unique : le pourcentage de messages remis aux destinataires par toutes les exécutions de la campagne (`successful-delivery-rate`).

Si vous ne trouvez pas de métrique standard qui regroupe les données de la manière souhaitée, vous pouvez développer une série de requêtes qui renvoient les données souhaitées. Vous pouvez ensuite décomposer ou combiner manuellement les résultats de la requête en groupes personnalisés que vous concevez.

Enfin, il est important de vérifier que vous êtes autorisé à accéder aux données que vous souhaitez interroger. Pour de plus amples informations, veuillez consulter Politiques IAM pour l'interrogation de données analytiques Amazon Pinpoint.

Interrogez les données Amazon Pinpoint pour une campagne

Pour interroger les données d'une campagne, utilisez l'API Métriques de campagne et spécifiez les valeurs des paramètres requis suivants :

- application-id : ID du projet, qui est l'identifiant unique du projet associé à la campagne. Dans Amazon Pinpoint, les termes projet et application ont la même signification.
- campaign-id : identifiant unique de la campagne.
- kpi-name : nom de la métrique à interroger. Cette valeur décrit la métrique associée et se compose de deux termes ou plus, composés de caractères alphanumériques minuscules, séparés par un trait d'union. Pour obtenir la liste complète des métriques prises en charge et la valeur `kpi-name` de chacune d'elles, reportez-vous à la section Mesures standard pour les projets, les campagnes et les parcours.

Vous pouvez également appliquer un filtre qui interroge les données pour une plage de dates spécifique. Si vous ne spécifiez pas de plage de dates, Amazon Pinpoint renvoie les données des 31 jours civils précédents. Pour filtrer les données selon différentes dates, utilisez les paramètres de plage de dates pris en charge pour spécifier les premières et dernières date et heure de la plage de dates. Les valeurs doivent être au format ISO 8601 étendu et utiliser le temps universel coordonné (UTC) : par exemple, `2019-07-19T20:00:00Z` pour 20h00 UTC le 19 juillet 2019. Notez que les plages de dates sont inclusives et doivent être limitées à 31 jours calendaires au plus. En outre, les premières date et heure doivent être inférieures à 90 jours à compter du jour actuel.

Les exemples suivants montrent comment interroger des données d'analyse pour une campagne à l'aide de l'API REST Amazon Pinpoint, du AWS CLI, et du. AWS SDK pour Java Vous pouvez utiliser n'importe quel AWS SDK compatible pour interroger les données d'analyse d'une campagne. Les AWS CLI exemples sont formatés pour Microsoft Windows. Pour Unix, Linux et macOS, remplacez le caractère caret (^) de continuation de ligne par une barre oblique inverse (\).

REST API

Pour interroger les données analytiques d'une campagne à l'aide de l'API REST Amazon Pinpoint, envoyez une demande HTTP(S) GET à l'URI des métriques de campagne. Dans l'URI, spécifiez les valeurs appropriées pour les paramètres de chemin requis :

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

Où :

- *endpoint*est le point de terminaison Amazon Pinpoint pour la AWS région qui héberge le projet associé à la campagne.
- *application-id* est l'identifiant unique du projet associé à la campagne.
- *campaign-id* est l'identifiant unique de la campagne.
- *kpi-name*est la `kpi-name` valeur de la métrique à interroger.

Tous les paramètres doivent être codés par URL.

Pour appliquer un filtre qui interroge les données pour une plage de dates spécifique, ajoutez les paramètres `start-time` et `end-time` et leurs valeurs à l'URI. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Utilisez une esperluette (&) pour séparer les paramètres.

Par exemple, la demande suivante extrait le nombre de points de terminaison uniques auxquels les messages ont été remis, par toutes les exécutions d'une campagne, du 19 juillet 2019 au 26 juillet 2019 :

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456789012345example/
campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Où :

- `pinpoint.us-east-1.amazonaws.com` est le point de terminaison Amazon Pinpoint de la région AWS qui héberge le projet.
- `12345678901234567890123456789012345example` est l'identifiant unique du projet associé à la campagne.
- `80b8efd84042ff8d9c96ce2f8example` est l'identifiant unique de la campagne.
- `unique-deliveries` est la valeur `kpi-name` de la métrique de campagne endpoint deliveries (remises de points de terminaison). Cette métrique indique le nombre de points de terminaison uniques auxquels les messages ont été remis, pour toutes les exécutions d'une campagne.
- `2019-07-19T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.

- `2019-07-26T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

AWS CLI

Pour interroger les données d'analyse d'une campagne à l'aide de AWS CLI, utilisez la get-campaign-date-range-kpi commande et spécifiez les valeurs appropriées pour les paramètres requis :

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id application-id ^
    --campaign-id campaign-id ^
    --kpi-name kpi-name
```

Où :

- *`application-id`* est l'identifiant unique du projet associé à la campagne.
- *`campaign-id`* est l'identifiant unique de la campagne.
- *`kpi-name`*est la `kpi-name` valeur de la métrique à interroger.

Pour appliquer un filtre qui interroge les données pour une plage de dates spécifique, ajoutez les paramètres `start-time` et`end-time` et leurs valeurs à votre requête. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Par exemple, la demande suivante extrait le nombre de points de terminaison uniques auxquels les messages ont été remis, par toutes les exécutions d'une campagne, du 19 juillet 2019 au 26 juillet 2019 :

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --campaign-id 80b8efd84042ff8d9c96ce2f8example ^
    --kpi-name unique-deliveries ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Où :

- `12345678901234567890123456example` est l'identifiant unique du projet associé à la campagne.

- `80b8efd84042ff8d9c96ce2f8example` est l'identifiant unique de la campagne.
- `unique-deliveries` est la valeur `kpi-name` de la métrique de campagne endpoint deliveries (remises de points de terminaison). Cette métrique indique le nombre de points de terminaison uniques auxquels les messages ont été remis, pour toutes les exécutions d'une campagne.
- `2019-07-19T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-07-26T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

SDK for Java

Pour interroger les données d'analyse d'une campagne à l'aide de AWS SDK pour Java, utilisez la GetCampaignDateRangeKpiRequest méthode de l'API Campaign Metrics. Spécifiez les valeurs appropriées pour les paramètres requis :

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withCampaignId("campaignId")
        .withKpiName("kpiName")
```

Où :

- *`applicationId`* est l'identifiant unique du projet associé à la campagne.
- *`campaignId`* est l'identifiant unique de la campagne.
- *`kpiName`*est la `kpi-name` valeur de la métrique à interroger.

Pour appliquer un filtre qui interroge les données pour une plage de dates spécifique, incluez les paramètres `startTime` et`endTime` et leurs valeurs dans votre requête. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Par exemple, la demande suivante extrait le nombre de points de terminaison uniques auxquels les messages ont été remis, par toutes les exécutions d'une campagne, du 19 juillet 2019 au 26 juillet 2019 :

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withCampaignId("80b8efd84042ff8d9c96ce2f8example")
        .withKpiName("unique-deliveries")
```

```
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Où :

- `12345678901234567890123456example` est l'identifiant unique du projet associé à la campagne.
- `80b8efd84042ff8d9c96ce2f8example` est l'identifiant unique de la campagne.
- `unique-deliveries` est la valeur `kpi-name` de la métrique de campagne endpoint deliveries (remises de points de terminaison). Cette métrique indique le nombre de points de terminaison uniques auxquels les messages ont été remis, pour toutes les exécutions d'une campagne.
- `2019-07-19T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-07-26T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

Une fois que vous avez envoyé votre requête, Amazon Pinpoint renvoie les résultats de la requête dans une réponse JSON. La structure des résultats varie en fonction de la métrique que vous avez interrogée. Certaines mesures ne renvoient qu'une seule valeur. Par exemple, la métrique d'application remises au point de terminaison (`unique-deliveries`), utilisée dans les exemples précédents, renvoie une seule valeur : le nombre de points de terminaison uniques auxquels les messages ont été remis, pour toutes les exécutions d'une campagne. Dans ce cas, la réponse JSON est la suivante :

```
{
    "CampaignDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "CampaignId":"80b8efd84042ff8d9c96ce2f8example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
```

```
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```

D'autres métriques renvoient plusieurs valeurs et regroupent les valeurs selon un champ pertinent. Si une mesure renvoie plusieurs valeurs, la réponse JSON inclut un champ qui indique quel champ a été utilisé pour regrouper les données.

Pour en savoir plus sur la structure des résultats de requête, reportez-vous à la section Utiliser les résultats des requêtes JSON.

Interrogez les données Amazon Pinpoint pour plusieurs campagnes

Il existe deux façons d'interroger les données pour plusieurs campagnes. La meilleure façon dépend de votre souhait d'interroger les données pour toutes les campagnes associées au même projet. Dans ce cas, cela dépend également de votre souhait d'interroger les données pour l'ensemble ou un sous-ensemble de ces campagnes.

Pour interroger les données des campagnes associées à différents projets ou pour un sous-ensemble de campagnes associées au même projet, la meilleure approche consiste à créer et à exécuter une série de requêtes individuelles, une pour chaque campagne pour laquelle vous souhaitez interroger les données. La section précédente explique comment interroger les données pour une seule campagne.

Pour interroger les données de toutes les campagnes associées au même projet, vous pouvez utiliser l'API métriques d'application. Spécifiez des valeurs pour les paramètres obligatoires suivants :

- application-id : ID du projet, qui est l'identifiant unique du projet. Dans Amazon Pinpoint, les termes projet et application ont la même signification.
- kpi-name : nom de la métrique à interroger. Cette valeur décrit la métrique associée et se compose de deux termes ou plus, composés de caractères alphanumériques minuscules, séparés par un trait d'union. Pour obtenir la liste complète des métriques prises en charge et la valeur `kpi-name` de chacune d'elles, reportez-vous à la section Mesures standard pour les projets, les campagnes et les parcours.

Vous pouvez également filtrer les données par plage de dates. Si vous ne spécifiez pas de plage de dates, Amazon Pinpoint renvoie les données des 31 jours civils précédents. Pour filtrer les données selon différentes dates, utilisez les paramètres de plage de dates pris en charge pour spécifier les premières et dernières date et heure de la plage de dates. Les valeurs doivent être au format ISO 8601 étendu et utiliser le temps universel coordonné (UTC) : par exemple, `2019-07-19T20:00:00Z` pour 20h00 UTC le 19 juillet 2019. Notez que les plages de dates sont inclusives et doivent être limitées à 31 jours calendaires au plus. En outre, les premières date et heure doivent être inférieures à 90 jours à compter du jour actuel.

Les exemples suivants montrent comment interroger des données d'analyse pour une campagne à l'aide de l'API REST Amazon Pinpoint, du AWS CLI, et du. AWS SDK pour Java Vous pouvez utiliser n'importe quel AWS SDK compatible pour interroger les données d'analyse d'une campagne. Les AWS CLI exemples sont formatés pour Microsoft Windows. Pour Unix, Linux et macOS, remplacez le caractère caret (^) de continuation de ligne par une barre oblique inverse (\).

REST API

Pour interroger les données analytiques de plusieurs campagnes à l'aide de l'API REST Amazon Pinpoint, envoyez une demande HTTP(S) GET à l'URI des métriques d'application. Dans l'URI, spécifiez les valeurs appropriées pour les paramètres de chemin requis :

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Où :

- *`endpoint`*est le point de terminaison Amazon Pinpoint pour la AWS région qui héberge le projet associé aux campagnes.
- *`application-id`* est l'identifiant unique du projet associé aux campagnes.
- *`kpi-name`*est la `kpi-name` valeur de la métrique à interroger.

Tous les paramètres doivent être codés par URL.

Pour appliquer un filtre qui extrait les données d'une plage de dates spécifique, ajoutez les paramètres de requête `start-time` et `end-time` et leurs valeurs à l'URI. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Utilisez une esperluette (&) pour séparer les paramètres.

Par exemple, la demande suivante extrait le nombre de points de terminaison uniques auxquels les messages ont été remis, par chacune des campagnes d'un projet, du 19 juillet 2019 au 26 juillet 2019 :

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/unique-deliveries-grouped-by-campaign?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Où :

- `pinpoint.us-east-1.amazonaws.com` est le point de terminaison Amazon Pinpoint de la région AWS qui héberge le projet.
- `12345678901234567890123456example` est l'identifiant unique du projet associé aux campagnes.
- `unique-deliveries-grouped-by-campaign` est la valeur `kpi-name` de la métrique de campagne endpoint deliveries (remises de points de terminaison). Cette métrique indique le nombre de points de terminaison uniques auxquels les messages ont été remis, pour chaque campagne.
- `2019-07-19T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-07-26T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

AWS CLI

Pour interroger les données d'analyse de plusieurs campagnes à l'aide de AWS CLI, utilisez la get-application-date-range-kpi commande et spécifiez les valeurs appropriées pour les paramètres requis :

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Où :

- *`application-id`* est l'identifiant unique du projet associé aux campagnes.
- *`kpi-name`*est la `kpi-name` valeur de la métrique à interroger.

Pour appliquer un filtre qui extrait les données d'une plage de dates spécifique, incluez les paramètres `start-time` et `end-time` et leurs valeurs dans votre requête. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Par exemple, la demande suivante extrait le nombre de points de terminaison uniques auxquels les messages ont été remis, par chacune des campagnes d'un projet, du 19 juillet 2019 au 26 juillet 2019 :

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name unique-deliveries-grouped-by-campaign ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Où :

- `12345678901234567890123456example` est l'identifiant unique du projet associé à la campagne.
- `unique-deliveries-grouped-by-campaign` est la valeur `kpi-name` de la métrique de campagne endpoint deliveries (remises de points de terminaison). Cette métrique indique le nombre de points de terminaison uniques auxquels les messages ont été remis, pour chaque campagne.
- `2019-07-19T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-07-26T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

SDK for Java

Pour interroger les données d'analyse de plusieurs campagnes à l'aide de AWS SDK pour Java, utilisez la GetApplicationDateRangeKpiRequest méthode de l'API Application Metrics. Spécifiez les valeurs appropriées pour les paramètres requis :

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Où :

- *applicationId* est l'identifiant unique du projet associé aux campagnes.
- *kpiName*est la kpi-name valeur de la métrique à interroger.

Pour appliquer un filtre qui extrait les données d'une plage de dates spécifique, incluez les paramètres startTime et endTime et leurs valeurs dans votre requête. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Par exemple, la demande suivante extrait le nombre de points de terminaison uniques auxquels les messages ont été remis, par chacune des campagnes d'un projet, du 19 juillet 2019 au 26 juillet 2019 :

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("unique-deliveries-grouped-by-campaign")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Où :

- 12345678901234567890123456example est l'identifiant unique du projet associé aux campagnes.
- unique-deliveries-grouped-by-campaign est la valeur kpi-name de la métrique de campagne endpoint deliveries (remises de points de terminaison). Cette métrique indique le nombre de points de terminaison uniques auxquels les messages ont été remis, pour chaque campagne.
- 2019-07-19T00:00:00Z est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- 2019-07-26T23:59:59Z est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

Une fois que vous avez envoyé votre requête, Amazon Pinpoint renvoie les résultats de la requête dans une réponse JSON. La structure des résultats varie en fonction de la métrique que vous avez interrogée. Certaines mesures ne renvoient qu'une seule valeur. D'autres métriques renvoient plusieurs valeurs, regroupées selon un champ pertinent. Si une mesure renvoie plusieurs valeurs, la réponse JSON inclut un champ qui indique quel champ a été utilisé pour regrouper les données.

Par exemple, la métrique d'application des remises de point de terminaison, regroupées par campagne (unique-deliveries-grouped-by-campaign), utilisée dans les exemples

précédents, renvoie plusieurs valeurs : le nombre de points de terminaison uniques auxquels les messages ont été remis, pour chaque campagne associée à un projet. Dans ce cas, la réponse JSON est la suivante :


```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
```

```
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```

Dans ce cas, le champ `GroupedBys` indique que les valeurs sont regroupées par ID de campagne (`CampaignId`).

Pour en savoir plus sur la structure des résultats de requête, reportez-vous à la section Utiliser les résultats des requêtes JSON.

Interrogez les données d'analyse Amazon Pinpoint pour les messages transactionnels

Outre les pages d'analyse de la console Amazon Pinpoint, vous pouvez utiliser Amazon Pinpoint APIs Analytics pour interroger les données d'analyse relatives à un sous-ensemble de mesures standard qui fournissent un aperçu des tendances en matière de diffusion et d'engagement des messages transactionnels envoyés dans le cadre d'un projet.

Chacune de ces métriques est une valeur mesurable, également appelée indicateur de performance clé (KPI), qui peut vous aider à surveiller et à évaluer les performances des messages transactionnels. Par exemple, vous pouvez utiliser une métrique pour savoir combien d'e-mails ou de SMS transactionnels vous avez envoyés, ou combien de ces messages ont été remis aux

destinataires. Amazon Pinpoint collecte et regroupe automatiquement ces données pour tous les e-mails et SMS transactionnels que vous envoyez dans le cadre d'un projet. Il stocke les données pendant 90 jours.

Si vous utilisez Amazon Pinpoint Analytics APIs pour interroger des données, vous pouvez choisir différentes options qui définissent l'étendue, les données, le regroupement et les filtres de votre requête. Pour ce faire, utilisez les paramètres qui spécifient le projet et la métrique que vous souhaitez interroger, en plus des filtres basés sur la date que vous souhaitez appliquer.

Cette rubrique explique et fournit des exemples de sélection de ces options et d'interrogation des données de messagerie transactionnelle pour un projet.

Conditions préalables

Avant d'interroger des données d'analyse pour les messages transactionnels, il est utile de recueillir les informations suivantes, que vous utilisez pour définir votre requête :

- ID du projet : identifiant unique du projet à partir duquel les messages ont été envoyés. Dans l'API Amazon Pinpoint, cette valeur est stockée dans la propriété `application-id`. Sur la console Amazon Pinpoint, cette valeur s'affiche sous la forme de l'ID de projet sur la page Tous les projets.
- Plage de dates : le cas échéant, premières et dernières date et heure de la plage de dates auxquelles interroger les données. Notez que les plages de dates sont inclusives et doivent être limitées à 31 jours calendaires au plus. En outre, elles doivent commencer dans moins de 90 jours à compter du jour actuel. Si vous ne spécifiez pas de plage de dates, Amazon Pinpoint interroge automatiquement les données des 31 jours civils précédents.
- Métrique : nom de la métrique à interroger, plus précisément la valeur `kpi-name` de la métrique. Pour obtenir la liste complète des métriques prises en charge et la valeur `kpi-name` de chacune d'elles, reportez-vous à la section Mesures standard pour les projets, les campagnes et les parcours.

Permet également de déterminer si vous souhaitez regrouper les données selon un champ pertinent. Si c'est le cas, vous pouvez simplifier votre analyse et vos rapports en choisissant une métrique conçue pour regrouper automatiquement les données. Par exemple, Amazon Pinpoint fournit plusieurs métriques standard qui indiquent le nombre de messages SMS transactionnels qui ont été remis aux destinataires. L'une de ces métriques regroupe automatiquement les données par date (`txn-sms-delivered-grouped-by-date`). Une autre métrique regroupe automatiquement les données par pays ou région (`txn-sms-delivered-grouped-by-country`). Une troisième

métrique renvoie simplement une valeur unique : le nombre de messages qui ont été remis aux destinataires (`txn-sms-delivered`). Si vous ne trouvez pas de métrique standard qui regroupe les données de la manière souhaitée, vous pouvez développer une série de requêtes qui renvoient les données souhaitées. Vous pouvez ensuite décomposer ou combiner manuellement les résultats de la requête en groupes personnalisés que vous concevez.

Enfin, il est important de vérifier que vous êtes autorisé à accéder aux données que vous souhaitez interroger. Pour de plus amples informations, veuillez consulter Politiques IAM pour l'interrogation de données analytiques Amazon Pinpoint.

Interrogez les données Amazon Pinpoint pour les e-mails transactionnels

Pour interroger les données des messages électroniques transactionnels envoyés pour un projet, utilisez l'API Application Metrics et spécifiez les valeurs pour les paramètres requis suivants :

- application-id : ID du projet, qui est l'identifiant unique du projet. Dans Amazon Pinpoint, les termes projet et application ont la même signification.
- kpi-name : nom de la métrique à interroger. Cette valeur décrit la métrique associée et se compose de deux termes ou plus, composés de caractères alphanumériques minuscules, séparés par un trait d'union. Pour obtenir la liste complète des métriques prises en charge et la valeur `kpi-name` de chacune d'elles, reportez-vous à la section Mesures standard pour les projets, les campagnes et les parcours.

Vous pouvez également appliquer un filtre qui interroge les données pour une plage de dates spécifique. Si vous ne spécifiez pas de plage de dates, Amazon Pinpoint renvoie les données des 31 jours civils précédents. Pour filtrer les données selon différentes dates, utilisez les paramètres de plage de dates pris en charge pour spécifier les premières et dernières date et heure de la plage de dates. Les valeurs doivent être au format ISO 8601 étendu et utiliser le temps universel coordonné (UTC). Par exemple, `2019-09-06T20:00:00Z` pour le 6 septembre 2019 à 20h00 UTC. Notez que les plages de dates sont inclusives et doivent être limitées à 31 jours calendaires au plus. En outre, les premières date et heure doivent être inférieures à 90 jours à compter du jour actuel.

Les exemples suivants montrent comment interroger des données d'analyse pour des e-mails transactionnels à l'aide de l'API REST Amazon Pinpoint, AWS CLI du, et du. AWS SDK pour Java Vous pouvez utiliser n'importe quel kit SDK AWS pris en charge pour interroger les données d'analyse pour les messages transactionnels. Les AWS CLI exemples sont formatés pour Microsoft Windows. Pour Unix, Linux et macOS, remplacez le caractère caret (^) de continuation de ligne par une barre oblique inverse (\).

REST API

Pour interroger les données analytiques des e-mails transactionnels à l'aide de l'API REST Amazon Pinpoint, envoyez une demande HTTP(S) GET à l'URI des Métriques d'application. Dans l'URI, spécifiez les valeurs appropriées pour les paramètres de chemin requis :

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Où :

- *endpoint*est le point de terminaison Amazon Pinpoint pour la AWS région qui héberge le projet.
- *application-id* est l'identifiant unique du projet.
- *kpi-name*est la `kpi-name` valeur de la métrique à interroger.

Tous les paramètres doivent être codés par URL.

Pour appliquer un filtre qui interroge les données pour une plage de dates spécifique, ajoutez les paramètres `start-time` et `end-time` et leurs valeurs à l'URI. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Utilisez une esperluette (&) pour séparer les paramètres.

Par exemple, la demande suivante récupère le nombre de messages électroniques transactionnels envoyés pour un projet du 6 septembre 2019 au 13 septembre 2019 :

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-emails-sent?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-13T23:59:59Z
```

Où :

- `pinpoint.us-east-1.amazonaws.com` est le point de terminaison Amazon Pinpoint de la région AWS qui héberge le projet.
- `12345678901234567890123456example` est l'identifiant unique du projet.
- `txn-emails-sent` est la valeur `kpi-name` de la métrique sends (envois) de l'application. Cette métrique indique le nombre de messages électroniques transactionnels envoyés pour un projet.

- `2019-09-06T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-09-13T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

AWS CLI

Pour interroger les données d'analyse des e-mails transactionnels à l'aide de AWS CLI, utilisez la get-application-date-range-kpi commande et spécifiez les valeurs appropriées pour les paramètres requis :

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Où :

- *`application-id`* est l'identifiant unique du projet.
- *`kpi-name`*est la `kpi-name` valeur de la métrique à interroger.

Pour appliquer un filtre qui interroge les données pour une plage de dates spécifique, ajoutez les paramètres `start-time` et`end-time` et leurs valeurs à votre requête. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Par exemple, la demande suivante récupère le nombre de messages électroniques transactionnels envoyés pour un projet du 6 septembre 2019 au 13 septembre 2019 :

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-emails-sent ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-13T23:59:59Z
```

Où :

- `12345678901234567890123456example` est l'identifiant unique du projet.

- `txn-emails-sent` est la valeur `kpi-name` de la métrique sends (envois) de l'application. Cette métrique indique le nombre de messages électroniques transactionnels envoyés pour un projet.
- `2019-09-06T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-09-13T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

SDK for Java

Pour interroger les données d'analyse des e-mails transactionnels à l'aide de AWS SDK pour Java, utilisez la GetApplicationDateRangeKpiRequest méthode de l'API Application Metrics. Spécifiez les valeurs appropriées pour les paramètres requis :

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Où :

- *`applicationId`* est l'identifiant unique du projet.
- *`kpiName`*est la `kpi-name` valeur de la métrique à interroger.

Pour appliquer un filtre qui interroge les données pour une plage de dates spécifique, incluez les paramètres `startTime` et`endTime` et leurs valeurs dans votre requête. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Par exemple, la demande suivante récupère le nombre de messages électroniques transactionnels envoyés pour un projet du 6 septembre 2019 au 13 septembre 2019 :

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-emails-sent")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

Où :

- `12345678901234567890123456example` est l'identifiant unique du projet.
- `txn-emails-sent` est la valeur `kpi-name` de la métrique sends (envois) de l'application. Cette métrique indique le nombre de messages électroniques transactionnels envoyés pour un projet.
- `2019-09-06T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-09-13T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

Une fois que vous avez envoyé votre requête, Amazon Pinpoint renvoie les résultats de la requête dans une réponse JSON. La structure des résultats varie en fonction de la métrique que vous avez interrogée. Certaines mesures ne renvoient qu'une seule valeur. Par exemple, la métrique d'application envois (`txn-emails-sent`), utilisée dans les exemples précédents, renvoie une seule valeur : le nombre d'e-mails transactionnels envoyés à partir d'un projet. Dans ce cas, la réponse JSON est la suivante :

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-13T23:59:59Z",
        "KpiName":"txn-emails-sent",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"TxnEmailsSent",
                            "Type":"Double",
                            "Value":"62.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

D'autres métriques renvoient plusieurs valeurs et regroupent les valeurs selon un champ pertinent. Si une mesure renvoie plusieurs valeurs, la réponse JSON inclut un champ qui indique quel champ a été utilisé pour regrouper les données.

Pour en savoir plus sur la structure des résultats de requête, reportez-vous à la section Utiliser les résultats des requêtes JSON.

Interrogez les données Amazon Pinpoint pour les SMS transactionnels

Pour interroger les données des messages SMS transactionnels envoyés pour un projet, utilisez l'API des métriques d'application et spécifiez les valeurs pour les paramètres requis suivants :

- application-id : ID du projet, qui est l'identifiant unique du projet. Dans Amazon Pinpoint, les termes projet et application ont la même signification.
- kpi-name : nom de la métrique à interroger. Cette valeur décrit la métrique associée et se compose de deux termes ou plus, composés de caractères alphanumériques minuscules, séparés par un trait d'union. Pour obtenir la liste complète des métriques prises en charge et la valeur `kpi-name` de chacune d'elles, reportez-vous à la section Mesures standard pour les projets, les campagnes et les parcours.

Vous pouvez également appliquer un filtre qui interroge les données pour une plage de dates spécifique. Si vous ne spécifiez pas de plage de dates, Amazon Pinpoint renvoie les données des 31 jours civils précédents. Pour filtrer les données selon différentes dates, utilisez les paramètres de plage de dates pris en charge pour spécifier les premières et dernières date et heure de la plage de dates. Les valeurs doivent être au format ISO 8601 étendu et utiliser le temps universel coordonné (UTC). Par exemple, `2019-09-06T20:00:00Z` pour le 6 septembre 2019 à 20h00 UTC. Notez que les plages de dates sont inclusives et doivent être limitées à 31 jours calendaires au plus. En outre, les premières date et heure doivent être inférieures à 90 jours à compter du jour actuel.

Les exemples suivants montrent comment interroger des données d'analyse pour des messages SMS transactionnels à l'aide de l'API REST Amazon Pinpoint, AWS CLI du, et du. AWS SDK pour Java Vous pouvez utiliser n'importe quel AWS SDK compatible pour interroger les données d'analyse des messages transactionnels. Les AWS CLI exemples sont formatés pour Microsoft Windows. Pour Unix, Linux et macOS, remplacez le caractère caret (^) de continuation de ligne par une barre oblique inverse (\).

REST API

Pour interroger les données analytiques des messages SMS transactionnels à l'aide de l'API REST Amazon Pinpoint, envoyez une demande HTTP(S) GET à l'URI des Métriques d'application. Dans l'URI, spécifiez les valeurs appropriées pour les paramètres de chemin requis :

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Où :

- *endpoint*est le point de terminaison Amazon Pinpoint pour la AWS région qui héberge le projet.
- *application-id* est l'identifiant unique du projet.
- *kpi-name*est la `kpi-name` valeur de la métrique à interroger.

Tous les paramètres doivent être codés par URL.

Pour appliquer un filtre qui extrait les données d'une plage de dates spécifique, ajoutez les paramètres de requête `start-time` et `end-time` et leurs valeurs à l'URI. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Utilisez une esperluette (&) pour séparer les paramètres.

Par exemple, la demande suivante récupère le nombre de messages SMS transactionnels envoyés chaque jour du 6 septembre 2019 au 8 septembre 2019 :

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-08T23:59:59Z
```

Où :

- `pinpoint.us-east-1.amazonaws.com` est le point de terminaison Amazon Pinpoint de la région AWS qui héberge le projet.
- `12345678901234567890123456example` est l'identifiant unique du projet.
- `txn-sms-sent-grouped-by-date` est la valeur `kpi-name` de la métrique sends, grouped by date (envois, regroupés par date) de l'application. Cette métrique renvoie le nombre de messages SMS transactionnels envoyés au cours de chaque jour de la plage de dates.

- `2019-09-06T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-09-08T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

AWS CLI

Pour interroger les données d'analyse des messages SMS transactionnels à l'aide de AWS CLI, utilisez la get-application-date-range-kpi commande et spécifiez les valeurs appropriées pour les paramètres requis :

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Où :

- *`application-id`* est l'identifiant unique du projet.
- *`kpi-name`*est la `kpi-name` valeur de la métrique à interroger.

Pour appliquer un filtre qui extrait les données d'une plage de dates spécifique, incluez les paramètres `start-time` et `end-time` et leurs valeurs dans votre requête. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Par exemple, la demande suivante récupère le nombre de messages SMS transactionnels envoyés chaque jour du 6 septembre 2019 au 8 septembre 2019 :

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-sms-sent-grouped-by-date ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-08T23:59:59Z
```

Où :

- `12345678901234567890123456example` est l'identifiant unique du projet.

- `txn-sms-sent-grouped-by-date` est la valeur `kpi-name` de la métrique sends, grouped by date (envois, regroupés par date) de l'application. Cette métrique renvoie le nombre de messages SMS transactionnels envoyés au cours de chaque jour de la plage de dates.
- `2019-09-06T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-09-08T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

SDK for Java

Pour interroger les données d'analyse des messages SMS transactionnels à l'aide de AWS SDK pour Java, utilisez la GetApplicationDateRangeKpiRequest méthode de l'API Application Metrics et spécifiez les valeurs appropriées pour les paramètres requis :

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Où :

- *`applicationId`* est l'identifiant unique du projet.
- *`kpiName`*est la `kpi-name` valeur de la métrique à interroger.

Pour appliquer un filtre qui extrait les données d'une plage de dates spécifique, incluez les paramètres `startTime` et `endTime` et leurs valeurs dans votre requête. À l'aide de ces paramètres, vous pouvez spécifier les premières et dernières date et l'heure, au format ISO 8601 étendu, d'une plage de dates incluse pour récupérer les données. Par exemple, la demande suivante récupère le nombre de messages SMS transactionnels envoyés chaque jour du 6 septembre 2019 au 8 septembre 2019 :

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-sms-sent-grouped-by-date")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-08T23:59:59Z")));
```

Où :

- `12345678901234567890123456example` est l'identifiant unique du projet.
- `txn-sms-sent-grouped-by-date` est la valeur `kpi-name` de la métrique sends, grouped by date (envois, regroupés par date) de l'application. Cette métrique renvoie le nombre de messages SMS transactionnels envoyés au cours de chaque jour de la plage de dates.
- `2019-09-06T00:00:00Z` est la première date et heure à laquelle récupérer les données, dans le cadre d'une plage de dates inclusive.
- `2019-09-08T23:59:59Z` est la dernière date et heure auxquelles récupérer les données, dans le cadre d'une plage de dates inclusive.

Une fois que vous avez envoyé votre requête, Amazon Pinpoint renvoie les résultats de la requête dans une réponse JSON. La structure des résultats varie en fonction de la métrique que vous avez interrogée. Certaines mesures ne renvoient qu'une seule valeur. D'autres métriques renvoient plusieurs valeurs et regroupent ces valeurs selon un champ pertinent. Si une mesure renvoie plusieurs valeurs, la réponse JSON inclut un champ qui indique quel champ a été utilisé pour regrouper les données.

Par exemple, la métrique d'application des envois, regroupés par date (`txn-sms-sent-grouped-by-date`) utilisée dans les exemples précédents, renvoie plusieurs valeurs : nombre de messages SMS transactionnels envoyés au cours de chaque jour de la plage de dates spécifiée. Dans ce cas, la réponse JSON est la suivante :


```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-08T23:59:59Z",
        "KpiName":"txn-sms-sent-grouped-by-date",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
                            "Type":"String",
                            "Value":"2019-09-06"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"TxnSmsSent",
```

```
                            "Type":"Double",
                            "Value":"29.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
                            "Type":"String",
                            "Value":"2019-09-07"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"TxnSmsSent",
                            "Type":"Double",
                            "Value":"35.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
                            "Type":"String",
                            "Value":"2019-09-08"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"TxnSmsSent",
                            "Type":"Double",
                            "Value":"10.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

Dans ce cas, le champ `GroupedBys` indique que les valeurs sont regroupées par jour calendaire (`Date`). Cela signifie que :

- 29 messages ont été envoyés le 6 septembre 2019.
- 35 messages ont été envoyés le 7 septembre 2019.
- 10 messages ont été envoyés le 8 septembre 2019.

Pour en savoir plus sur la structure des résultats de requête, reportez-vous à la section Utiliser les résultats des requêtes JSON.

Utiliser les résultats des requêtes JSON d'Amazon Pinpoint Analytics

Lorsque vous utilisez Amazon Pinpoint Analytics APIs pour interroger des données d'analyse, Amazon Pinpoint renvoie les résultats dans une réponse JSON. Pour les métriques d'application, de campagne et d'engagement du parcours, les données de la réponse sont conformes à un schéma JSON standard pour le reporting des données analytiques Amazon Pinpoint.

Cela signifie que vous pouvez utiliser le langage de programmation ou l'outil de votre choix pour implémenter une solution personnalisée qui interroge les données pour une ou plusieurs de ces mesures, capture les résultats de chaque requête, puis écrit les résultats dans une table, un objet ou un autre emplacement. Vous pouvez ensuite utiliser les résultats de la requête dans cet emplacement à l'aide d'un autre service ou application.

Par exemple, vous pouvez effectuer les actions suivantes :

- Créez un tableau de bord personnalisé qui interroge régulièrement un ensemble de métriques et affiche les résultats à l'aide de votre infrastructure de visualisation de données préférée.
- Créez un rapport qui suit les taux d'engagement en interrogeant les métriques appropriées et en affichant les résultats dans un graphique ou un autre type de rapport que vous concevez.
- Analysez et écrivez les données analytiques dans un format de stockage particulier, puis portez les résultats vers une solution de stockage à long terme.

Notez qu'Amazon Pinpoint Analytics n'est APIs pas conçu pour créer ou stocker des objets persistants que vous pouvez ensuite lire ou utiliser dans un projet Amazon Pinpoint ou dans votre compte Amazon Pinpoint. Ils APIs sont plutôt conçus pour vous aider à récupérer des données

analytiques et à les transférer vers d'autres services et applications pour une analyse, un stockage ou des rapports plus approfondis. Ils le font en partie en utilisant la même structure de réponse JSON et le même schéma pour toutes les données d'analyse que vous pouvez interroger par programmation pour les métriques d'application, de campagne et d'engagement du parcours.

Cette rubrique explique la structure, les objets et les champs d'une réponse JSON à une requête pour une métrique d'application, de campagne ou d'engagement du parcours. Pour de plus amples informations sur les champs d'une réponse JSON à une requête pour une métrique d'exécution du parcours ou d'exécution de l'activité du parcours, veuillez consulter Mesures standard qui s'appliquent aux projets, campagnes et parcours Amazon Pinpoint.

Structure JSON

Pour vous aider à analyser et à utiliser les résultats des requêtes, Amazon Pinpoint APIs Analytics utilise la même structure de réponse JSON pour toutes les données d'analyse Amazon Pinpoint que vous pouvez interroger par programmation pour obtenir des statistiques d'application, des statistiques de campagne et des statistiques d'engagement sur le parcours. Chaque réponse JSON spécifie les valeurs ayant défini la requête, telles que l'ID de projet (`ApplicationId`). La réponse inclut également un (et un seul) objet `KpiResult`. L'objet `KpiResult` contient le jeu de résultats global d'une requête.

Chaque objet `KpiResult` contient un objet `Rows`. Il s'agit d'un tableau d'objets contenant des résultats de requête et des métadonnées pertinentes sur les valeurs de ces résultats. La structure et le contenu d'un objet `Rows` présentent les caractéristiques générales suivantes :

- Chaque ligne de résultats de requête est un objet JSON distinct, nommé `Values`, dans l'objet `Rows`. Par exemple, si une requête renvoie trois valeurs, l'objet `Rows` contient trois objets `Values`. Chaque objet `Values` contient un résultat individuel pour la requête.
- Chaque colonne de résultats de requête est une propriété de l'objet `Values` auquel elle s'applique. Le nom de la colonne est stocké dans le champ `Key` de l'objet `Values`.
- Pour les résultats de requête groupés, chaque objet `Values` a un objet `GroupedBys` associé. L'objet `GroupedBys` indique quel champ a été utilisé pour regrouper les résultats. Il fournit également la valeur de regroupement de l'objet `Values` associé.
- Si le résultat de la requête pour une métrique est nul, l'objet `Rows` est vide.

Au-delà de ces caractéristiques générales, la structure et le contenu de l'objet `Rows` varient en fonction de la métrique. En effet, Amazon Pinpoint prend en charge deux types de métriques, les métriques à valeur unique et les métriques à valeur multiples.

Une métrique à valeur unique ne fournit qu'une seule valeur cumulative. Un exemple est le pourcentage de messages remis aux destinataires par toutes les exécutions d'une campagne. Une métrique à valeurs multiples fournit plus d'une valeur et regroupe ces valeurs par un champ pertinent. Un exemple est le pourcentage de messages remis aux destinataires pour chaque exécution d'une campagne, regroupés par exécution de campagne.

Vous pouvez déterminer rapidement si une métrique est à valeur unique ou à valeur multiple en faisant référence à son nom. Si le nom ne contient pas `grouped-by`, il s'agit d'une métrique à valeur unique. Dans le cas contraire, il s'agit d'une métrique à plusieurs valeurs. Pour obtenir la liste complète des métriques que vous pouvez interroger par programme, reportez-vous à la section Mesures standard qui s'appliquent aux projets, campagnes et parcours Amazon Pinpoint.

Métriques à valeur unique

Pour une métrique à valeur unique, l'objet `Rows` contient un objet `Values` qui :

- spécifie le nom convivial de la mesure interrogée ;
- fournit la valeur de la mesure interrogée ;
- identifie le type de données de la valeur renvoyée.

Par exemple, la réponse JSON suivante contient les résultats de la requête pour une métrique à valeur unique. Cette métrique indique le nombre de points de terminaison uniques auxquels les messages ont été remis par toutes les campagnes associées à un projet, du 1er au 31 août 2019 :

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
```

```
                            "Value":"1368.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Dans cet exemple, la réponse indique que toutes les campagnes du projet ont remis des messages à 1 368 points de terminaison uniques du 1er au 31 août 2019, dans lesquels :

- `Key` est le nom convivial de la métrique dont la valeur est spécifiée dans le champ `Value` (`UniqueDeliveries`).
- `Type` est le type de données de la valeur spécifiée dans le champ `Value` (`Double`).
- `Value` est la valeur réelle de la métrique interrogée, y compris les filtres appliqués (`1368.0`).

Si le résultat de la requête pour une métrique à valeur unique est null (pas supérieur ou égal à zéro), l'objet `Rows` est vide. Amazon Pinpoint renvoie une valeur nulle pour une métrique s'il n'y a aucune donnée à renvoyer pour cette métrique. Par exemple :

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"234567890123456789012345example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[

            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Métriques à valeurs multiples

La structure et le contenu de l'objet `Rows` pour une métrique à valeurs multiples sont essentiellement les mêmes que ceux d'une métrique à valeur unique. L'objet `Rows` d'une métrique à valeurs multiples

contient également un objet `Values`. L'objet `Values` spécifie le nom convivial de la métrique interrogée, fournit la valeur de cette métrique et identifie le type de données de cette valeur.

Toutefois, l'objet `Rows` objet d'une métrique à plusieurs valeurs contient également un ou plusieurs objets `GroupedBy`. Il y a un objet `GroupedBy` pour chaque objet `Values` des résultats de la requête. L'objet `GroupedBy` indique quel champ a été utilisé pour regrouper les données dans les résultats et le type de données de ce champ. Il indique également la valeur de regroupement pour ce champ (pour l'objet `Values` associé).

Par exemple, la réponse JSON suivante contient les résultats de la requête pour une mesure à valeurs multiples qui indique le nombre de points de terminaison uniques auxquels les messages ont été remis, pour chaque campagne associée à un projet, du 1er août 2019 au 31 août 2019 :

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
```

```
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Dans cet exemple, la réponse indique que trois des campagnes du projet ont transmis des messages à des points de terminaison uniques du 1er août 2019 au 31 août 2019. Pour chacune de ces campagnes, la ventilation du nombre de livraisons est la suivante :

- La campagne `80b8efd84042ff8d9c96ce2f8example` a envoyé des messages à 123 points de terminaison uniques.
- La campagne `810c7aab86d42fb2b56c8c966example` a envoyé des messages à 456 points de terminaison uniques.
- La campagne `42d8c7eb0990a57ba1d5476a3example` a envoyé des messages à 789 points de terminaison uniques.

Lorsque la structure générale des objets et des champs est :

- `GroupedBys.Key` : nom de la propriété ou du champ qui stocke la valeur de regroupement spécifiée dans le champ `GroupedBys.Value` (`CampaignId`).
- `GroupedBys.Type` : type de données de la valeur spécifiée dans le champ `GroupedBys.Value` (`String`).
- `GroupedBys.Value` : valeur réelle du champ utilisé pour regrouper les données, comme spécifié dans le champ `GroupedBys.Key` (ID de campagne).
- `Values.Key` : nom convivial de la métrique dont la valeur est spécifiée dans le champ `Values.Value` (`UniqueDeliveries`).
- `Values.Type` : type de données de la valeur spécifiée dans le champ `Values.Value` (`Double`).
- `Values.Value` : valeur réelle de la métrique interrogée, y compris les filtres appliqués.

Si le résultat de la requête pour une métrique à valeurs multiples est null (pas supérieur ou égal à zéro) pour un projet, une campagne ou une autre ressource spécifique, Amazon Pinpoint ne renvoie aucun objet ou champ pour la ressource. Si le résultat de la requête pour une métrique à valeurs multiples est null pour toutes les ressources, Amazon Pinpoint renvoie un objet `Rows` vide.

Objets et champs JSON

Outre la spécification des valeurs ayant défini une requête, telles que l'ID de projet (`ApplicationId`), chaque réponse JSON à une requête pour une métrique d'application, de campagne ou d'engagement du parcours inclut un objet `KpiResult`. Cet objet contient le jeu de résultats global d'une requête, que vous pouvez analyser pour envoyer les données d'analyse à un autre service ou application. Chaque objet `KpiResult` contient une partie ou la totalité des objets et champs standard suivants, selon la métrique.

Objet ou champ	Description
`Rows`	Tableau d'objets contenant le jeu de résultats d'une requête.
`Rows.GroupedBys`	Pour une métrique à valeurs multiples, un tableau de champs qui définit le champ et les valeurs utilisées pour regrouper des données dans les résultats de la requête.

Objet ou champ	Description
`Rows.GroupedBys.Key`	Pour une métrique à valeurs multiples, nom de la propriété ou du champ qui stocke la valeur spécifiée dans le champ `GroupedBys.Value` .
`Rows.GroupedBys.Type`	Pour une métrique à valeurs multiples, type de données de la valeur spécifiée dans le champ `GroupedBys.Value` .
`Rows.GroupedBys.Value`	Pour une métrique à plusieurs valeurs, la valeur réelle du champ utilisé pour regrouper des données dans les résultats de la requête. Cette valeur est en corrélation avec un objet `Values` associé.
`Rows.Values`	Tableau de champs contenant les résultats de la requête.
`Rows.Values.Key`	Le nom convivial de la mesure interrogée. La valeur de la métrique est spécifiée dans le champ `Values.Value` .
`Rows.Values.Type`	Type de données de la valeur spécifiée dans le champ `Values.Value` .
`Rows.Values.Value`	Valeur réelle de la métrique interrogée, y compris les filtres appliqués.

Pour de plus amples informations sur les champs d'une réponse JSON à une requête pour une métrique d'exécution du parcours ou d'exécution de l'activité du parcours, veuillez consulter Mesures standard qui s'appliquent aux projets, campagnes et parcours Amazon Pinpoint.

Enregistrez les appels d'API Amazon Pinpoint avec AWS CloudTrail

Amazon Pinpoint est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou un AWS service dans Amazon Pinpoint. CloudTrail capture les appels d'API pour Amazon Pinpoint sous forme d'événements. Les appels capturés incluent les appels de la console Amazon Pinpoint et les appels de code vers les opérations d'API Amazon Pinpoint.

Si vous créez un suivi, vous pouvez activer la diffusion continue des CloudTrail événements vers un bucket Amazon Simple Storage Service (Amazon S3), y compris les événements pour Amazon Pinpoint. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents en utilisant l'historique des événements sur la CloudTrail console. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon Pinpoint, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le guide de AWS CloudTrail l'utilisateur.

Informations Amazon Pinpoint dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité événementielle prise en charge se produit dans Amazon Pinpoint, 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 AWS compte. Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour Amazon Pinpoint, créez un historique. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. 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

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

- Si la demande a été faite avec les informations Gestion des identités et des accès AWS d'identification root ou utilisateur.
- 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 Élément CloudTrail userIdentity.

Vous pouvez créez un journal de suivi et stocker vos fichiers journaux dans votre compartiment Amazon S3 aussi longtemps que vous le souhaitez. Vous pouvez également définir des règles de cycle de vie d'Amazon S3 pour archiver ou supprimer les fichiers journaux automatiquement. Par défaut, vos fichiers journaux sont chiffrés avec le chiffrement côté serveur (SSE) d'Amazon S3.

Pour être informé de la livraison des fichiers journaux, configurez CloudTrail pour publier des notifications Amazon SNS lorsque de nouveaux fichiers journaux sont livrés. Pour plus d'informations, consultez Configuration des notifications Amazon SNS pour CloudTrail.

Vous pouvez également agréger les fichiers journaux Amazon Pinpoint provenant de plusieurs AWS régions et de plusieurs AWS comptes dans un seul compartiment Amazon S3. Pour plus d'informations, consultez Réception des fichiers journaux CloudTrail de plusieurs régions et Réception des fichiers journaux CloudTrail de plusieurs comptes.

Vous pouvez l'utiliser CloudTrail pour enregistrer les actions pour l'Amazon APIs Pinpoint suivant :

- API Amazon Pinpoint
- API SMS et voix Amazon Pinpoint

Actions d'API Amazon Pinpoint prises en charge dans les fichiers journaux CloudTrail

L'API Amazon Pinpoint permet de consigner les actions suivantes sous forme d'événements dans des fichiers CloudTrail journaux :

- CreateApp
- CreateCampaign
- CreateEmailTemplate
- CreateExportJob
- CreateImportJob
- CreateJourney
- CreatePushTemplate
- CreateRecommenderConfiguration
- CreateSegment
- CreateSmsTemplate
- CreateVoiceTemplate
- DeleteAdmChannel
- DeleteApnsChannel
- DeleteApnsSandboxChannel
- DeleteApnsVoipChannel
- DeleteApnsVoipSandboxChannel
- DeleteApp
- DeleteBaiduChannel
- DeleteCampaign
- DeleteEmailChannel
- DeleteEmailTemplate
- DeleteEndpoint
- DeleteEventStream
- DeleteGcmChannel

- DeleteJourney
- DeletePushTemplate
- DeleteRecommenderConfiguration
- DeleteSegment
- DeleteSmsChannel
- DeleteSmsTemplate
- DeleteUserEndpoints
- DeleteVoiceChannel
- DeleteVoiceTemplate
- GetAdmChannel
- GetApnsChannel
- GetApnsSandboxChannel
- GetApnsVoipChannel
- GetApnsVoipSandboxChannel
- GetApp
- GetApplicationDateRangeKpi
- GetApplicationSettings
- GetApps
- GetBaiduChannel
- GetCampaign
- GetCampaignActivities
- GetCampaignDateRangeKpi
- GetCampaignVersion
- GetCampaignVersions
- GetCampaigns
- GetChannels
- GetEmailChannel
- GetEmailTemplate
- GetEndpoint

- GetEventStream
- GetExportJob
- GetExportJobs
- GetGcmChannel
- GetImportJob
- GetImportJobs
- GetJourney
- GetJourneyDateRangeKpi
- GetJourneyExecutionActivityMetrics
- GetJourneyExecutionMetrics
- GetPushTemplate
- GetRecommenderConfiguration
- GetRecommenderConfigurations
- GetSegment
- GetSegmentExportJobs
- GetSegmentImportJobs
- GetSegmentVersion
- GetSegmentVersions
- GetSegments
- GetSmsChannel
- GetSmsTemplate
- GetUserEndpoints
- GetVoiceChannel
- GetVoiceTemplate
- ListJourneys
- ListTagsForResource
- ListTemplates
- ListTemplateVersions
- PhoneNumberValidate

- PutEvents
- PutEventStream
- RemoveAttributes
- SendMessages
- Envoyer OTPMessage
- SendUsersMessages
- TagResource
- UntagResource
- UpdateAdmChannel
- UpdateApnsChannel
- UpdateApnsSandboxChannel
- UpdateApnsVoipChannel
- UpdateApnsVoipSandboxChannel
- UpdateApplicationSettings
- UpdateBaiduChannel
- UpdateCampaign
- UpdateEmailChannel
- UpdateEmailTemplate
- UpdateEndpoint
- UpdateEndpointsBatch
- UpdateGcmChannel
- UpdateJourney
- UpdateJourneyState
- UpdatePushTemplate
- UpdateRecommenderConfiguration
- UpdateSegment
- UpdateSmsChannel
- UpdateSmsTemplate
- UpdateTemplateActiveVersion
- UpdateVoiceChannel

- UpdateVoiceTemplate

Actions de l'API de messagerie Amazon Pinpoint prises en charge dans les fichiers journaux CloudTrail

L'API Amazon Pinpoint Email prend en charge l'enregistrement des actions suivantes sous forme d'événements dans des fichiers CloudTrail journaux :

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- CreateDedicatedIpPool
- CreateEmailIdentity
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- DeleteDedicatedIpPool
- DeleteEmailIdentity
- GetAccount
- GetConfigurationSet
- GetConfigurationSetEventDestinations
- GetDedicatedIp
- GetDedicatedIps
- GetEmailIdentity
- ListConfigurationSets
- ListDedicatedIpPools
- ListEmailIdentities
- PutAccountDedicatedIpWarmupAttributes
- PutAccountSendingAttributes
- PutConfigurationSetDeliveryOptions
- PutConfigurationSetReputationOptions
- PutConfigurationSetSendingOptions
- PutConfigurationSetTrackingOptions

- PutDedicatedIpInPool
- PutDedicatedIpWarmupAttributes
- PutEmailIdentityDkimAttributes
- PutEmailIdentityFeedbackAttributes
- PutEmailIdentityMailFromAttributes
- UpdateConfigurationSetEventDestination

L'action suivante de l'API Amazon Pinpoint Email n'est pas connectée : CloudTrail

- SendEmail

Actions de la version 1 de l'API vocale et SMS Amazon Pinpoint prises en charge dans les fichiers journaux CloudTrail

L'API Amazon Pinpoint SMS and Voice version 1 permet de consigner les actions suivantes sous forme d'événements dans des fichiers CloudTrail journaux :

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- GetConfigurationSetEventDestinations
- UpdateConfigurationSetEventDestination

L'action d'API Amazon Pinpoint SMS and Voice version 1 suivante n'est pas connectée : CloudTrail

- SendVoiceMessage

CloudTrail exemples de saisie de journal illustrant les actions de l'API Amazon Pinpoint

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 individuelle d'une source quelconque. Il 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.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre les `CreateCampaign` actions `GetCampaigns` et les actions de l'API Amazon Pinpoint.


```
{
  "Records": [
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "GetCampaigns",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T00:56:48Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": true,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "application-id": "example71dfa4c1aab66332a5839798f",
        "page-size": "1000"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Jersey/${project.version} (HttpUrlConnection 1.8.0_144)",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "principalId": "123456789012",
        "sessionContext": {
          "attributes": {
            "creationDate": "2018-02-02T16:55:29Z",
            "mfaAuthenticated": "false"
          }
        },
        "type": "Root"
      }
    },
    {
```

```
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "CreateCampaign",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T01:05:16Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": false,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "Description": "***",
        "HoldoutPercent": 0,
        "IsPaused": false,
        "MessageConfiguration": "***",
        "Name": "***",
        "Schedule": {
          "Frequency": "ONCE",
          "IsLocalTime": true,
          "StartTime": "2018-02-03T00:00:00-08:00",
          "Timezone": "utc-08"
        },
        "SegmentId": "exampleda204adf991a80281aa0e591",
        "SegmentVersion": 1,
        "application-id": "example71dfa4c1aab66332a5839798f"
      },
      "responseElements": {
        "ApplicationId": "example71dfa4c1aab66332a5839798f",
        "CreationDate": "2018-02-03T01:05:16.425Z",
        "Description": "***",
        "HoldoutPercent": 0,
        "Id": "example54a654f80948680cbba240ede",
        "IsPaused": false,
        "LastModifiedDate": "2018-02-03T01:05:16.425Z",
        "MessageConfiguration": "***",
        "Name": "***",
        "Schedule": {
          "Frequency": "ONCE",
          "IsLocalTime": true,
          "StartTime": "2018-02-03T00:00:00-08:00",
          "Timezone": "utc-08"
        },
        "SegmentId": "example4da204adf991a80281example",
        "SegmentVersion": 1,
```

```
        "State": {
          "CampaignStatus": "SCHEDULED"
        },
        "Version": 1
      },
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.9 Python/3.4.3 Linux/3.4.0+ botocore/1.8.34",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/userName",
        "principalId": "AIDAIHTHRCDA62EXAMPLE",
        "type": "IAMUser",
        "userName": "userName"
      }
    }
  ]
}
```

L'exemple suivant montre une entrée de CloudTrail journal illustrant les `CreateConfigurationSetEventDestination` actions `CreateConfigurationSet` et dans l'API SMS et voix Amazon Pinpoint.

```
{
  "Records": [
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:45:55Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSet",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "ConfigurationSetName":"MyConfigurationSet"
```

```
      },
      "responseElements":null,
      "requestID":"56dcc091-e20d-11e8-87d2-9994aexample",
      "eventID":"725843fc-8846-41f4-871a-7c52dexample",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "recipientAccountId":"123456789012"
    },
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:47:08Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSetEventDestination",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "EventDestinationName":"CloudWatchEventDestination",
        "ConfigurationSetName":"MyConfigurationSet",
        "EventDestination":{
          "Enabled":true,
          "MatchingEventTypes":[
            "INITIATED_CALL",
            "INITIATED_CALL"
          ],
          "CloudWatchLogsDestination":{
            "IamRoleArn":"arn:aws:iam::111122223333:role/iamrole-01",
            "LogGroupArn":"arn:aws:logs:us-east-1:111122223333:log-
group:clientloggroup-01"
          }
        }
      },
      "responseElements":null,
      "requestID":"81de1e73-e20d-11e8-b158-d5536example",
      "eventID":"fcafc21f-7c93-4a3f-9e72-fca2dexample",
      "readOnly":false,
```

```
      "eventType":"AwsApiCall",
      "recipientAccountId":"111122223333"
    }
  ]
}
```

Utilisez un modèle de recommandation dans Amazon Pinpoint avec AWS Lambda

Dans Amazon Pinpoint, vous pouvez récupérer des recommandations personnalisées à partir d'un modèle de recommandation et les ajouter aux messages que vous envoyez à partir de campagnes et de parcours. Un modèle de recommandation est un type de modèle de machine learning qui trouve des modèles dans les données et génère des prédictions et des recommandations basées sur les modèles qu'il trouve. Il prédit ce qu'un utilisateur particulier préférera à partir d'un ensemble donné de produits ou d'articles, et il fournit ces informations sous la forme d'un ensemble de recommandations pour l'utilisateur.

En utilisant des modèles de recommandation avec Amazon Pinpoint, vous pouvez envoyer des recommandations personnalisées aux destinataires des messages en fonction des attributs et du comportement de chaque destinataire. Avec AWS Lambda, vous pouvez également personnaliser et améliorer ces recommandations. Par exemple, vous pouvez transformer dynamiquement une recommandation d'une valeur de texte unique (par exemple un nom ou un ID de produit) en contenu plus sophistiqué (par exemple un nom de produit, une description et une image). Et vous pouvez le faire en temps réel, quand Amazon Pinpoint envoie le message.

Cette fonctionnalité est disponible dans les AWS régions suivantes : USA Est (Virginie du Nord) ; USA Ouest (Oregon) ; Asie-Pacifique (Mumbai) ; Asie-Pacifique (Sydney) ; et Europe (Irlande).

Ajouter des recommandations de modèles de recommandation aux messages dans Amazon Pinpoint

Pour utiliser un modèle de recommandation avec Amazon Pinpoint, vous commencez par créer une solution Amazon Personalize et par déployer cette solution en tant que campagne Amazon Personalize. Ensuite, vous créez une configuration pour le modèle de recommandation dans Amazon Pinpoint. Dans la configuration, vous spécifiez les paramètres qui déterminent comment récupérer et traiter les données de recommandation de la campagne Amazon Personalize. Cela inclut l'opportunité d'invoquer une AWS Lambda fonction pour effectuer un traitement supplémentaire des données récupérées.

Amazon Personalize est un AWS service conçu pour vous aider à créer des modèles de machine learning fournissant des recommandations personnalisées en temps réel aux clients qui utilisent vos applications. Amazon Personalize vous guide tout au long du processus de création et

d'entraînement d'un modèle de machine learning, puis de préparation et de déploiement du modèle sous forme de campagne Amazon Personalize. Vous pouvez ensuite récupérer des recommandations personnalisées et en temps réel de la campagne. Pour en savoir plus sur Amazon Personalize, consultez le Guide du développeur Amazon Personalize.

AWS Lambda est un service de calcul que vous pouvez utiliser pour exécuter du code sans provisionner ni gérer de serveurs. Vous empaqueter votre code et le télécharger en AWS Lambda tant que fonction Lambda. AWS Lambda exécute ensuite la fonction lorsque la fonction est invoquée. Une fonction peut être appelée manuellement, par vous, automatiquement en réponse à des événements, ou en réponse à des demandes d'applications ou de services, y compris Amazon Pinpoint. Pour en savoir plus sur la création et l'invocation de fonctions Lambda, consultez le Guide du développeur AWS Lambda.

Après avoir créé une configuration Amazon Pinpoint pour un modèle de recommandation, vous pouvez ajouter des recommandations du modèle aux messages que vous envoyez à partir de campagnes et de parcours. Pour cela, utilisez des modèles de message qui contiennent des variables de message pour les attributs recommandés. Un attribut recommandé est un point de terminaison dynamique ou un attribut utilisateur conçu pour stocker les données de recommandation. Vous définissez ces attributs lorsque vous créez la configuration d'un modèle de recommandation.

Vous pouvez utiliser des variables pour les attributs recommandés dans les types de modèles de message suivants :

- Modèles d'e-mail, pour les messages électroniques que vous envoyez à partir de campagnes ou de parcours
- Modèles de notification Push, pour les notifications Push que vous envoyez à partir de campagnes
- Modèles SMS, pour les SMS que vous envoyez à partir de campagnes

Pour plus d'informations sur l'utilisation des modèles de recommandation avec Amazon Pinpoint, consultez Modèles de machine learning dans le Guide de l'utilisateur d'Amazon Pinpoint.

Si vous configurez Amazon Pinpoint pour invoquer une fonction Lambda qui traite les données de recommandation, Amazon Pinpoint exécute les tâches générales suivantes chaque fois qu'il envoie des recommandations personnalisées dans un message pour une campagne ou un parcours :

1. Évalue et traite les paramètres de configuration et le contenu du message et du modèle de message.
2. Détermine que le modèle de message est connecté à un modèle de recommandation.

3. Évalue les paramètres de configuration pour la connexion au modèle et l'utilisation du modèle. Ces paramètres sont définis par la ressource Recommender Model du modèle.
4. Détecte une ou plusieurs variables de message pour les attributs recommandés qui sont définis par les paramètres de configuration du modèle.
5. Récupère les données de recommandation de la campagne Amazon Personalize spécifiée dans les paramètres de configuration du modèle. Il utilise le GetRecommendationsfonctionnement de l'API Amazon Personalize Runtime pour effectuer cette tâche.
6. Ajoute les données de recommandation appropriées à un attribut recommandé dynamique (`RecommendationItems`) pour chaque destinataire de message.
7. Invoque votre fonction Lambda et envoie les données de recommandation pour chaque destinataire à cette fonction afin qu'elles soient traitées.

   Les données sont envoyées sous la forme d'un objet JSON qui contient la définition de point de terminaison pour chaque destinataire. Chaque définition de point de terminaison inclut un champ `RecommendationItems` contenant un tableau ordonné de 1 à 5 valeurs. Le nombre de valeurs dans le tableau dépend des paramètres de configuration du modèle.
8. Attend que votre fonction Lambda traite les données et renvoie les résultats.

   Les résultats prennent la forme d'un objet JSON qui contient une définition de point de terminaison mise à jour pour chaque destinataire. Chaque définition de point de terminaison mise à jour contient un nouvel objet `Recommendations`. Cet objet contient 1 à 10 champs, un pour chaque attribut recommandé personnalisé que vous avez défini dans les paramètres de configuration du modèle. Chacun de ces champs stocke des données de recommandation améliorées pour le point de terminaison.
9. Utilise la définition de point de terminaison mise à jour pour chaque destinataire afin de remplacer chaque variable de message par la valeur appropriée pour ce destinataire.
10. Il envoie une version du message qui contient les recommandations personnalisées pour chaque destinataire du message.

Pour personnaliser et améliorer les recommandations de cette manière, commencez par créer une fonction Lambda qui traite les définitions de point de terminaison envoyées par Amazon Pinpoint, et renvoie les définitions de point de terminaison mises à jour. Ensuite, affectez une stratégie de fonction Lambda à la fonction et autorisez Amazon Pinpoint à invoquer la fonction. Configurez ensuite le modèle de recommandation dans Amazon Pinpoint. Lorsque vous configurez le modèle, spécifiez la fonction à appeler et définissez les attributs recommandés à utiliser.

Création d'une fonction Lambda qu'Amazon Pinpoint appellera pour un modèle de recommandation

Pour savoir comment créer une fonction Lambda, consultez Démarrage dans le Guide du développeur AWS Lambda . Lorsque vous concevez et développez votre fonction, gardez à l'esprit les exigences et consignes suivantes.

Données d'événement en entrée

Lorsqu'Amazon Pinpoint invoque une fonction Lambda pour un modèle de recommandation, il envoie une charge utile qui contient la configuration et d'autres paramètres pour la campagne ou le parcours qui envoie le message. La charge utile inclut un `Endpoints` objet, qui est une carte qui associe le point de terminaison aux définitions de point de IDs terminaison pour les destinataires des messages.

Les définitions de point de terminaison utilisent la structure définie par la ressource Endpoint de l'API Amazon Pinpoint. Cependant, elles incluent également un champ pour un attribut recommandé dynamique nommé `RecommendationItems`. Le champ `RecommendationItems` contient un ou plusieurs éléments recommandés pour le point de terminaison, tels que renvoyés par la campagne Amazon Personalize. La valeur de ce champ est un tableau ordonné de 1 à 5 éléments recommandés (sous forme de chaînes). Le nombre d'éléments dans le tableau dépend du nombre d'éléments recommandés que vous avez configuré dans Amazon Pinpoint afin qu'ils soient récupérés pour chaque point de terminaison ou utilisateur.

Par exemple :

```
"Endpoints": {
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
```

```
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z"
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
```

```
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z"
    }
}
```

Dans l'exemple précédent, les paramètres Amazon Pinpoint pertinents sont :

- Le modèle de recommandation est configuré pour récupérer trois éléments recommandés pour chaque point de terminaison ou utilisateur. (La valeur de la propriété `RecommendationsPerMessage` est définie sur `3`.) Avec ce paramètre, Amazon Pinpoint récupère et ajoute uniquement les trois premiers éléments recommandés pour chaque point de terminaison ou utilisateur.
- Le projet est configuré pour utiliser des attributs utilisateur personnalisés qui stockent le prénom, le nom de famille et le quartier de chaque utilisateur. (L'objet `UserAttributes` contient les valeurs de ces attributs.)
- Le projet est configuré pour utiliser un attribut de point de terminaison personnalisé (`AddressType`) qui indique si le point de terminaison est l'adresse préférée (canal) de l'utilisateur pour recevoir des messages du projet. (L'objet `Attributes` contient la valeur de cet attribut.)

Lorsqu'Amazon Pinpoint appelle la fonction Lambda et envoie cette charge utile en tant que données d'événement, AWS Lambda transmet les données à la fonction Lambda afin qu'elles soient traitées.

Chaque charge utile peut contenir des données pour un maximum de 50 points de terminaison. Si un segment contient plus de 50 points de terminaison, Amazon Pinpoint appelle la fonction à plusieurs reprises, pour un maximum de 50 points de terminaison à la fois, jusqu'à ce que la fonction traite toutes les données.

Données et exigences de réponse

Lorsque vous concevez et développez votre fonction Lambda, gardez à l'esprit les quotas pour les modèles de machine learning. Si la fonction ne répond pas aux conditions définies par ces quotas, Amazon Pinpoint ne sera pas en mesure de traiter et d'envoyer le message.

Gardez également à l'esprit les exigences suivantes :

- La fonction doit renvoyer les définitions de point de terminaison mises à jour dans le même format que celui fourni par les données d'événement en entrée.
- Chaque définition de point de terminaison mise à jour peut contenir 1 à 10 attributs personnalisés recommandés pour le point de terminaison ou l'utilisateur. Les noms de ces attributs doivent correspondre aux noms d'attributs que vous spécifiez lorsque vous configurez le modèle de recommandation dans Amazon Pinpoint.
- Tous les attributs recommandés personnalisés doivent être renvoyés dans un seul objet `Recommendations` pour chaque point de terminaison ou utilisateur. Cette exigence permet de s'assurer qu'aucun conflit de dénomination ne survient. Vous pouvez ajouter l'objet `Recommendations` à n'importe quel emplacement dans une définition de point de terminaison.
- La valeur de chaque attribut recommandé personnalisé doit être une chaîne (valeur unique) ou un tableau de chaînes (valeurs multiples). Si la valeur est un tableau de chaînes, nous vous recommandons de préserver l'ordre des éléments recommandés renvoyés par Amazon Personalize, comme indiqué dans le champ `RecommendationItems`. Sinon, votre contenu risque de ne pas refléter les prédictions du modèle pour un point de terminaison ou un utilisateur.
- La fonction ne doit pas modifier d'autres éléments dans les données d'événement, y compris d'autres valeurs d'attribut pour un point de terminaison ou un utilisateur. Il ne doit ajouter et renvoyer des valeurs que pour les attributs recommandés personnalisés. Amazon Pinpoint n'acceptera aucune mise à jour d'autres valeurs dans la réponse de la fonction.
- La fonction doit être hébergée dans la même AWS région que le projet Amazon Pinpoint qui appelle la fonction. Si la fonction et le projet ne se trouvent pas dans la même région, Amazon Pinpoint ne peut pas envoyer de données d'événement à la fonction.

Si l'une des exigences précédentes n'est pas satisfaite, Amazon Pinpoint ne sera pas en mesure de traiter et d'envoyer le message à un ou plusieurs points de terminaison. Cela peut entraîner l'échec d'une activité de campagne ou de parcours.

Enfin, nous vous recommandons de réserver 256 exécutions simultanées pour la fonction.

Dans l'ensemble, votre fonction Lambda doit traiter les données d'événement envoyées par Amazon Pinpoint et renvoyer les définitions de point de terminaison modifiées. Pour cela, elle peut itérer chaque point de terminaison de l'objet `Endpoints` et, pour chaque point de terminaison, créer et définir des valeurs pour les attributs personnalisés recommandés que vous souhaitez utiliser. L'exemple de gestionnaire suivant, écrit en Python et qui poursuit l'exemple précédent de données d'événement d'entrée, montre ceci :

```
import json
import string

def lambda_handler(event, context):
    print("Received event: " + json.dumps(event))
    print("Received context: " +  str(context))
    segment_endpoints = event["Endpoints"]
    new_segment = dict()
    for endpoint_id in segment_endpoints.keys():
        endpoint = segment_endpoints[endpoint_id]
        if supported_endpoint(endpoint):
            new_segment[endpoint_id] = add_recommendation(endpoint)

    print("Returning endpoints: " + json.dumps(new_segment))
    return new_segment

def supported_endpoint(endpoint):
    return True

def add_recommendation(endpoint):
    endpoint["Recommendations"] = dict()

    customTitleList = list()
    customGenreList = list()
    for i,item in enumerate(endpoint["RecommendationItems"]):
        item = int(item)
        if item == 1210:
            customTitleList.insert(i, "Hanna")
            customGenreList.insert(i, "Action")
        elif item == 1527:
            customTitleList.insert(i, "Catastrophe")
            customGenreList.insert(i, "Comedy")
        elif item == 1815:
            customTitleList.insert(i, "Fleabag")
            customGenreList.insert(i, "Comedy")
        elif item == 2009:
            customTitleList.insert(i, "Late Night")
            customGenreList.insert(i, "Drama")
        elif item == 4582:
            customTitleList.insert(i, "Agatha Christie\'s The ABC Murders")
            customGenreList.insert(i, "Crime")
        elif item == 6542:
            customTitleList.insert(i, "Hunters")
```

```
            customGenreList.insert(i, "Drama")

    endpoint["Recommendations"]["Title"] = customTitleList
    endpoint["Recommendations"]["Genre"] = customGenreList

    return endpoint
```

Dans l'exemple précédent, AWS Lambda transmet les données de l'événement au gestionnaire en tant que `event` paramètre. Le gestionnaire itère chaque point de terminaison de l'objet `Endpoints` et définit les valeurs des attributs recommandés personnalisés nommés `Recommendations.Title` et `Recommendations.Genre`. L'instruction `return` renvoie chaque définition de point de terminaison mise à jour à Amazon Pinpoint.

Pour poursuivre l'exemple précédent de données d'événement en entrée, les définitions de point de terminaison mises à jour sont les suivantes :

```
"Endpoints":{
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
```

```
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z",
        "Recommendations":{
            "Title":[
                "Fleabag",
                "Late Night",
                "Catastrophe"
            ],
            "Genre":[
                "Comedy",
                "Comedy",
                "Comedy"
            ]
        }
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
```

```
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z",
        "Recommendations":{
            "Title":[
                "Hanna",
                "Hunters",
                "Agatha Christie\'s The ABC Murders"
            ],
            "Genre":[
                "Action",
                "Drama",
                "Crime"
            ]
        }
    }
}
```

Dans l'exemple précédent, la fonction a modifié l'objet `Endpoints` reçu et a renvoyé les résultats. L'objet `Endpoint` de chaque point de terminaison contient désormais un nouvel objet `Recommendations`, qui contient des champs `Title` et `Genre`. Chacun de ces champs stocke un tableau ordonné de trois valeurs (sous forme de chaînes), où chaque valeur fournit un contenu amélioré pour un élément recommandé correspondant dans le champ `RecommendationItems`.

Attribuez une politique de fonction Lambda pour autoriser Amazon Pinpoint à traiter les données de recommandation

Avant de pouvoir utiliser votre fonction Lambda pour traiter les données de recommandation, vous devez autoriser Amazon Pinpoint à invoquer la fonction. Pour accorder l'autorisation d'invocation, attribuez une stratégie de fonction Lambda à la fonction. Une stratégie de fonction Lambda est une stratégie d'autorisations basée sur les ressources qui désigne les entités qui peuvent utiliser une fonction et les actions que ces entités peuvent effectuer. Pour plus d'informations, consultez Utilisation des stratégies fondées sur les ressources pour AWS Lambda dans le Guide du développeur AWS Lambda .

L'exemple de politique suivant permet au responsable du service Amazon Pinpoint d'utiliser l'`lambda:InvokeFunction`action pour une campagne Amazon Pinpoint spécifique *`campaignId`* () dans un projet Amazon Pinpoint particulier () : *`projectId`*

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

La stratégie de fonction exige un bloc `Condition` qui inclut une clé `AWS:SourceArn`. Cette clé spécifie quelle ressource est autorisée à appeler la fonction. Dans l'exemple précédent, la stratégie permet à une campagne spécifique d'appeler la fonction.

Vous pouvez également rédiger une politique qui autorise le responsable du service Amazon Pinpoint à utiliser l'`lambda:InvokeFunction`action pour toutes les campagnes et tous les parcours d'un projet Amazon Pinpoint spécifique (). *`projectId`* L'exemple de stratégie ci-dessous montre ceci :

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

Contrairement au premier exemple, la clé `AWS:SourceArn` dans le bloc `Condition` de cet exemple permet à un projet particulier d'appeler la fonction. Cette autorisation s'applique à toutes les campagnes et à tous les parcours du projet.

Pour écrire une stratégie plus générique, vous pouvez utiliser des caractères génériques (*) correspondant à plusieurs caractères. Par exemple, vous pouvez utiliser le bloc `Condition` suivant pour autoriser n'importe quel projet Amazon Pinpoint à invoquer la fonction :

```
"Condition": {
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}
```

Si vous souhaitez utiliser la fonction Lambda avec tous les projets de votre compte Amazon Pinpoint, nous vous recommandons de configurer le bloc `Condition` de la stratégie comme indiqué précédemment. Toutefois, la bonne pratique consiste à créer des stratégies qui incluent uniquement les autorisations requises pour effectuer une action spécifique sur une ressource spécifique.

Autoriser Amazon Pinpoint à appeler une fonction Lambda à l'aide de la commande et de la commande Lambda add AWS CLI - permission

Après avoir attribué une politique de fonction Lambda à une fonction, vous pouvez ajouter des autorisations qui permettent à Amazon Pinpoint d'invoquer la fonction pour un projet, une campagne ou un parcours spécifique. Vous pouvez le faire à l'aide de la commande AWS Command Line Interface (AWS CLI) et de la commande Lambda `add-permission`. L'exemple suivant montre comment procéder pour un projet spécifique (*`projectId`*) :

```
$ aws lambda add-permission \
--function-name function-name \
--statement-id sid \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*
```

L'exemple précédent est mis en forme pour Unix, Linux et macOS. Sous Microsoft Windows, remplacez le caractère de continuité de ligne (\) par le caret (^).

Si la commande s'exécute correctement, vous obtenez une sortie similaire à ce qui suit :

```
{
```

```
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:recommenders/*\"}}}"
}
```

La valeur `Statement` est une version de la chaîne JSON correspondant à l'instruction ajoutée à la politique de la fonction Lambda.

Configurer Amazon Pinpoint pour appeler la fonction Lambda pour un modèle de recommandation

Pour configurer Amazon Pinpoint afin qu'il invoque la fonction Lambda d'un modèle de recommandation, spécifiez les paramètres de configuration spécifiques à Lambda suivants pour le modèle :

- `RecommendationTransformerUri` : cette propriété spécifie le nom ou l'Amazon Resource Name (ARN) de la fonction Lambda.
- `Attributes` : cet objet est une carte qui définit les attributs recommandés personnalisés que la fonction ajoute à chaque définition de point de terminaison. Chacun de ces attributs peut être utilisé comme variable de message dans un modèle de message.

Vous pouvez spécifier ces paramètres à l'aide de la ressource Modèles de recommandation de l'API Amazon Pinpoint (lorsque vous créez la configuration d'un modèle) ou de la ressource Modèle de recommandation de l'API Amazon Pinpoint (si vous mettez à jour la configuration d'un modèle). Vous pouvez également définir ces paramètres à l'aide de la console Amazon Pinpoint.

Pour plus d'informations sur l'utilisation des modèles de recommandation avec Amazon Pinpoint, consultez Modèles de machine learning dans le Guide de l'utilisateur d'Amazon Pinpoint.

Supprimez votre projet Amazon Pinpoint et supprimez les données personnelles sensibles

En fonction de la façon dont vous l'utilisez, Amazon Pinpoint peut stocker certaines données pouvant être considérées comme des renseignements personnels. Par exemple, un point de terminaison dans Amazon Pinpoint contient les informations de contact pour un utilisateur final, telles que son adresse e-mail ou son numéro de téléphone portable.

Vous pouvez utiliser la console ou l'API Amazon Pinpoint pour supprimer définitivement ces données personnelles. Cette rubrique comprend des procédures pour supprimer différents types de données pouvant être considérées comme des renseignements personnels.

Vous pouvez également fermer complètement votre AWS compte. Pour plus d'informations, consultez la section Clôture d'un AWS compte dans le Guide de Gestion de compte AWS référence.

Supprimer toutes les données du projet Amazon Pinpoint

Il est possible de supprimer définitivement toutes les données que vous avez stockées pour un projet Amazon Pinpoint. Pour cela, vous pouvez supprimer le projet.

> **Warning**
>
> Si vous supprimez un projet, Amazon Pinpoint supprime tous les paramètres et données spécifiques au projet. Ces informations ne peuvent pas être récupérées.

Lorsque vous supprimez un projet, Amazon Pinpoint supprime tous les paramètres spécifiques au projet pour la notification push et les canaux de messagerie SMS bidirectionnels. Sont également supprimés tous les segments, campagnes, parcours et données analytiques spécifiques au projet qui sont stockés dans Amazon Pinpoint, comme les éléments suivants :

- Segments : tous les paramètres et données des segments. Pour les segments dynamiques, cela inclut les groupes et les filtres de segments que vous avez définis. Pour les segments importés, cela inclut les points de terminaison, les données utilisateur IDs et les autres données que vous avez importées, ainsi que les filtres que vous avez appliqués.
- Campagnes : tous les messages, les traitements de message et les variables, les données d'analyse, les programmes et autres paramètres.

- Parcours : toutes les activités, données analytiques, programmes et autres paramètres.
- Analyses : données pour toutes les métriques d'engagement, par exemple, le nombre de messages envoyés et remis pour les campagnes et les parcours, ainsi que toutes les métriques d'exécution de parcours. Pour les applications mobiles et Web, toutes les données d'événements qui n'ont pas été diffusées vers un autre AWS service tel qu'Amazon Kinesis, tous les entonnoirs, ainsi que les données relatives à l'utilisation des applications, aux revenus et aux statistiques démographiques. Avant de supprimer un projet, nous vous recommandons d’exporter ces données vers un autre emplacement.

Vous pouvez supprimer un projet à l'aide de la console Amazon Pinpoint. Pour en savoir plus, consultez Supprimer un projet dans le Guide de l'utilisateur Amazon Pinpoint. Vous pouvez également supprimer un projet par programme à l'aide de la ressource App de l'API Amazon Pinpoint.

Exemples de code pour Amazon Pinpoint à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Amazon Pinpoint avec un kit de développement AWS logiciel (SDK).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- Exemples de code pour Amazon Pinpoint à l'aide de AWS SDKs
  - Exemples de base pour l'utilisation d'Amazon Pinpoint AWS SDKs
    - Actions pouvant être utilisées par Amazon Pinpoint AWS SDKs
      - Utilisation CreateApp avec un AWS SDK ou une CLI
      - Utilisation CreateCampaign avec un AWS SDK
      - Utilisation CreateExportJob avec un AWS SDK
      - Utilisation CreateImportJob avec un AWS SDK
      - Utilisation CreateSegment avec un AWS SDK
      - Utilisation DeleteApp avec un AWS SDK ou une CLI
      - Utilisation DeleteEndpoint avec un AWS SDK
      - Utilisation GetEndpoint avec un AWS SDK ou une CLI
      - Utilisation GetSegments avec un AWS SDK
      - Utilisation GetSmsChannel avec un AWS SDK ou une CLI
      - Utilisation GetUserEndpoints avec un AWS SDK
      - Utilisation SendMessages avec un AWS SDK ou une CLI
      - Utilisation UpdateEndpoint avec un AWS SDK
  - Scénarios d'utilisation d'Amazon Pinpoint AWS SDKs
    - Commencer à utiliser Final User Messaging Push
- Exemples de code pour l'API SMS et voix Amazon Pinpoint à l'aide de AWS SDKs
  - Exemples de base d'utilisation de l'API SMS et Voice Amazon Pinpoint AWS SDKs

- Actions pour l'API SMS et voix Amazon Pinpoint à l'aide de AWS SDKs
  - Utilisation CreateConfigurationSet avec un AWS SDK
  - Utilisation CreateConfigurationSetEventDestination avec un AWS SDK
  - Utilisation DeleteConfigurationSet avec un AWS SDK
  - Utilisation DeleteConfigurationSetEventDestination avec un AWS SDK
  - Utilisation GetConfigurationSetEventDestinations avec un AWS SDK
  - Utilisation ListConfigurationSets avec un AWS SDK
  - Utilisation SendVoiceMessage avec un AWS SDK
  - Utilisation UpdateConfigurationSetEventDestination avec un AWS SDK

Exemples de code pour Amazon Pinpoint à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Amazon Pinpoint avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Amazon Pinpoint

- Exemples de base pour l'utilisation d'Amazon Pinpoint AWS SDKs
  - Actions pouvant être utilisées par Amazon Pinpoint AWS SDKs
    - Utilisation CreateApp avec un AWS SDK ou une CLI
    - Utilisation CreateCampaign avec un AWS SDK
    - Utilisation CreateExportJob avec un AWS SDK
    - Utilisation CreateImportJob avec un AWS SDK
    - Utilisation CreateSegment avec un AWS SDK

    - Utilisation DeleteApp avec un AWS SDK ou une CLI
    - Utilisation DeleteEndpoint avec un AWS SDK
    - Utilisation GetEndpoint avec un AWS SDK ou une CLI
    - Utilisation GetSegments avec un AWS SDK
    - Utilisation GetSmsChannel avec un AWS SDK ou une CLI
    - Utilisation GetUserEndpoints avec un AWS SDK
    - Utilisation SendMessages avec un AWS SDK ou une CLI
    - Utilisation UpdateEndpoint avec un AWS SDK
- Scénarios d'utilisation d'Amazon Pinpoint AWS SDKs
    - Commencer à utiliser Final User Messaging Push

Exemples de base pour l'utilisation d'Amazon Pinpoint AWS SDKs

Les exemples de code suivants montrent comment utiliser les bases d'Amazon Pinpoint avec. AWS SDKs

Exemples

- Actions pouvant être utilisées par Amazon Pinpoint AWS SDKs
    - Utilisation CreateApp avec un AWS SDK ou une CLI
    - Utilisation CreateCampaign avec un AWS SDK
    - Utilisation CreateExportJob avec un AWS SDK
    - Utilisation CreateImportJob avec un AWS SDK
    - Utilisation CreateSegment avec un AWS SDK
    - Utilisation DeleteApp avec un AWS SDK ou une CLI
    - Utilisation DeleteEndpoint avec un AWS SDK
    - Utilisation GetEndpoint avec un AWS SDK ou une CLI
    - Utilisation GetSegments avec un AWS SDK
    - Utilisation GetSmsChannel avec un AWS SDK ou une CLI
    - Utilisation GetUserEndpoints avec un AWS SDK
    - Utilisation SendMessages avec un AWS SDK ou une CLI
    - Utilisation UpdateEndpoint avec un AWS SDK

Actions pouvant être utilisées par Amazon Pinpoint AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions Amazon Pinpoint individuelles avec. AWS SDKs Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Ces extraits appellent l'API Amazon Pinpoint et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. Vous pouvez voir les actions dans leur contexte dans Scénarios d'utilisation d'Amazon Pinpoint AWS SDKs.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez la référence de l'API Amazon Pinpoint.

Exemples

- Utilisation CreateApp avec un AWS SDK ou une CLI
- Utilisation CreateCampaign avec un AWS SDK
- Utilisation CreateExportJob avec un AWS SDK
- Utilisation CreateImportJob avec un AWS SDK
- Utilisation CreateSegment avec un AWS SDK
- Utilisation DeleteApp avec un AWS SDK ou une CLI
- Utilisation DeleteEndpoint avec un AWS SDK
- Utilisation GetEndpoint avec un AWS SDK ou une CLI
- Utilisation GetSegments avec un AWS SDK
- Utilisation GetSmsChannel avec un AWS SDK ou une CLI
- Utilisation GetUserEndpoints avec un AWS SDK
- Utilisation SendMessages avec un AWS SDK ou une CLI
- Utilisation UpdateEndpoint avec un AWS SDK

Utilisation **`CreateApp`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `CreateApp`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Commencer à utiliser Final User Messaging Push

CLI

AWS CLI

Exemple 1 : pour créer une application

L'exemple `create-app` suivant crée une application (projet).

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

Sortie :

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

Exemple 2 : pour créer une application balisée

L'exemple `create-app` suivant crée une application (projet) et associe une balise (clé et valeur) à l'application.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

Sortie :

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
```

```
            "Stack": "Test"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateAppà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateAppRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateAppResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateApplicationRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateApp {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appName>

                Where:
```

```
                  appName - The name of the application to create.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
        String appName = args[0];
        System.out.println("Creating an application with name: " + appName);

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String appID = createApplication(pinpoint, appName);
        System.out.println("App ID is: " + appID);
        pinpoint.close();
    }

    public static String createApplication(PinpointClient pinpoint, String
 appName) {
        try {
            CreateApplicationRequest appRequest =
 CreateApplicationRequest.builder()
                    .name(appName)
                    .build();

            CreateAppRequest request = CreateAppRequest.builder()
                    .createApplicationRequest(appRequest)
                    .build();

            CreateAppResponse result = pinpoint.createApp(request);
            return result.applicationResponse().id();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateAppà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun createApplication(applicationName: String?): String? {
    val createApplicationRequestOb =
        CreateApplicationRequest {
            name = applicationName
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.createApp(
                CreateAppRequest {
                    createApplicationRequest = createApplicationRequestOb
                },
            )
        return result.applicationResponse?.id
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateAppà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateCampaign`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateCampaign`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créer une campagne.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = """

                Usage:   <appId> <segmentId>
```

```
                Where:
                  appId - The ID of the application to create the campaign in.
                  segmentId - The ID of the segment to create the campaign from.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId,
String segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
    }

    public static CampaignResponse createCampaign(PinpointClient client, String
appID, String segmentID) {

        try {
            Schedule schedule = Schedule.builder()
                    .startTime("IMMEDIATE")
                    .build();

            Message defaultMessage = Message.builder()
                    .action(Action.OPEN_APP)
                    .body("My message body.")
                    .title("My message title.")
                    .build();

            MessageConfiguration messageConfiguration =
MessageConfiguration.builder()
```

```
                    .defaultMessage(defaultMessage)
                    .build();

            WriteCampaignRequest request = WriteCampaignRequest.builder()
                    .description("My description")
                    .schedule(schedule)
                    .name("MyCampaign")
                    .segmentId(segmentID)
                    .messageConfiguration(messageConfiguration)
                    .build();

            CreateCampaignResponse result =
 client.createCampaign(CreateCampaignRequest.builder()
                    .applicationId(appID)
                    .writeCampaignRequest(request).build());

            System.out.println("Campaign ID: " + result.campaignResponse().id());
            return result.campaignResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return null;
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateCampaignà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l’exemple complet et découvrez comment le configurer et l’exécuter dans le référentiel d’exemples de code AWS.

```
suspend fun createPinCampaign(
    appId: String,
    segmentIdVal: String,
) {
    val scheduleOb =
        Schedule {
            startTime = "IMMEDIATE"
        }

    val defaultMessageOb =
        Message {
            action = Action.OpenApp
            body = "My message body"
            title = "My message title"
        }

    val messageConfigurationOb =
        MessageConfiguration {
            defaultMessage = defaultMessageOb
        }

    val writeCampaign =
        WriteCampaignRequest {
            description = "My description"
            schedule = scheduleOb
            name = "MyCampaign"
            segmentId = segmentIdVal
            messageConfiguration = messageConfigurationOb
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result: CreateCampaignResponse =
            pinpoint.createCampaign(
                CreateCampaignRequest {
                    applicationId = appId
                    writeCampaignRequest = writeCampaign
                },
            )
        println("Campaign ID is ${result.campaignResponse?.id}")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateCampaignà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateExportJob`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`CreateExportJob`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Exporter un point de terminaison.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
```

```
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-
export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = """

                This program performs the following steps:

                1. Exports the endpoints to an Amazon S3 bucket.
                2. Downloads the exported endpoints files from Amazon S3.
                3. Parses the endpoints files to obtain the endpoint IDs and
 prints them.
                Usage: ExportEndpoints <applicationId> <s3BucketName>
 <iamExportRoleArn> <path>

                Where:
                  applicationId - The ID of the Amazon Pinpoint application that
 has the endpoint.
                  s3BucketName - The name of the Amazon S3 bucket to export the
 JSON file to.\s
                  iamExportRoleArn - The ARN of an IAM role that grants Amazon
 Pinpoint write permissions to the S3 bucket.  path - The path where the files
 downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
```

```
                """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
        String s3BucketName = args[1];
        String iamExportRoleArn = args[2];
        String path = args[3];
        System.out.println("Deleting an application with ID: " + applicationId);

        Region region = Region.US_EAST_1;
        PinpointClient pinpoint = PinpointClient.builder()
                .region(region)
                .build();

        S3Client s3Client = S3Client.builder()
                .region(region)
                .build();

        exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
        pinpoint.close();
        s3Client.close();
    }

    public static void exportAllEndpoints(PinpointClient pinpoint,
            S3Client s3Client,
            String applicationId,
            String s3BucketName,
            String path,
            String iamExportRoleArn) {

        try {
            List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
s3BucketName, iamExportRoleArn,
                    applicationId);
            List<String> endpointFileKeys = objectKeys.stream().filter(o ->
o.endsWith(".gz"))
                    .collect(Collectors.toList());
            downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);
```

```
        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static List<String> exportEndpointsToS3(PinpointClient pinpoint,
 S3Client s3Client, String s3BucketName,
            String iamExportRoleArn, String applicationId) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
        String endpointsKeyPrefix = "exports/" + applicationId + "_" +
 dateFormat.format(new Date());
        String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
 "/";
        List<String> objectKeys = new ArrayList<>();
        String key;

        try {
            // Defines the export job that Amazon Pinpoint runs.
            ExportJobRequest jobRequest = ExportJobRequest.builder()
                    .roleArn(iamExportRoleArn)
                    .s3UrlPrefix(s3UrlPrefix)
                    .build();

            CreateExportJobRequest exportJobRequest =
 CreateExportJobRequest.builder()
                    .applicationId(applicationId)
                    .exportJobRequest(jobRequest)
                    .build();

            System.out.format("Exporting endpoints from Amazon Pinpoint
 application %s to Amazon S3 " +
                    "bucket %s . . .\n", applicationId, s3BucketName);

            CreateExportJobResponse exportResult =
 pinpoint.createExportJob(exportJobRequest);
            String jobId = exportResult.exportJobResponse().id();
            System.out.println(jobId);
            printExportJobStatus(pinpoint, applicationId, jobId);

            ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                    .bucket(s3BucketName)
```

```
                    .prefix(endpointsKeyPrefix)
                    .build();

            // Create a list of object keys.
            ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
            List<S3Object> objects = v2Response.contents();
            for (S3Object object : objects) {
                key = object.key();
                objectKeys.add(key);
            }

            return objectKeys;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void printExportJobStatus(PinpointClient pinpointClient,
            String applicationId,
            String jobId) {

        GetExportJobResponse getExportJobResult;
        String status;

        try {
            // Checks the job status until the job completes or fails.
            GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
                    .jobId(jobId)
                    .applicationId(applicationId)
                    .build();

            do {
                getExportJobResult =
pinpointClient.getExportJob(exportJobRequest);
                status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
                System.out.format("Export job %s . . .\n", status);
                TimeUnit.SECONDS.sleep(3);

            } while (!status.equals("COMPLETED") && !status.equals("FAILED"));
```

```
            if (status.equals("COMPLETED")) {
                System.out.println("Finished exporting endpoints.");
            } else {
                System.err.println("Failed to export endpoints.");
                System.exit(1);
            }

        } catch (PinpointException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Download files from an Amazon S3 bucket and write them to the path
location.
    public static void downloadFromS3(S3Client s3Client, String path, String
s3BucketName, List<String> objectKeys) {

        String newPath;
        try {
            for (String key : objectKeys) {
                GetObjectRequest objectRequest = GetObjectRequest.builder()
                        .bucket(s3BucketName)
                        .key(key)
                        .build();

                ResponseBytes<GetObjectResponse> objectBytes =
s3Client.getObjectAsBytes(objectRequest);
                byte[] data = objectBytes.asByteArray();

                // Write the data to a local file.
                String fileSuffix = new
SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
                newPath = path + fileSuffix + ".gz";
                File myFile = new File(newPath);
                OutputStream os = new FileOutputStream(myFile);
                os.write(data);
            }
            System.out.println("Download finished.");

        } catch (S3Exception | NullPointerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
```

```
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateExportJobà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateImportJob`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`CreateImportJob`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Importer un segment.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <bucket> <key> <roleArn>\s

                Where:
                  appId - The application ID to create a segment for.
                  bucket - The name of the Amazon S3 bucket that contains the
 segment definitons.
                  key - The key of the S3 object.
                  roleArn - ARN of the role that allows Amazon
 Pinpoint to access S3. You need to set trust management for this
 to work. See https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_policies_elements_principal.html
                  """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket,
 key, roleArn);
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for
 import job status.");
        System.out.println("See application " + response.jobStatus() + " for
 import job status.");
        pinpoint.close();
    }
```

```
    public static ImportJobResponse createImportSegment(PinpointClient client,
            String appId,
            String bucket,
            String key,
            String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                    .defineSegment(true)
                    .registerEndpoints(true)
                    .roleArn(roleArn)
                    .format(Format.JSON)
                    .s3Url("s3://" + bucket + "/" + key)
                    .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                    .importJobRequest(importRequest)
                    .applicationId(appId)
                    .build();

            CreateImportJobResponse jobResponse =
 client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateImportJobà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateSegment`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `CreateSegment`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateSegment {
        public static void main(String[] args) {
                final String usage = """
```

```
                                Usage:   <appId>

                                Where:
                                  appId - The application ID to create a segment
for.

                                """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String appId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                SegmentResponse result = createSegment(pinpoint, appId);
                System.out.println("Segment " + result.name() + " created.");
                System.out.println(result.segmentType());
                pinpoint.close();
        }

        public static SegmentResponse createSegment(PinpointClient client, String
appId) {
                try {
                        Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();
                        segmentAttributes.put("Team",
AttributeDimension.builder()
                                        .attributeType(AttributeType.INCLUSIVE)
                                        .values("Lakers")
                                        .build());

                        RecencyDimension recencyDimension =
RecencyDimension.builder()
                                        .duration("DAY_30")
                                        .recencyType("ACTIVE")
                                        .build();

                        SegmentBehaviors segmentBehaviors =
SegmentBehaviors.builder()
```

```
                                        .recency(recencyDimension)
                                        .build();

                        SegmentDemographics segmentDemographics =
SegmentDemographics
                                        .builder()
                                        .build();

                        SegmentLocation segmentLocation = SegmentLocation
                                        .builder()
                                        .build();

                        SegmentDimensions dimensions = SegmentDimensions
                                        .builder()
                                        .attributes(segmentAttributes)
                                        .behavior(segmentBehaviors)
                                        .demographic(segmentDemographics)
                                        .location(segmentLocation)
                                        .build();

                        WriteSegmentRequest writeSegmentRequest =
WriteSegmentRequest.builder()
                                        .name("MySegment")
                                        .dimensions(dimensions)
                                        .build();

                        CreateSegmentRequest createSegmentRequest =
CreateSegmentRequest.builder()
                                        .applicationId(appId)
                                        .writeSegmentRequest(writeSegmentRequest)
                                        .build();

                        CreateSegmentResponse createSegmentResult =
client.createSegment(createSegmentRequest);
                        System.out.println("Segment ID: " +
createSegmentResult.segmentResponse().id());
                        System.out.println("Done");
                        return createSegmentResult.segmentResponse();

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
                return null;
```

```
        }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateSegmentà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun createPinpointSegment(applicationIdVal: String?): String? {
    val segmentAttributes = mutableMapOf<String, AttributeDimension>()
    val myList = mutableListOf<String>()
    myList.add("Lakers")

    val atts =
        AttributeDimension {
            attributeType = AttributeType.Inclusive
            values = myList
        }

    segmentAttributes["Team"] = atts
    val recencyDimension =
        RecencyDimension {
            duration = Duration.fromValue("DAY_30")
            recencyType = RecencyType.fromValue("ACTIVE")
        }

    val segmentBehaviors =
        SegmentBehaviors {
            recency = recencyDimension
        }

    val segmentLocation = SegmentLocation {}
```

```
    val dimensionsOb =
        SegmentDimensions {
            attributes = segmentAttributes
            behavior = segmentBehaviors
            demographic = SegmentDemographics {}
            location = segmentLocation
        }

    val writeSegmentRequestOb =
        WriteSegmentRequest {
            name = "MySegment101"
            dimensions = dimensionsOb
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val createSegmentResult: CreateSegmentResponse =
            pinpoint.createSegment(
                CreateSegmentRequest {
                    applicationId = applicationIdVal
                    writeSegmentRequest = writeSegmentRequestOb
                },
            )
        println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
        return createSegmentResult.segmentResponse?.id
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateSegmentà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DeleteApp`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DeleteApp`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Commencer à utiliser Final User Messaging Push

CLI

AWS CLI

Pour supprimer une application

L'exemple `delete-app` suivant supprime une application (projet).

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

Sortie :

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAppà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimer une application.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteAppRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteAppResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteApp {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appId>

                Where:
                 appId - The ID of the application to delete.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        System.out.println("Deleting an application with ID: " + appId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinApp(pinpoint, appId);
        System.out.println("Done");
        pinpoint.close();
    }

    public static void deletePinApp(PinpointClient pinpoint, String appId) {
```

```
        try {
            DeleteAppRequest appRequest = DeleteAppRequest.builder()
                    .applicationId(appId)
                    .build();

            DeleteAppResponse result = pinpoint.deleteApp(appRequest);
            String appName = result.applicationResponse().name();
            System.out.println("Application " + appName + " has been deleted.");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAppà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun deletePinApp(appId: String?) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.deleteApp(
                DeleteAppRequest {
                    applicationId = appId
                },
            )
        val appName = result.applicationResponse?.name
        println("Application $appName has been deleted.")
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAppà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DeleteEndpoint`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `DeleteEndpoint`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprime un point de terminaison.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteEndpoint {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appName> <endpointId >

                Where:
                  appId - The id of the application to delete.
                  endpointId - The id of the endpoint to delete.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinEncpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEncpoint(PinpointClient pinpoint, String appId,
String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id  " + id);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteEndpointà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun deletePinEncpoint(
    appIdVal: String?,
    endpointIdVal: String?,
) {
    val deleteEndpointRequest =
        DeleteEndpointRequest {
            applicationId = appIdVal
            endpointId = endpointIdVal
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.deleteEndpoint(deleteEndpointRequest)
        val id = result.endpointResponse?.id
        println("The deleted endpoint is  $id")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteEndpointà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`GetEndpoint`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `GetEndpoint`.

CLI

AWS CLI

Pour récupérer des informations sur les paramètres et les attributs d'un point de terminaison spécifique pour une application

L'exemple `get-endpoint` suivant extrait des informations sur les paramètres et les attributs d'un point de terminaison spécifique pour une application.

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

Sortie :

```
{
    "EndpointResponse": {
        "Address": "+11234567890",
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "Attributes": {},
        "ChannelType": "SMS",
        "CohortId": "63",
        "CreationDate": "2019-01-28T23:55:11.534Z",
        "EffectiveDate": "2021-08-06T00:04:51.763Z",
        "EndpointStatus": "ACTIVE",
        "Id": "testendpoint",
        "Location": {
            "Country": "USA"
        },
        "Metrics": {
            "SmsDelivered": 1.0
        },
        "OptOut": "ALL",
        "RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",
        "User": {
            "UserAttributes": {
                "Age": [
```

```
                    "24"
                ]
            },
        "UserId": "testuser"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetEndpointà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class LookUpEndpoint {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <endpoint>

                Where:
                  appId - The ID of the application to delete.
                  endpoint - The ID of the endpoint.\s
                  """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpoint = args[1];
        System.out.println("Looking up an endpoint point with ID: " + endpoint);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        lookupPinpointEndpoint(pinpoint, appId, endpoint);
        pinpoint.close();
    }

    public static void lookupPinpointEndpoint(PinpointClient pinpoint, String
appId, String endpoint) {
        try {
            GetEndpointRequest appRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpoint)
                    .build();

            GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
            EndpointResponse endResponse = result.endpointResponse();

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

            String endpointJson = gson.toJson(endResponse);
```

```
            System.out.println(endpointJson);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetEndpointà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun lookupPinpointEndpoint(
    appId: String?,
    endpoint: String?,
) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.getEndpoint(
                GetEndpointRequest {
                    applicationId = appId
                    endpointId = endpoint
                },
            )
        val endResponse = result.endpointResponse

        // Uses the Google Gson library to pretty print the endpoint JSON.
        val gson: com.google.gson.Gson =
            GsonBuilder()
```

```
                .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                .setPrettyPrinting()
                .create()

        val endpointJson: String = gson.toJson(endResponse)
        println(endpointJson)
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetEndpointà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`GetSegments`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `GetSegments`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Répertorier les segments.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import java.util.List;

/**
```

```
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListSegments {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId>

                Where:
                  appId - The ID of the application that contains a segment.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listSegs(pinpoint, appId);
        pinpoint.close();
    }

    public static void listSegs(PinpointClient pinpoint, String appId) {
        try {
            GetSegmentsRequest request = GetSegmentsRequest.builder()
                    .applicationId(appId)
                    .build();

            GetSegmentsResponse response = pinpoint.getSegments(request);
            List<SegmentResponse> segments = response.segmentsResponse().item();
            for (SegmentResponse segment : segments) {
                System.out
```

```
                        .println("Segement " + segment.id() + " " +
 segment.name() + " " + segment.lastModifiedDate());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSegmentsà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun listSegs(appId: String?) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val response =
            pinpoint.getSegments(
                GetSegmentsRequest {
                    applicationId = appId
                },
            )
        response.segmentsResponse?.item?.forEach { segment ->
            println("Segement id is ${segment.id}")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSegmentsà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`GetSmsChannel`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `GetSmsChannel`.

CLI

AWS CLI

Pour récupérer des informations sur le statut et les paramètres du canal SMS pour une application

L'exemple `get-sms-channel` suivant récupère le statut et les paramètres du canal SMS pour une application.

```
aws pinpoint get-sms-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Sortie :

```
{
    "SMSChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:39:18.511Z",
        "Enabled": true,
        "Id": "sms",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:39:18.511Z",
        "Platform": "SMS",
        "PromotionalMessagesPerSecond": 20,
        "TransactionalMessagesPerSecond": 20,
        "Version": 1
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSmsChannelà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelResponse;
import software.amazon.awssdk.services.pinpoint.model.GetSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateChannel {
    public static void main(String[] args) {
        final String usage = """

                Usage: CreateChannel <appId>

                Where:
                  appId - The name of the application whose channel is updated.

                """;
```

```
        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        SMSChannelResponse getResponse = getSMSChannel(pinpoint, appId);
        toggleSmsChannel(pinpoint, appId, getResponse);
        pinpoint.close();
    }

    private static SMSChannelResponse getSMSChannel(PinpointClient client, String
appId) {
        try {
            GetSmsChannelRequest request = GetSmsChannelRequest.builder()
                    .applicationId(appId)
                    .build();

            SMSChannelResponse response =
client.getSmsChannel(request).smsChannelResponse();
            System.out.println("Channel state is " + response.enabled());
            return response;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void toggleSmsChannel(PinpointClient client, String appId,
SMSChannelResponse getResponse) {
        boolean enabled = !getResponse.enabled();
        try {
            SMSChannelRequest request = SMSChannelRequest.builder()
                    .enabled(enabled)
                    .build();
```

```
            UpdateSmsChannelRequest updateRequest =
 UpdateSmsChannelRequest.builder()
                    .smsChannelRequest(request)
                    .applicationId(appId)
                    .build();

            UpdateSmsChannelResponse result =
 client.updateSmsChannel(updateRequest);
            System.out.println("Channel state: " +
 result.smsChannelResponse().enabled());

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetSmsChannelà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`GetUserEndpoints`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`GetUserEndpoints`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListEndpointIds {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <applicationId> <userId>

                Where:
                   applicationId - The ID of the Amazon Pinpoint application that
 has the endpoint.
                   userId - The user id applicable to the endpoints""";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
        String userId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllEndpoints(pinpoint, applicationId, userId);
        pinpoint.close();
    }

    public static void listAllEndpoints(PinpointClient pinpoint,
            String applicationId,
```

```
            String userId) {

        try {
            GetUserEndpointsRequest endpointsRequest =
 GetUserEndpointsRequest.builder()
                    .userId(userId)
                    .applicationId(applicationId)
                    .build();

            GetUserEndpointsResponse response =
 pinpoint.getUserEndpoints(endpointsRequest);
            List<EndpointResponse> endpoints =
 response.endpointsResponse().item();

            // Display the results.
            for (EndpointResponse endpoint : endpoints) {
                System.out.println("The channel type is: " +
 endpoint.channelType());
                System.out.println("The address is  " + endpoint.address());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetUserEndpointsà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`SendMessages`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `SendMessages`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Commencer à utiliser Final User Messaging Push

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Envoyer un e-mail.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load test settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally load local settings.
        .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";
```

```
       // The "From" address. This address has to be verified in Amazon
Pinpoint
       // in the region you're using to send email.
       string senderAddress = configuration["SenderAddress"]!;

       // The address on the "To" line. If your Amazon Pinpoint account is in
       // the sandbox, this address also has to be verified.
       string toAddress = configuration["ToAddress"]!;

       // The Amazon Pinpoint project/application ID to use when you send this
message.
       // Make sure that the SMS channel is enabled for the project or
application
       // that you choose.
       string appId = configuration["AppId"]!;

       try
       {
           await SendEmailMessage(region, appId, toAddress, senderAddress);
       }
       catch (Exception ex)
       {
           Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
       }
   }

   public static async Task<MessageResponse> SendEmailMessage(
       string region, string appId, string toAddress, string senderAddress)
   {
       var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

       // The subject line of the email.
       string subject = "Amazon Pinpoint Email test";

       // The body of the email for recipients whose email clients don't
       // support HTML content.
       string textBody = @"Amazon Pinpoint Email Test (.NET)"
                         + "\n---------------------------------"
                         + "\nThis email was sent using the Amazon Pinpoint API
using the AWS SDK for .NET.";
```

```
        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK
 for .NET)</h1>"
                          + "\n  <p>This email was sent using the "
                          + "\n    <a href='https://aws.amazon.com/
pinpoint/'>Amazon Pinpoint</a> API "
                          + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                          + "\n  </p>"
                          + "\n</body>"
                          + "\n</html>";

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        string charset = "UTF-8";

        var sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses = new Dictionary<string, AddressConfiguration>
                {
                    {
                        toAddress,
                        new AddressConfiguration
                        {
                            ChannelType = ChannelType.EMAIL
                        }
                    }
                },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    EmailMessage = new EmailMessage
                    {
                        FromAddress = senderAddress,
                        SimpleEmail = new SimpleEmail
                        {
                            HtmlPart = new SimpleEmailPart
                            {
```

```
                                Charset = charset,
                                Data = htmlBody
                            },
                            TextPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = textBody
                            },
                            Subject = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = subject
                            }
                        }
                    }
                }
            }
        };
        Console.WriteLine("Sending message...");
        SendMessagesResponse response = await
 client.SendMessagesAsync(sendRequest);
        Console.WriteLine("Message sent!");
        return response.MessageResponse;
    }
}
```

Envoyer un SMS.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
```

```
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone
 number
        // or short code that you specify has to be associated with your Amazon
 Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number.  For best results, you should specify
 the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;

        // The Pinpoint project/ application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or
 application
        // that you choose.
        string appId = configuration["AppId"]!;

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;

        // The registered keyword associated with the originating short code.
        string? registeredKeyword = configuration["RegisteredKeyword"];

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        string? senderId = configuration["SenderId"];
```

```
        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
{response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
{messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";


        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
```

```
                            new AddressConfiguration { ChannelType =
 ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await
 client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

- Pour plus de détails sur l'API, reportez-vous SendMessagesà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

Pour envoyer un message SMS à l'aide du point de terminaison d'une application

L'exemple `send-messages` suivant envoie un message direct pour une application dotée d'un point de terminaison.

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

Contenu de `myfile.json` :

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Endpoints": {
        "testendpoint": {}
    }
}
```

Sortie :

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

Pour plus d'informations, consultez Canal Amazon Pinpoint SMS dans le Guide de l'utilisateur Amazon Pinpoint.

- Pour plus de détails sur l'API, reportez-vous SendMessagesà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Envoyer un e-mail.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class SendEmailMessage {

       // The character encoding the you want to use for the subject line and
       // message body of the email.
       public static String charset = "UTF-8";

   // The body of the email for recipients whose email clients support HTML
 content.
   static final String body = """
       Amazon Pinpoint test (AWS SDK for Java 2.x)

       This email was sent through the Amazon Pinpoint Email API using the AWS
 SDK for Java 2.x

       """;

       public static void main(String[] args) {
               final String usage = """

                                Usage:    <subject> <appId> <senderAddress>
 <toAddress>

           Where:
              subject - The email subject to use.
              senderAddress - The from address. This address has to be verified
 in Amazon Pinpoint in the region you're using to send email\s
              toAddress - The to address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email\s
           """;

       if (args.length != 3) {
           System.out.println(usage);
           System.exit(1);
       }

       String subject = args[0];
       String senderAddress = args[1];
       String toAddress = args[2];
       System.out.println("Sending a message");
       PinpointEmailClient pinpoint = PinpointEmailClient.builder()
           .region(Region.US_EAST_1)
           .build();

       sendEmail(pinpoint, subject, senderAddress, toAddress);
```

```
        System.out.println("Email was sent");
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
 subject, String senderAddress, String toAddress) {
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Envoyez un e-mail avec les valeurs CC.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import  software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;
import java.util.ArrayList;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendEmailMessageCC {

    // The body of the email.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS
 SDK for Java 2.x

        """;
    public static void main(String[] args) {
        final String usage = """

            Usage:    <subject> <senderAddress> <toAddress> <ccAddress>

            Where:
               subject - The email subject to use.
               senderAddress - The from address. This address has to be verified
 in Amazon Pinpoint in the region you're using to send email\s
```

```
                toAddress - The to address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
                ccAddress - The CC address.
            """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String subject = args[0];
        String senderAddress = args[1];
        String toAddress = args[2];
        String ccAddress = args[3];

        System.out.println("Sending a message");
        PinpointEmailClient pinpoint = PinpointEmailClient.builder()
            .region(Region.US_EAST_1)
            .build();

        ArrayList<String> ccList = new ArrayList<>();
        ccList.add(ccAddress);
        sendEmail(pinpoint, subject, senderAddress, toAddress, ccList);
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress, ArrayList<String> ccAddresses)
{
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
```

```
                .toAddresses(toAddress)
                .ccAddresses(ccAddresses)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            // Handle exception
            e.printStackTrace();
        }
    }
}
```

Envoyer un SMS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
```

```
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessage {

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application
 ID to use when you send this message.
                                  originationNumber - The phone number or
 short code that you specify has to be associated with your Amazon Pinpoint
 account. For best results, specify long codes in E.164 format (for example,
 +1-555-555-5654).
                                  destinationNumber - The recipient's phone
 number.  For best results, you should specify the phone number in E.164 format
 (for example, +1-555-555-5654).\s
                                  """;

                if (args.length != 4) {
                        System.out.println(usage);
```

```
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();
```

```
                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request =
 SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();

                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" +
 v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

Envoyer des SMS par lots.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessageBatch {

    // The type of SMS message that you want to send. If you plan to send
    // time-sensitive content, specify TRANSACTIONAL. If you plan to send
    // marketing-related content, specify PROMOTIONAL.
    public static String messageType = "TRANSACTIONAL";

    // The registered keyword associated with the originating short code.
    public static String registeredKeyword = "myKeyword";

    // The sender ID to use when sending the message. Support for sender ID
    // varies by country or region. For more information, see
    // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
    public static String senderId = "MySenderID";

    public static void main(String[] args) {
        final String usage = """

                Usage:   <message> <appId> <originationNumber>
 <destinationNumber> <destinationNumber1>\s

                Where:
                  message - The body of the message to send.
                  appId - The Amazon Pinpoint project/application ID to use when
 you send this message.
                  originationNumber - The phone number or short code that
 you specify has to be associated with your Amazon Pinpoint account. For best
 results, specify long codes in E.164 format (for example, +1-555-555-5654).
```

```
                destinationNumber - The recipient's phone number.  For best
results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).
                destinationNumber1 - The second recipient's phone number.  For
best results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
               """;

       if (args.length != 5) {
           System.out.println(usage);
           System.exit(1);
       }

       String message = args[0];
       String appId = args[1];
       String originationNumber = args[2];
       String destinationNumber = args[3];
       String destinationNumber1 = args[4];
       System.out.println("Sending a message");
       PinpointClient pinpoint = PinpointClient.builder()
               .region(Region.US_EAST_1)
               .build();

       sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber, destinationNumber1);
       pinpoint.close();
   }

   public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                                     String originationNumber,
                                     String destinationNumber, String
destinationNumber1) {
       try {
           Map<String, AddressConfiguration> addressMap = new HashMap<String,
AddressConfiguration>();
           AddressConfiguration addConfig = AddressConfiguration.builder()
                   .channelType(ChannelType.SMS)
                   .build();

           // Add an entry to the Map object for each number to whom you want to
send a
           // message.
           addressMap.put(destinationNumber, addConfig);
```

```
            addressMap.put(destinationNumber1, addConfig);
            SMSMessage smsMessage = SMSMessage.builder()
                    .body(message)
                    .messageType(messageType)
                    .originationNumber(originationNumber)
                    .senderId(senderId)
                    .keyword(registeredKeyword)
                    .build();

            // Create a DirectMessageConfiguration object.
            DirectMessageConfiguration direct =
 DirectMessageConfiguration.builder()
                    .smsMessage(smsMessage)
                    .build();

            MessageRequest msgReq = MessageRequest.builder()
                    .addresses(addressMap)
                    .messageConfiguration(direct)
                    .build();

            // Create a SendMessagesRequest object.
            SendMessagesRequest request = SendMessagesRequest.builder()
                    .applicationId(appId)
                    .messageRequest(msgReq)
                    .build();

            SendMessagesResponse response = pinpoint.sendMessages(request);
            MessageResponse msg1 = response.messageResponse();
            Map map1 = msg1.result();

            // Write out the result of sendMessage.
            map1.forEach((k, v) -> System.out.println((k + ":" + v)));

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous SendMessagesà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez le client dans un module séparé et exportez-le.

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
// Set the AWS Region.
const REGION = "us-east-1";
export const pinClient = new PinpointClient({ region: REGION });
```

Envoyer un e-mail.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

// The subject line of the email.
const subject = "Amazon Pinpoint Test (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
const body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https://aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
const body_html = `<html>
<head></head>
```

```
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint Email API</a>
 using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding for the subject line and message body of the email.
const charset = "UTF-8";

const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

const run = async () => {
```

```
  try {
    const { MessageResponse } = await pinClient.send(
      new SendMessagesCommand(params),
    );

    if (!MessageResponse) {
      throw new Error("No message response.");
    }

    if (!MessageResponse.Result) {
      throw new Error("No message result.");
    }

    const recipientResult = MessageResponse.Result[toAddress];

    if (recipientResult.StatusCode !== 200) {
      throw new Error(recipientResult.StatusMessage);
    }
    console.log(recipientResult.MessageId);
  } catch (err) {
    console.log(err.message);
  }
};

run();
```

Envoyer un SMS.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

/* The phone number or short code to send the message from. The phone number
 or short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format. */
const originationNumber = "SENDER_NUMBER"; //e.g., +1XXXXXXXXXX

// The recipient's phone number.  For best results, you should specify the phone
 number in E.164 format.
const destinationNumber = "RECEIVER_NUMBER"; //e.g., +1XXXXXXXXXX
```

```
// The content of the SMS message.
const message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

/*The Amazon Pinpoint project/application ID to use when you send this message.
Make sure that the SMS channel is enabled for the project or application
that you choose.*/
const projectId = "PINPOINT_PROJECT_ID"; //e.g., XXXXXXXX66e4e9986478cXXXXXXXXX

/* The type of SMS message that you want to send. If you plan to send
time-sensitive content, specify TRANSACTIONAL. If you plan to send
marketing-related content, specify PROMOTIONAL.*/
const messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
const registeredKeyword = "myKeyword";

/* The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html.*/

const senderId = "MySenderID";

// Specify the parameters to pass to the API.
const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
```

```
  },
};

const run = async () => {
  try {
    const data = await pinClient.send(new SendMessagesCommand(params));
    console.log(
      `Message sent!
 ${data.MessageResponse.Result[destinationNumber].StatusMessage}`,
    );
  } catch (err) {
    console.log(err);
  }
};
run();
```

- Pour plus de détails sur l'API, reportez-vous SendMessagesà la section Référence des AWS SDK pour JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Envoyer un e-mail.

```
"use strict";

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

// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";
```

```
// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using
 the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
// message body of the email.
var charset = "UTF-8";

// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });
```

```
//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

//Try to send the email.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"][toAddress]["MessageId"]
    );
```

```
  }
});
```

Envoyer un SMS.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number.  For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
```

```
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
var senderId = "MySenderID";

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

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
```

```
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

- Pour plus de détails sur l'API, reportez-vous SendMessagesà la section Référence des AWS SDK pour JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */

val body: String =
    """
    Amazon Pinpoint test (AWS SDK for Kotlin)

    This email was sent through the Amazon Pinpoint Email API using the AWS SDK
 for Kotlin.
```

```
    """.trimIndent()

suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <subject> <appId> <senderAddress> <toAddress>

    Where:
        subject - The email subject to use.
        senderAddress - The from address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email
        toAddress - The to address. This address has to be verified in Amazon
 Pinpoint in the region you're using to send email
    """

    if (args.size != 3) {
        println(usage)
        exitProcess(0)
    }

    val subject = args[0]
    val senderAddress = args[1]
    val toAddress = args[2]
    sendEmail(subject, senderAddress, toAddress)
}

suspend fun sendEmail(
    subjectVal: String?,
    senderAddress: String,
    toAddressVal: String,
) {
    var content =
        Content {
            data = body
        }

    val messageBody =
        Body {
            text = content
        }

    val subContent =
        Content {
            data = subjectVal
```

```
        }

    val message =
        Message {
            body = messageBody
            subject = subContent
        }

    val destinationOb =
        Destination {
            toAddresses = listOf(toAddressVal)
        }

    val emailContent =
        EmailContent {
            simple = message
        }

    val sendEmailRequest =
        SendEmailRequest {
            fromEmailAddress = senderAddress
            destination = destinationOb
            this.content = emailContent
        }

    PinpointEmailClient.fromEnvironment { region = "us-east-1" }.use
 { pinpointemail ->
        pinpointemail.sendEmail(sendEmailRequest)
        println("Message Sent")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous SendMessagesà la section AWS SDK pour la référence de l'API Kotlin.

Python

Kit SDK for Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Envoyer un e-mail.

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

logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
 account
                         is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and
 message
```

```
                    body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
 can
                         display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": sender,
                        "SimpleEmail": {
                            "Subject": {"Charset": char_set, "Data": subject},
                            "HtmlPart": {"Charset": char_set, "Data":
 html_message},
                            "TextPart": {"Charset": char_set, "Data":
 text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


def main():
    app_id = "ce796be37f32f178af652b26eexample"
```

```
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -------------------------------------
    This email was sent with Amazon Pinpoint using the AWS SDK for Python
 (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
                """
    html_message = """<html>
    <head></head>
    <body>
      <h1>Amazon Pinpoint Test (SDK for Python (Boto3)</h1>
      <p>This email was sent with
        <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
        <a href='https://aws.amazon.com/sdk-for-python/'>
          AWS SDK for Python (Boto3)</a>.</p>
    </body>
    </html>
                """

    print("Sending email.")
    message_ids = send_email_message(
        boto3.client("pinpoint"),
        app_id,
        sender,
        [to_address],
        char_set,
        subject,
        html_message,
        text_message,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

Envoyer un SMS.

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

logger = logging.getLogger(__name__)


def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you
 send
                   this message. The SMS channel must be enabled for the project
 or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This
 phone
                               number must be associated with your Amazon
 Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you
 send
                         time-sensitive content, specify TRANSACTIONAL. If you
 send
                         marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
```

```
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK
 for "
        "Python (Boto 3)."
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

Envoyer un e-mail à partir d'un modèle d'e-mail existant.

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

logger = logging.getLogger(__name__)


def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
 template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
                         account is in the sandbox, these addresses must be
 verified.
    :param template_name: The name of the email template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {"EmailMessage": {"FromAddress":
 sender}},
                "TemplateConfiguration": {
                    "EmailTemplate": {
                        "Name": template_name,
                        "Version": template_version,
```

```
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
        template_version,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

Envoyer un SMS à partir d'un modèle de SMS existant.

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

```
logger = logging.getLogger(__name__)


def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
    origination_number,
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
    :param message_type: The type of SMS message (promotional or transactional).
    :param origination_number: The phone number that the message is sent from.
    :param template_name: The name of the SMS template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
                "TemplateConfiguration": {
                    "SMSTemplate": {"Name": template_name, "Version":
 template_version}
                },
            },
        )
```

```
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),
        project_id,
        destination_number,
        message_type,
        origination_number,
        template_name,
        template_version,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

- Pour plus de détails sur l'API, consultez SendMessagesle AWS manuel de référence de l'API SDK for Python (Boto3).

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Envoyer un e-mail.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
        io_simpleemail = NEW /aws1/cl_pptsimpleemail(
          io_subject = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_subject
          )
          io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_html_message
          )
          io_textpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_text_message
          )
        )
```

```
      )
    )
  )
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Email message sent successfully.' TYPE 'I'.
```

Envoyer un SMS.

```
" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_body = iv_message
        iv_messagetype = iv_message_type
        iv_originationnumber = iv_origination_number
      )
    )
  )
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
```

```
  ENDIF.
ENDLOOP.

MESSAGE 'SMS message sent successfully.' TYPE 'I'.
```

Envoyer un e-mail à partir d'un modèle d'e-mail existant.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
      )
    )
    io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
      io_emailtemplate = NEW /aws1/cl_ppttemplate(
        iv_name = iv_template_name
        iv_version = iv_template_version
      )
    )
  )
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Templated email message sent successfully.' TYPE 'I'.
```

Envoyer un SMS à partir d'un modèle de SMS existant.

```
" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_messagetype = iv_message_type
        iv_originationnumber = iv_origination_number
      )
    )
    io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
      io_smstemplate = NEW /aws1/cl_ppttemplate(
        iv_name = iv_template_name
        iv_version = iv_template_version
      )
    )
  )
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
  ENDIF.
ENDLOOP.

MESSAGE 'Templated SMS message sent successfully.' TYPE 'I'.
```

- Pour plus de détails sur l'API, reportez-vous SendMessagesà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`UpdateEndpoint`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`UpdateEndpoint`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.EndpointDemographic;
import software.amazon.awssdk.services.pinpoint.model.EndpointLocation;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.UUID;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Date;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appId>

                Where:
                  appId - The ID of the application to create an endpoint for.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        EndpointResponse response = createEndpoint(pinpoint, appId);
        System.out.println("Got Endpoint: " + response.id());
        pinpoint.close();
    }

    public static EndpointResponse createEndpoint(PinpointClient client, String
 appId) {
        String endpointId = UUID.randomUUID().toString();
        System.out.println("Endpoint ID: " + endpointId);

        try {
            EndpointRequest endpointRequest = createEndpointRequestData();
```

```
            UpdateEndpointRequest updateEndpointRequest =
UpdateEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .endpointRequest(endpointRequest)
                    .build();

            UpdateEndpointResponse updateEndpointResponse =
client.updateEndpoint(updateEndpointRequest);
            System.out.println("Update Endpoint Response: " +
updateEndpointResponse.messageBody());

            GetEndpointRequest getEndpointRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            GetEndpointResponse getEndpointResponse =
client.getEndpoint(getEndpointRequest);
            System.out.println(getEndpointResponse.endpointResponse().address());

System.out.println(getEndpointResponse.endpointResponse().channelType());

System.out.println(getEndpointResponse.endpointResponse().applicationId());

System.out.println(getEndpointResponse.endpointResponse().endpointStatus());

System.out.println(getEndpointResponse.endpointResponse().requestId());
            System.out.println(getEndpointResponse.endpointResponse().user());

            return getEndpointResponse.endpointResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static EndpointRequest createEndpointRequestData() {
        try {
            List<String> favoriteTeams = new ArrayList<>();
            favoriteTeams.add("Lakers");
            favoriteTeams.add("Warriors");
```

```
            HashMap<String, List<String>> customAttributes = new HashMap<>();
            customAttributes.put("team", favoriteTeams);

            EndpointDemographic demographic = EndpointDemographic.builder()
                    .appVersion("1.0")
                    .make("apple")
                    .model("iPhone")
                    .modelVersion("7")
                    .platform("ios")
                    .platformVersion("10.1.1")
                    .timezone("America/Los_Angeles")
                    .build();

            EndpointLocation location = EndpointLocation.builder()
                    .city("Los Angeles")
                    .country("US")
                    .latitude(34.0)
                    .longitude(-118.2)
                    .postalCode("90068")
                    .region("CA")
                    .build();

            Map<String, Double> metrics = new HashMap<>();
            metrics.put("health", 100.00);
            metrics.put("luck", 75.00);

            EndpointUser user = EndpointUser.builder()
                    .userId(UUID.randomUUID().toString())
                    .build();

            DateFormat df = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm'Z'"); //
Quoted "Z" to indicate UTC, no timezone
                                                                           //
offset
            String nowAsISO = df.format(new Date());

            return EndpointRequest.builder()
                    .address(UUID.randomUUID().toString())
                    .attributes(customAttributes)
                    .channelType("APNS")
                    .demographic(demographic)
                    .effectiveDate(nowAsISO)
                    .location(location)
                    .metrics(metrics)
```

```
                    .optOut("NONE")
                    .requestId(UUID.randomUUID().toString())
                    .user(user)
                    .build();

            } catch (PinpointException e) {
                System.err.println(e.awsErrorDetails().errorMessage());
                System.exit(1);
            }
            return null;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous UpdateEndpointà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Scénarios d'utilisation d'Amazon Pinpoint AWS SDKs

Les exemples de code suivants vous montrent comment implémenter des scénarios courants dans Amazon Pinpoint avec. AWS SDKs Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'Amazon Pinpoint ou en les combinant à d'autres. Services AWS Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la configuration et l'exécution du code.

Les scénarios ciblent un niveau d'expérience intermédiaire pour vous aider à comprendre les actions de service dans leur contexte.

Exemples

- Commencer à utiliser Final User Messaging Push

Commencer à utiliser Final User Messaging Push

L'exemple de code suivant illustre comment :

- Création d'une application

- Activer les canaux de notification push
- Envoyer une notification push
- nettoyer des ressources ;

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code.

```
#!/bin/bash

# AWS End User Messaging Push Getting Started Script
# This script creates an AWS End User Messaging Push application and demonstrates
# how to enable push notification channels and send a test message.
#
# Prerequisites:
# - AWS CLI installed and configured
# - Appropriate IAM permissions for Pinpoint operations
#
# Usage: ./aws-end-user-messaging-gs.sh [--auto-cleanup]

set -uo pipefail

# Security: Set secure umask for created files
umask 0077

# Set up logging
LOG_DIR="./aws-eump-logs"
mkdir -p "$LOG_DIR"
LOG_FILE="$LOG_DIR/aws-end-user-messaging-push-script-$(date +%Y%m%d-%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS End User Messaging Push setup script..."
echo "Logging to $LOG_FILE"
echo "Timestamp: $(date)"
```

```
# Security: Track created resources for cleanup
declare -a TEMP_FILES=()
declare -a AWS_RESOURCES=()

# Cleanup function with improved security
cleanup() {
    local exit_code=$?
    echo "Cleaning up temporary resources..."

    # Remove temporary files securely
    for temp_file in "${TEMP_FILES[@]+"${TEMP_FILES[@]}"}"; do
        if [ -f "$temp_file" ]; then
            shred -vfz -n 3 "$temp_file" 2>/dev/null || rm -f "$temp_file"
        fi
    done

    # Optionally delete AWS resources
    if [ "${DELETE_AWS_RESOURCES:-false}" = "true" ]; then
        for resource in "${AWS_RESOURCES[@]+"${AWS_RESOURCES[@]}"}"; do
            echo "Deleting AWS resource: $resource"
            aws pinpoint delete-app --application-id "$resource" 2>/dev/null || \
                echo "Warning: Failed to delete application $resource"
        done
    fi

    exit "$exit_code"
}

trap cleanup EXIT INT TERM

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2
    local ignore_error=${3:-false}

    if echo "$output" | grep -qi "error\|exception\|fail"; then
        echo "ERROR: Command failed: $cmd" >&2
        echo "Error details: $output" >&2

        if [ "$ignore_error" = "true" ]; then
            echo "Ignoring error and continuing..." >&2
            return 1
        else
```

```
            return 2
        fi
    fi

    return 0
}

# Function to validate AWS CLI is configured
validate_aws_cli() {
    echo "Validating AWS CLI configuration..."

    # Check if AWS CLI is installed
    if ! command -v aws &> /dev/null; then
        echo "ERROR: AWS CLI is not installed. Please install it first." >&2
        echo "Visit: https://docs.aws.amazon.com/cli/latest/userguide/getting-
started-install.html" >&2
        return 1
    fi

    # Check AWS CLI version
    AWS_VERSION=$(aws --version 2>&1 | head -n1)
    echo "AWS CLI version: $AWS_VERSION"

    # Verify credentials are set (check for credential env vars or config file)
    if ! aws sts get-caller-identity &> /dev/null; then
        echo "ERROR: AWS CLI credentials are not configured or invalid." >&2
        echo "Please configure credentials via environment variables, credential
 file, or 'aws configure'" >&2
        return 1
    fi

    # Get current AWS identity and region
    CALLER_IDENTITY=$(aws sts get-caller-identity)
    CURRENT_REGION=$(aws configure get region 2>/dev/null || echo "us-east-1")
    echo "AWS CLI configured for:"
    echo "$CALLER_IDENTITY"
    echo "Current region: $CURRENT_REGION"
    echo ""

    return 0
}

# Function to check if jq is available for JSON parsing
check_json_tools() {
```

```
    if command -v jq &> /dev/null; then
        USE_JQ=true
        echo "jq is available for JSON parsing"
    else
        USE_JQ=false
        echo "jq is not available, using grep for JSON parsing"
        echo "Consider installing jq for better JSON handling: https://
stedolan.github.io/jq/"
    fi
}

# Function to extract JSON values safely
extract_json_value() {
    local json=$1
    local key=$2

    if [ "$USE_JQ" = "true" ]; then
        echo "$json" | jq -r ".ApplicationResponse.$key // empty" 2>/dev/null ||
 echo ""
    else
        # Fallback to grep method with better validation
        echo "$json" | grep -o "\"$key\": \"[^\"]*" | cut -d'"' -f4 | head -n1 ||
 echo ""
    fi
}

# Function to validate required IAM permissions
validate_permissions() {
    echo "Validating IAM permissions..."

    # Test basic Pinpoint permissions
    if ! aws pinpoint get-apps > /dev/null 2>&1; then
        echo "WARNING: Unable to list Pinpoint applications." >&2
        echo "Please ensure you have appropriate IAM permissions for Pinpoint
 operations." >&2
        echo "Required permissions:" >&2
        echo "  - mobiletargeting:GetApps" >&2
        echo "  - mobiletargeting:CreateApp" >&2
        echo "  - mobiletargeting:DeleteApp" >&2
        echo "  - mobiletargeting:UpdateGcmChannel" >&2
        echo "  - mobiletargeting:UpdateApnsChannel" >&2
        echo "  - mobiletargeting:SendMessages" >&2
    else
        echo "Basic Pinpoint permissions validated."
```

```
    fi
}

# Function to validate input parameters
validate_input() {
    local app_name=$1

    # Validate app name length and characters
    if [ ${#app_name} -gt 64 ]; then
        echo "ERROR: Application name exceeds maximum length of 64 characters"
 >&2
        return 1
    fi

    if ! [[ "$app_name" =~ ^[a-zA-Z0-9_-]+$ ]]; then
        echo "ERROR: Application name contains invalid characters" >&2
        return 1
    fi

    return 0
}

# Function to create secure temporary files
create_temp_file() {
    local temp_file
    temp_file=$(mktemp) || {
        echo "ERROR: Failed to create temporary file" >&2
        return 1
    }
    chmod 600 "$temp_file"
    TEMP_FILES+=("$temp_file")
    echo "$temp_file"
}

# Validate prerequisites
if ! validate_aws_cli; then
    exit 1
fi

check_json_tools
validate_permissions

# Generate a random suffix for resource names to avoid conflicts
RANDOM_SUFFIX=$(LC_ALL=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 8 | head -n1)
```

```
APP_NAME="PushNotificationApp-${RANDOM_SUFFIX}"

# Validate input
if ! validate_input "$APP_NAME"; then
    exit 1
fi

echo "Creating application with name: $APP_NAME"

# Step 1: Create an application
echo "Executing: aws pinpoint create-app --create-application-request Name=
${APP_NAME}"
CREATE_APP_OUTPUT=$(aws pinpoint create-app --create-application-request "Name=
${APP_NAME},tags={project=doc-smith,tutorial=aws-end-user-messaging-gs}" 2>&1)

if ! check_error "$CREATE_APP_OUTPUT" "create-app"; then
    exit 1
fi

echo "Application created successfully:"
echo "$CREATE_APP_OUTPUT"

# Extract the application ID from the output
APP_ID=$(extract_json_value "$CREATE_APP_OUTPUT" "Id")

if [ -z "$APP_ID" ] || [ "$APP_ID" = "null" ]; then
    echo "ERROR: Failed to extract application ID from output" >&2
    echo "Output was: $CREATE_APP_OUTPUT" >&2
    exit 1
fi

echo "Application ID: $APP_ID"
AWS_RESOURCES+=("$APP_ID")

# Step 2: Enable FCM (GCM) channel with a sample API key
echo ""
echo "========================================="
echo "ENABLING FCM (GCM) CHANNEL"
echo "========================================="
echo "Note: This is using a placeholder API key for demonstration purposes only."
echo "In a production environment, you should use your actual FCM API key from
 Firebase Console."
echo ""
```

```
echo "IMPORTANT: The following command will likely fail because we're using a
 placeholder API key."
echo "This is expected behavior for this demonstration script."

echo "Executing: aws pinpoint update-gcm-channel --application-id $APP_ID --gcm-
channel-request ..."
UPDATE_GCM_OUTPUT=$(aws pinpoint update-gcm-channel \
    --application-id "$APP_ID" \
    --gcm-channel-request '{"Enabled": true, "ApiKey": "sample-fcm-api-key-for-
demo-only"}' 2>&1)

# We'll ignore this specific error since we're using a placeholder API key
if check_error "$UPDATE_GCM_OUTPUT" "update-gcm-channel" "true"; then
    echo "FCM channel enabled successfully:"
    echo "$UPDATE_GCM_OUTPUT"
else
    echo "As expected, FCM channel update failed with the placeholder API key."
    echo "Error details: $UPDATE_GCM_OUTPUT"
    echo ""
    echo "To enable FCM in production:"
    echo "1. Go to Firebase Console (https://console.firebase.google.com/)"
    echo "2. Create or select your project"
    echo "3. Go to Project Settings > Cloud Messaging"
    echo "4. Copy the Server Key"
    echo "5. Replace 'sample-fcm-api-key-for-demo-only' with your actual Server
 Key"
fi

# Step 3: Try to enable APNS channel (this will also fail without real
 certificates)
echo ""
echo "========================================="
echo "ENABLING APNS CHANNEL (OPTIONAL)"
echo "========================================="
echo "Attempting to enable APNS channel with placeholder certificate..."
echo "This will also fail without real APNS certificates, which is expected."

# Create a placeholder APNS configuration
echo "Executing: aws pinpoint update-apns-channel --application-id $APP_ID --
apns-channel-request ..."
UPDATE_APNS_OUTPUT=$(aws pinpoint update-apns-channel \
    --application-id "$APP_ID" \
    --apns-channel-request '{"Enabled": true, "Certificate": "placeholder-
certificate", "PrivateKey": "placeholder-private-key"}' 2>&1)
```

```
if check_error "$UPDATE_APNS_OUTPUT" "update-apns-channel" "true"; then
    echo "APNS channel enabled successfully:"
    echo "$UPDATE_APNS_OUTPUT"
else
    echo "As expected, APNS channel update failed with placeholder certificates."
    echo "Error details: $UPDATE_APNS_OUTPUT"
    echo ""
    echo "To enable APNS in production:"
    echo "1. Generate APNS certificates from Apple Developer Console"
    echo "2. Convert certificates to PEM format"
    echo "3. Use the actual certificate and private key in the update-apns-
channel command"
fi

# Step 4: Create message files for different platforms
echo ""
echo "=========================================="
echo "CREATING MESSAGE FILES"
echo "=========================================="

# Create FCM message file securely
GCM_MESSAGE_FILE=$(create_temp_file)
echo "Creating FCM message file..."
cat > "$GCM_MESSAGE_FILE" << 'EOF'
{
  "Addresses": {
    "SAMPLE-DEVICE-TOKEN-FCM": {
      "ChannelType": "GCM"
    }
  },
  "MessageConfiguration": {
    "GCMMessage": {
      "Action": "OPEN_APP",
      "Body": "Hello from AWS End User Messaging Push! This is an FCM
 notification.",
      "Priority": "normal",
      "SilentPush": false,
      "Title": "My First FCM Push Notification",
      "TimeToLive": 30,
      "Data": {
        "key1": "value1",
        "key2": "value2"
      }
```

```
    }
  }
}
EOF

# Create APNS message file securely
APNS_MESSAGE_FILE=$(create_temp_file)
echo "Creating APNS message file..."
cat > "$APNS_MESSAGE_FILE" << 'EOF'
{
  "Addresses": {
    "SAMPLE-DEVICE-TOKEN-APNS": {
      "ChannelType": "APNS"
    }
  },
  "MessageConfiguration": {
    "APNSMessage": {
      "Action": "OPEN_APP",
      "Body": "Hello from AWS End User Messaging Push! This is an APNS
 notification.",
      "Priority": "normal",
      "SilentPush": false,
      "Title": "My First APNS Push Notification",
      "TimeToLive": 30,
      "Badge": 1,
      "Sound": "default"
    }
  }
}
EOF

echo "Message files created:"
echo "- FCM message file (for FCM/Android)"
echo "- APNS message file (for APNS/iOS)"
echo ""
echo "Note: These messages use placeholder device tokens and will not actually be
 delivered."
echo "To send real messages, you would need to replace the sample device tokens
 with actual ones."

# Step 5: Demonstrate how to send messages (this will fail with placeholder
 tokens)
echo ""
echo "========================================="
```

```
echo "DEMONSTRATING MESSAGE SENDING"
echo "========================================="
echo "Attempting to send FCM message (will fail with placeholder token)..."

echo "Executing: aws pinpoint send-messages --application-id $APP_ID --message-
request file://<gcm-message>"
SEND_FCM_OUTPUT=$(aws pinpoint send-messages \
    --application-id "$APP_ID" \
    --message-request "file://$GCM_MESSAGE_FILE" 2>&1)

if check_error "$SEND_FCM_OUTPUT" "send-messages (FCM)" "true"; then
    echo "FCM message sent successfully:"
    echo "$SEND_FCM_OUTPUT"
else
    echo "As expected, FCM message sending failed with placeholder token."
    echo "Error details: $SEND_FCM_OUTPUT"
fi

echo ""
echo "Attempting to send APNS message (will fail with placeholder token)..."

echo "Executing: aws pinpoint send-messages --application-id $APP_ID --message-
request file://<apns-message>"
SEND_APNS_OUTPUT=$(aws pinpoint send-messages \
    --application-id "$APP_ID" \
    --message-request "file://$APNS_MESSAGE_FILE" 2>&1)

if check_error "$SEND_APNS_OUTPUT" "send-messages (APNS)" "true"; then
    echo "APNS message sent successfully:"
    echo "$SEND_APNS_OUTPUT"
else
    echo "As expected, APNS message sending failed with placeholder token."
    echo "Error details: $SEND_APNS_OUTPUT"
fi

# Step 6: Show application details
echo ""
echo "========================================="
echo "APPLICATION DETAILS"
echo "========================================="
echo "Retrieving application details..."

echo "Executing: aws pinpoint get-app --application-id $APP_ID"
GET_APP_OUTPUT=$(aws pinpoint get-app --application-id "$APP_ID" 2>&1)
```

```
if check_error "$GET_APP_OUTPUT" "get-app"; then
    echo "Application details:"
    echo "$GET_APP_OUTPUT"
fi

# Display summary of created resources
echo ""
echo "==========================================="
echo "RESOURCES CREATED"
echo "==========================================="
echo "AWS Resources:"
for resource in "${AWS_RESOURCES[@]+"${AWS_RESOURCES[@]}"}"; do
    echo "- Application: $resource"
done

echo ""
echo "Files created:"
echo "- $LOG_FILE (script log)"

# Auto-cleanup information
echo ""
echo "==========================================="
echo "CLEANUP INFORMATION"
echo "==========================================="
echo "This script created AWS resources that may incur charges."
echo "AWS resources will be automatically cleaned up on script exit."
echo ""
echo "To manually delete resources later, use:"
echo "  aws pinpoint delete-app --application-id $APP_ID"

# Set flag to delete AWS resources on cleanup
DELETE_AWS_RESOURCES=true

echo ""
echo "==========================================="
echo "SCRIPT COMPLETED SUCCESSFULLY"
echo "==========================================="
echo "This script demonstrated:"
echo "1. Creating an AWS End User Messaging Push application"
echo "2. Attempting to enable FCM and APNS channels (with placeholder
 credentials)"
echo "3. Creating message templates for different platforms"
echo "4. Demonstrating message sending commands (with placeholder tokens)"
echo "5. Retrieving application details"
```

```
echo "6. Proper cleanup of resources"
echo ""
echo "Security best practices implemented:"
echo "- Secure temporary file handling with restricted permissions"
echo "- Input validation for application names"
echo "- Error handling with proper exit codes"
echo "- Credential validation before AWS operations"
echo "- Automatic cleanup on script exit"
echo "- Secure file destruction for sensitive data"
echo ""
echo "For production use:"
echo "- Replace placeholder API keys with real FCM server keys"
echo "- Replace placeholder certificates with real APNS certificates"
echo "- Replace placeholder device tokens with real device tokens"
echo "- Use AWS IAM roles instead of long-term credentials"
echo "- Implement comprehensive error handling"
echo "- Store sensitive credentials in AWS Secrets Manager or Parameter Store"
echo "- Enable CloudTrail for audit logging"
echo "- Use VPC endpoints for private AWS API access"
echo ""
echo "Log file: $LOG_FILE"
echo "Script completed at: $(date)"
```

- Pour plus de détails sur l'API, consultez les rubriques suivantes dans la Référence des commandes de l'AWS CLI .
  - CreateApp
  - DeleteApp
  - GetApp
  - GetApps
  - GetCallerIdentity
  - SendMessages
  - UpdateApnsChannel
  - UpdateGcmChannel

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Exemples de code pour l'API SMS et voix Amazon Pinpoint à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser l'API SMS et voix Amazon Pinpoint avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

API de messages SMS et vocaux Amazon Pinpoint

- Exemples de base d'utilisation de l'API SMS et Voice Amazon Pinpoint AWS SDKs
  - Actions pour l'API SMS et voix Amazon Pinpoint à l'aide de AWS SDKs
    - Utilisation CreateConfigurationSet avec un AWS SDK
    - Utilisation CreateConfigurationSetEventDestination avec un AWS SDK
    - Utilisation DeleteConfigurationSet avec un AWS SDK
    - Utilisation DeleteConfigurationSetEventDestination avec un AWS SDK
    - Utilisation GetConfigurationSetEventDestinations avec un AWS SDK
    - Utilisation ListConfigurationSets avec un AWS SDK
    - Utilisation SendVoiceMessage avec un AWS SDK
    - Utilisation UpdateConfigurationSetEventDestination avec un AWS SDK

Exemples de base d'utilisation de l'API SMS et Voice Amazon Pinpoint AWS SDKs

Les exemples de code suivants montrent comment utiliser les bases de l'API SMS et vocale Amazon Pinpoint avec. AWS SDKs

Exemples

- Actions pour l'API SMS et voix Amazon Pinpoint à l'aide de AWS SDKs

- Utilisation CreateConfigurationSet avec un AWS SDK
- Utilisation CreateConfigurationSetEventDestination avec un AWS SDK
- Utilisation DeleteConfigurationSet avec un AWS SDK
- Utilisation DeleteConfigurationSetEventDestination avec un AWS SDK
- Utilisation GetConfigurationSetEventDestinations avec un AWS SDK
- Utilisation ListConfigurationSets avec un AWS SDK
- Utilisation SendVoiceMessage avec un AWS SDK
- Utilisation UpdateConfigurationSetEventDestination avec un AWS SDK

Actions pour l'API SMS et voix Amazon Pinpoint à l'aide de AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions individuelles sur les SMS et l'API vocale Amazon Pinpoint avec. AWS SDKs Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez les Références de l'API de messages SMS et vocaux d'Amazon Pinpoint.

Exemples

- Utilisation CreateConfigurationSet avec un AWS SDK
- Utilisation CreateConfigurationSetEventDestination avec un AWS SDK
- Utilisation DeleteConfigurationSet avec un AWS SDK
- Utilisation DeleteConfigurationSetEventDestination avec un AWS SDK
- Utilisation GetConfigurationSetEventDestinations avec un AWS SDK
- Utilisation ListConfigurationSets avec un AWS SDK
- Utilisation SendVoiceMessage avec un AWS SDK
- Utilisation UpdateConfigurationSetEventDestination avec un AWS SDK

Utilisation **`CreateConfigurationSet`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`CreateConfigurationSet`.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        " Create a new configuration set
        lo_pps->createconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        MESSAGE 'Configuration set created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous CreateConfigurationSetà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`CreateConfigurationSetEventDestination`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`CreateConfigurationSetEventDestination`.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        " Create event destination for the configuration set
        lo_pps->createconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
```

```
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous CreateConfigurationSetEventDestinationà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DeleteConfigurationSet`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DeleteConfigurationSet`.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        " Delete the configuration set
        lo_pps->deleteconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        MESSAGE 'Configuration set deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
```

```
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous DeleteConfigurationSetà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`DeleteConfigurationSetEventDestination`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DeleteConfigurationSetEventDestination`.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        " Delete the event destination
        lo_pps->deleteconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
```

```
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
        ).

        MESSAGE 'Event destination deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous DeleteConfigurationSetEventDestinationà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`GetConfigurationSetEventDestinations`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`GetConfigurationSetEventDestinations`.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        " Get event destinations for the configuration set
        oo_result = lo_pps->getconfseteventdestinations(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        " Process the event destinations
        LOOP AT oo_result->get_eventdestinations( ) INTO DATA(lo_event_dest).
          DATA(lv_dest_name) = lo_event_dest->get_name( ).
          DATA(lv_enabled) = lo_event_dest->get_enabled( ).

          MESSAGE |Event destination: { lv_dest_name }, Enabled: { lv_enabled }|
 TYPE 'I'.

          " Check for CloudWatch Logs destination
          DATA(lo_cloudwatch_dest) = lo_event_dest->get_cloudwatchlogsdst( ).
          IF lo_cloudwatch_dest IS NOT INITIAL.
            DATA(lv_log_group_arn) = lo_cloudwatch_dest->get_loggrouparn( ).
            MESSAGE |  CloudWatch Logs destination: { lv_log_group_arn }| TYPE
 'I'.
          ENDIF.

          " Check for Kinesis Firehose destination
          DATA(lo_firehose_dest) = lo_event_dest->get_kinesisfirehosedst( ).
          IF lo_firehose_dest IS NOT INITIAL.
            DATA(lv_delivery_stream) = lo_firehose_dest-
>get_deliverystreamarn( ).
            MESSAGE |  Kinesis Firehose destination: { lv_delivery_stream }| TYPE
 'I'.
          ENDIF.

          " Check for SNS destination
          DATA(lo_sns_dest) = lo_event_dest->get_snsdestination( ).
          IF lo_sns_dest IS NOT INITIAL.
            DATA(lv_topic_arn) = lo_sns_dest->get_topicarn( ).
            MESSAGE |  SNS destination: { lv_topic_arn }| TYPE 'I'.
          ENDIF.
        ENDLOOP.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
```

```
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous GetConfigurationSetEventDestinationsà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`ListConfigurationSets`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`ListConfigurationSets`.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        " List all configuration sets
        oo_result = lo_pps->listconfigurationsets(
          iv_nexttoken = iv_next_token    " Optional: Token for pagination
          iv_pagesize = iv_page_size      " Optional: Number of results per page,
e.g., '10'
        ).
```

```
        " Process the configuration sets
        LOOP AT oo_result->get_configurationsets( ) INTO DATA(lo_config_set).
          DATA(lv_config_set_name) = lo_config_set->get_value( ).
          MESSAGE |Configuration set: { lv_config_set_name }| TYPE 'I'.
        ENDLOOP.

        " Check if there are more results
        DATA(lv_next_token) = oo_result->get_nexttoken( ).
        IF lv_next_token IS NOT INITIAL.
          MESSAGE |More results available. Next token: { lv_next_token }| TYPE
'I'.
        ENDIF.

      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous ListConfigurationSetsà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`SendVoiceMessage`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `SendVoiceMessage`.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a
 list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";
```

```
   // The language to use when sending the message. For a list of supported
   // languages, see
   // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
   static final String languageCode = "en-US";

   // The content of the message. This example uses SSML to customize and
control
   // certain aspects of the message, such as by adding pauses and changing
   // phonation. The message can't contain any line breaks.
   static final String ssmlMessage = "<speak>This is a test message sent from "
           + "<emphasis>Amazon Pinpoint</emphasis> "
           + "using the <break strength='weak'/>AWS "
           + "SDK for Java. "
           + "<amazon:effect phonation='soft'>Thank "
           + "you for listening.</amazon:effect></speak>";

   public static void main(String[] args) {

       final String usage = """
               Usage:   <originationNumber> <destinationNumber>\s

               Where:
                 originationNumber - The phone number or short code that
you specify has to be associated with your Amazon Pinpoint account. For best
results, specify long codes in E.164 format (for example, +1-555-555-5654).
                 destinationNumber - The recipient's phone number.  For best
results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
               """;

       if (args.length != 2) {
           System.out.println(usage);
           System.exit(1);
       }
       String originationNumber = args[0];
       String destinationNumber = args[1];
       System.out.println("Sending a voice message");

       // Set the content type to application/json.
       List<String> listVal = new ArrayList<>();
       listVal.add("application/json");
       Map<String, List<String>> values = new HashMap<>();
       values.put("Content-Type", listVal);
```

```
        ClientOverrideConfiguration config2 =
ClientOverrideConfiguration.builder()
                .headers(values)
                .build();

        PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
                .overrideConfiguration(config2)
                .region(Region.US_EAST_1)
                .build();

        sendVoiceMsg(client, originationNumber, destinationNumber);
        client.close();
    }

    public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                                    String destinationNumber) {
        try {
            SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                    .languageCode(languageCode)
                    .text(ssmlMessage)
                    .voiceId(voiceName)
                    .build();

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

            SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()
                    .destinationPhoneNumber(destinationNumber)
                    .originationPhoneNumber(originationNumber)
                    .content(content)
                    .build();

            client.sendVoiceMessage(voiceMessageRequest);
            System.out.println("The message was sent successfully.");

        } catch (PinpointSmsVoiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous SendVoiceMessageà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best
 results, you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/
SupportedLanguage.html
var languageCode = "en-US";
```

```
// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint
 account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

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

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointsmsvoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
```

```
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointsmsvoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

- Pour plus de détails sur l'API, reportez-vous SendVoiceMessageà la section Référence des AWS SDK pour JavaScript API.

Python

Kit SDK for Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
```

```
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your
 Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your
 Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
 message.
    :param ssml_message: The content of the message. This example uses SSML to
 control
                         certain aspects of the message, such as the volume and
 the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
```

```
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
        )
        raise
    else:
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


if __name__ == "__main__":
    main()
```

- Pour plus de détails sur l'API, consultez SendVoiceMessagele AWS manuel de référence de l'API SDK for Python (Boto3).

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        " Create SSML message type object with voice parameters
        DATA(lo_ssml_message) = NEW /aws1/cl_ppsssmlmessagetype(
          iv_languagecode = iv_language_code    " e.g., 'en-US'
          iv_voiceid = iv_voice_id              " e.g., 'Matthew'
          iv_text = iv_ssml_message             " SSML formatted message text
        ).

        " Create voice message content with the SSML message
        DATA(lo_content) = NEW /aws1/cl_ppsvoicemessagecont(
          io_ssmlmessage = lo_ssml_message
        ).

        " Send the voice message
        DATA(lo_result) = lo_pps->sendvoicemessage(
          iv_originationphonenumber = iv_origination_number  " e.g.,
'+12065550110'
          iv_callerid = iv_caller_id                         " e.g.,
'+12065550199'
          iv_destinationphonenumber = iv_destination_number  " e.g.,
'+12065550142'
          io_content = lo_content
        ).

        " Retrieve the message ID from the response
        ov_message_id = lo_result->get_messageid( ).

        MESSAGE 'Voice message sent successfully.' TYPE 'I'.
```

```
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous SendVoiceMessageà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Utilisation **`UpdateConfigurationSetEventDestination`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`UpdateConfigurationSetEventDestination`.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        " Update the event destination
        lo_pps->updateconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
```

```
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination updated successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous UpdateConfigurationSetEventDestinationà la section de référence du AWS SDK pour l'API SAP ABAP.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Pinpoint avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Sécurité dans Amazon Pinpoint

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture 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. Third-party les auditeurs 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 Amazon Pinpoint, consultez la section 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 d'Amazon Pinpoint. Les rubriques suivantes vous montrent comment configurer Amazon Pinpoint 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 Amazon Pinpoint.

Pour plus d'informations, consultez le Guide d'architecture résiliente Amazon Pinpoint sur les architectures de référence.

Rubriques

- Protection des données dans Amazon Pinpoint
- Gestion des identités et des accès pour Amazon Pinpoint
- Journalisation et surveillance dans Amazon Pinpoint
- Validation de la conformité pour Amazon Pinpoint
- Résilience dans Amazon Pinpoint
- Sécurité de l'infrastructure dans Amazon Pinpoint

- Configuration et analyse des vulnérabilités dans Amazon Pinpoint
- Bonnes pratiques de sécurité pour Amazon Pinpoint

Protection des données dans Amazon Pinpoint

Le modèle de responsabilité AWS partagée (modèle de) s'applique à la protection des données dans Amazon Pinpoint. 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 la FAQ sur la confidentialité des données et les . Pour plus d'informations sur la protection des données en Europe, consultez le Centre du règlement général sur la protection des données (RGPD).

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 Amazon Pinpoint ou une autre entreprise à Services AWS l'aide de la console, de l'API ou AWS des AWS CLI SDK. 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.

Selon la façon dont vous configurez et utilisez le service, Amazon Pinpoint peut stocker les types de données personnelles suivants pour vous ou sur vos clients :

Données de configuration

Cela inclut les données de configuration de projet telles que les informations d'identification et les paramètres qui définissent comment et quand Amazon Pinpoint envoie des messages via les canaux pris en charge, ainsi que les segments d'utilisateurs auxquels il envoie des messages. Pour envoyer des messages, ces données peuvent inclure des adresses IP dédiées aux messages électroniques, des codes courts et des ID d'expéditeur pour les messages texte SMS, ainsi que des informations d'identification pour communiquer avec les services de notification push tels que le service Apple Push Notification (APN) et Firebase Cloud Messaging (FCM).

Données d'utilisateur et de point de terminaison

Cela inclut les attributs standard et personnalisés que vous utilisez pour stocker et gérer des données sur les utilisateurs et les points de terminaison d'un projet Amazon Pinpoint. Un attribut peut stocker des informations sur un utilisateur spécifique (son nom, par exemple) ou un point de terminaison spécifique pour un utilisateur (son adresse e-mail, son numéro de téléphone mobile ou le jeton de son appareil mobile). Ces données peuvent également inclure des ID utilisateur externes qui mettent en corrélation les utilisateurs d'un projet Amazon Pinpoint avec les utilisateurs d'un système externe, tel qu'un système de gestion de la relation client. Pour plus d'informations sur ce que ces données peuvent inclure, consultez les schémas Utilisateur et Point de terminaison dans la Référence de l'API Amazon Pinpoint.

Données analytiques

Cela inclut des données pour les métriques, également appelées indicateurs de performances clés, qui fournissent un aperçu de la performance d'un projet Amazon Pinpoint dans des

domaines tels que l'engagement des utilisateurs et l'activité d'achat. Cela inclut également des données pour les métriques qui fournissent un aperçu de la démographie des utilisateurs pour un projet. Les données peuvent dériver d'attributs standard et personnalisés pour les utilisateurs et les points de terminaison, tels que la ville où vit un utilisateur. Elles peuvent également dériver d'événements, tels que les événements d'ouverture et de clic pour les messages électroniques que vous envoyez pour un projet.

Données importées

Cela inclut toutes les données d'utilisateur, de segmentation et d'analyse que vous ajoutez ou importez à partir de sources externes et utilisez dans Amazon Pinpoint. Un exemple est un fichier JSON que vous importez dans Amazon Pinpoint (via la console directement ou à partir d'un compartiment Amazon S3) pour créer un segment statique. D'autres exemples sont les données de point de terminaison que vous ajoutez par programmation pour créer un segment dynamique, les adresses de point de terminaison auxquelles vous envoyez des messages directs et les événements pour lesquels vous configurez une application chargée de générer un rapport pour Amazon Pinpoint.

Rubriques

- Chiffrement des données
- Confidentialité du trafic inter-réseau
- Création d'un point de terminaison de VPC d'interface pour l'API Amazon Pinpoint

Chiffrement des données

Les données Amazon Pinpoint sont chiffrées en transit et au repos. Lorsque vous soumettez des données à Amazon Pinpoint, il les chiffre au fur et à mesure qu'il les reçoit et les stocke. Lorsque vous récupérez des données à partir d'Amazon Pinpoint, celles-ci vous sont transmises à l'aide des protocoles de sécurité actuels.

Chiffrement au repos

Amazon Pinpoint chiffre toutes les données qu'il stocke pour vous. Cela inclut les données de configuration, les données utilisateur et de point de terminaison, les données analytiques et toutes les données que vous ajoutez ou importez dans Amazon Pinpoint. Pour chiffrer vos données, Amazon Pinpoint utilise des clés AWS Key Management Service internes 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

Amazon Pinpoint utilise 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, Amazon Pinpoint utilise le protocole HTTPS et le protocole TLS 1.2. En outre, lorsque vous créez et gérez des ressources Amazon Pinpoint à l'aide de la console, d'un AWS SDK ou du AWS Command Line Interface, toutes les communications sont sécurisées via HTTPS et TLS 1.2.

Gestion des clés

Pour chiffrer vos données Amazon Pinpoint, Amazon Pinpoint utilise des clés AWS KMS 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 Amazon Pinpoint.

Confidentialité du trafic inter-réseau

La confidentialité du trafic interréseau fait référence à la sécurisation des connexions et du trafic entre Amazon Pinpoint et vos clients et applications sur site, et entre Amazon Pinpoint et les AWS autres ressources de la même région. AWS Les fonctionnalités et pratiques suivantes peuvent vous aider à garantir la confidentialité du trafic inter-réseaux pour Amazon Pinpoint.

Trafic entre Amazon Pinpoint et les clients et applications sur site

Pour établir une connexion privée entre Amazon Pinpoint et des clients et des applications sur votre réseau sur site, 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 à Amazon Pinpoint par le biais d'API publiées, nous vous recommandons de respecter les exigences d'Amazon Pinpoint relatives aux appels d'API. Amazon Pinpoint exige que les clients utilisent le protocole TLS (Transport Layer Security) version 1.2 ou ultérieure. Les clients doivent également prendre en charge les suites de chiffrement dotées d'une confidentialité absolue (PFS), telles que Ephemeral (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). 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 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 Amazon Pinpoint et d'autres AWS resources

Pour sécuriser les communications entre Amazon Pinpoint et les autres AWS ressources de la même AWS région, Amazon Pinpoint utilise HTTPS et TLS 1.2 par défaut.

Création d'un point de terminaison de VPC d'interface pour l'API Amazon Pinpoint

Vous pouvez établir une connexion privée entre votre cloud privé virtuel (VPC) et un point de terminaison dans Amazon Pinpoint en créant un point de terminaison de VPC d'interface.

Les points de terminaison d'interface sont alimentés par AWS PrivateLinkune technologie qui vous permet d'accéder en privé aux API Amazon Pinpoint sans passerelle Internet, appareil NAT, connexion VPN ou. Direct Connect Les instances de votre VPC ne requièrent pas d'adresses IP publiques pour communiquer avec les API Amazon Pinpoint intégrées à AWS PrivateLink.

Pour plus d’informations, consultez le Guide de l’utilisateur AWS PrivateLink.

Création d'un point de terminaison de VPC d'interface

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

Amazon Pinpoint prend en charge les noms de service suivants :

- `com.amazonaws.region.pinpoint`
- `com.amazonaws.region.pinpoint-sms-voice-v2`

Si vous activez le DNS privé pour un point de terminaison d'interface, vous pouvez envoyer des demandes d'API à Amazon Pinpoint en utilisant le nom DNS par défaut pour Région AWS, par

exemple,. `com.amazonaws.us-east-1.pinpoint` Pour plus d'informations, consultez Noms d'hôte DNS dans le Guide AWS PrivateLink .

Pour obtenir une liste de toutes les régions et de tous les points de terminaison où Amazon Pinpoint est actuellement disponible, consultez les points de terminaison du service AWS dans Référence générale d'Amazon Web Services.

Création d’une stratégie de point de terminaison de VPC

Vous pouvez attacher une politique de point de terminaison à votre point de terminaison de VPC qui contrôle l’accès. La politique spécifie les informations suivantes :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

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 : stratégie de point de terminaison de VPC

La stratégie de point de terminaison VPC suivante accorde l'accès aux actions Amazon Pinpoint répertoriées pour tous les principaux sur toutes les ressources.

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

Gestion des identités et des accès pour Amazon Pinpoint

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. Des administrateurs IAM contrôlent les personnes qui peuvent être authentifiées (connectées) et autorisées (disposant d'autorisations) à utiliser des ressources Amazon Pinpoint. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion de l’accès à l’aide de politiques
- Fonctionnement d'Amazon Pinpoint avec IAM
- Actions Amazon Pinpoint pour les politiques IAM
- Exemples de politique basée sur l'identité d'Amazon Pinpoint
- Rôles IAM pour les tâches Amazon Pinpoint courantes
- Dépannage de la gestion des identités et des accès Amazon Pinpoint

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 Dépannage de la gestion des identités et des accès Amazon Pinpoint)
- Administrateur du service : déterminez l’accès des utilisateurs et soumettez les demandes d’autorisation (voir Fonctionnement d'Amazon Pinpoint avec IAM)
- Administrateur IAM : rédigez des politiques pour gérer l’accès (voir Exemples de politique basée sur l'identité d'Amazon Pinpoint)

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.

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 d'utilisateur à un rôle

IAM (console) ou en appelant une opération d' AWS API AWS CLI ou d'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.

Identity-based politiques

Identity-based les politiques sont des documents de politique d'autorisation 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.

Identity-based les politiques 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.

Amazon Pinpoint prend en charge l'utilisation de stratégies basées sur l'identité pour contrôler l'accès aux ressources Amazon Pinpoint.

Resource-based politiques

Resource-based les politiques 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.

Resource-based les politiques sont des politiques intégrées qui se trouvent dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Amazon Pinpoint prend en charge l'utilisation de stratégies basées sur les ressources pour contrôler l'accès aux ressources Amazon Pinpoint.

Listes de contrôle d'accès (ACL)

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

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Vue d'ensemble des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Amazon Pinpoint ne prend pas en charge l'utilisation de listes ACL pour contrôler l'accès aux ressources Amazon Pinpoint.

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 (SCP) : spécifient 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 (RCP) : définissent les autorisations maximales disponibles pour les ressources de votre organisation. Pour plus d'informations, consultez Politiques de contrôle des ressources (RCP) dans le Guide de l'utilisateur AWS Organizations .
- 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.

Amazon Pinpoint prend en charge l'utilisation de ces types de stratégies pour contrôler l'accès aux ressources Amazon Pinpoint.

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.

Fonctionnement d'Amazon Pinpoint avec IAM

Pour utiliser Amazon Pinpoint, les utilisateurs de votre AWS compte ont besoin d'autorisations leur permettant de consulter les données d'analyse, de créer des projets, de définir des segments d'utilisateurs, de déployer des campagnes, etc. Si vous intégrez une application mobile ou Web avec Amazon Pinpoint, les utilisateurs de votre application ont également besoin d'accéder à Amazon Pinpoint. Cet accès permet à votre application d'enregistrer des points de terminaison et de signaler des données d'utilisation à Amazon Pinpoint. Pour accorder l'accès aux fonctionnalités d'Amazon Pinpoint, créez des politiques Gestion des identités et des accès AWS (IAM) qui autorisent les actions Amazon Pinpoint pour les identités IAM ou les ressources Amazon Pinpoint.

IAM est un service qui aide les administrateurs à contrôler en toute sécurité l'accès aux AWS ressources. Les politiques IAM incluent les instructions qui autorisent ou refusent certaines opérations que des utilisateurs spécifiques peuvent effectuer sur des ressources spécifiques. Amazon Pinpoint fournit un ensemble d'actions que vous pouvez utiliser dans les politiques IAM pour spécifier des autorisations granulaires pour les utilisateurs et les ressources Amazon Pinpoint. Cela signifie que vous pouvez octroyer le niveau d'accès approprié à Amazon Pinpoint sans créer des politiques excessivement permissives qui pourraient exposer des données importantes ou compromettre vos ressources. Par exemple, vous pouvez accorder un accès illimité à un administrateur Amazon Pinpoint et un accès en lecture seule aux personnes ayant besoin d'accéder seulement à un projet spécifique.

Avant d'utiliser IAM pour gérer l'accès à Amazon Pinpoint, vous devez comprendre quelles sont les fonctions IAM disponibles dans cette situation. Pour obtenir une vue d'ensemble de la manière dont Amazon Pinpoint et les autres AWS services fonctionnent avec IAM, consultez les AWS services compatibles avec IAM dans le guide de l'utilisateur d'IAM.

Rubriques

- Politiques basées sur l'identité Amazon Pinpoint
- Politiques d'autorisation basées sur les ressources Amazon Pinpoint
- Autorisation basée sur les balises Amazon Pinpoint
- Rôles IAM Amazon Pinpoint

Politiques basées sur l'identité Amazon Pinpoint

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. Amazon Pinpoint est compatible avec des actions, des ressources et des clés de condition spécifiques. 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.

Actions

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.

Cela signifie que les actions de stratégie contrôlent ce que les utilisateurs peuvent faire sur la console Amazon Pinpoint. Ils contrôlent également ce que les utilisateurs peuvent faire par programmation en utilisant directement les AWS SDK, le AWS Command Line Interface (AWS CLI) ou les API Amazon Pinpoint.

Les actions de politique dans Amazon Pinpoint utilisent les préfixes suivants :

- **`mobiletargeting`** : pour les actions dérivant de l'API Amazon Pinpoint, qui est l'API principale pour Amazon Pinpoint.

- **`sms-voice`** : pour les actions qui dérivent d'Amazon Pinpoint, une API supplémentaire qui fournit des options avancées pour l'utilisation et la gestion des canaux SMS et vocaux dans Amazon Pinpoint.

Par exemple, pour accorder à quelqu'un l'autorisation d'afficher des informations sur tous les segments d'un projet, action qui correspond à l'opération `GetSegments` dans l'API Amazon Pinpoint, insérez l'action `mobiletargeting:GetSegments` dans sa stratégie. Les déclarations de politique doivent inclure un élément `Action` ou `NotAction`. Amazon Pinpoint définit son propre ensemble d'actions qui décrivent les tâches que les utilisateurs peuvent effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules :

```
"Action": [
      "mobiletargeting:action1",
      "mobiletargeting:action2"
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Get`, incluez l’action suivante :

```
"Action": "mobiletargeting:Get*"
```

Cependant, une bonne pratique consiste à créer des stratégies qui suivent le principe du moindre privilège. En d'autres termes, vous devez créer des stratégies qui incluent uniquement les autorisations requises pour effectuer une action spécifique.

Pour obtenir la liste des actions Amazon Pinpoint que vous pouvez utiliser dans les politiques IAM, consultez Actions Amazon Pinpoint pour les politiques IAM.

Ressources

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": "*"
```

Par exemple, l'action `mobiletargeting:GetSegments` récupère des informations sur tous les segments associés à un projet Amazon Pinpoint spécifique. Vous identifiez un projet avec un ARN dans le format suivant :

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

Pour de plus amples informations sur le format des ARN, veuillez consulter Amazon Ressource Names (ARN) dans la Références générales AWS.

Dans les politiques IAM, vous pouvez spécifier des ARN pour les types de ressources Amazon Pinpoint suivants :

- Campagnes
- Parcours
- Modèles de message (appelés modèles dans certains contextes)
- Projets (appelés applis ou applications dans certains contextes)
- Modèles de recommandations (appelés recommandations dans certains contextes)
- Segments

Par exemple, pour créer une déclaration de stratégie pour le projet dont l'ID de projet est `810c7aab86d42fb2b56c8c966example`, utilisez l'ARN suivant :

```
"Resource": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

Pour spécifier tous les objets qui appartiennent à un compte spécifique, utilisez le caractère générique (*) :

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

Certaines actions Amazon Pinpoint, telles que des actions de création de ressources, ne peuvent pas être exécutées sur une ressource spécifique. Dans ce cas, vous devez utiliser le caractère générique (*) :

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

Dans les politiques IAM, vous pouvez aussi spécifier des ARN pour les types de ressources SMS et vocales Amazon Pinpoint suivants :

- Jeu de configurations
- Liste de désinscriptions
- Numéro de téléphone
- Groupe
- Id de l'expéditeur

Par exemple, pour créer une instruction de politique pour un numéro de téléphone dont l'ID est le numéro de téléphone `phone-123456789012345678901234567890`12, utilisez l'ARN suivant :

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-123456789012345678901234567890"
```

Pour indiquer tous les numéros de téléphone appartenant à un compte spécifique, utilisez un caractère générique (*) à la place de l'ID du numéro de téléphone :

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

Certaines actions vocales et SMS Amazon Pinpoint ne sont pas effectuées sur une ressource spécifique, comme celles relatives à la gestion des paramètres au niveau du compte, tels que les limites de dépenses. Dans ce cas, vous devez utiliser le caractère générique (*) :

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

Certaines actions d'API Amazon Pinpoint nécessitent plusieurs ressources. Par exemple, l'action `TagResource` peut ajouter une balise à plusieurs projets. Pour spécifier plusieurs ressources dans une seule instruction, séparez les ARN par des virgules.

```
"Resource": [
      "resource1",
      "resource2"
```

Pour afficher une liste des types de ressources Amazon Pinpoint et de leurs ARN, consultez Ressources définies par Amazon Pinpoint dans le Guide de l'utilisateur IAM. Pour découvrir les actions avec lesquelles vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon Pinpoint dans le Guide de l'utilisateur IAM.

Clés de condition

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.

Amazon Pinpoint définit son propre ensemble de clés de condition et prend également en charge l'utilisation des clés de condition globales. Pour consulter la liste de toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM. Pour afficher une liste des clés de condition Amazon Pinpoint, consultez Clés de condition pour Amazon Pinpoint dans le Guide de l'utilisateur IAM. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon Pinpoint dans le Guide de l'utilisateur IAM.

Exemples

Pour voir des exemples de politiques Amazon Pinpoint basées sur l'identité, consultez Exemples de politique basée sur l'identité d'Amazon Pinpoint.

Politiques d'autorisation basées sur les ressources Amazon Pinpoint

Resource-based les politiques d'autorisation sont des documents de politique JSON qui spécifient les actions qu'un principal spécifié peut effectuer sur une ressource Amazon Pinpoint et dans quelles conditions. Amazon Pinpoint prend en charge les politiques d'autorisation basées sur les ressources pour les campagnes, les parcours, les modèles de messages (modèles), les modèles de recommandation (moteurs de recommandation), les projets (applications) et les segments.

Exemples

Pour consulter des exemples de politiques basées sur les ressources Amazon Pinpoint, consultez the section called “Identity-based exemples de politiques”.

Autorisation basée sur les balises Amazon Pinpoint

Vous pouvez associer des balises à certains types de ressources Amazon Pinpoint ou transmettre des balises dans une demande à Amazon Pinpoint. 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/${TagKey}`, `aws:RequestTag/${TagKey}` ou `aws:TagKeys`.

Pour plus d'informations sur le balisage des ressources Amazon Pinpoint, y compris un exemple de politique IAM, consultez Gérer les balises de ressources Amazon Pinpoint.

Rôles IAM Amazon Pinpoint

Un rôle IAM est une entité au sein de votre compte AWS qui dispose d’autorisations spécifiques.

Utilisation d'informations d'identification temporaires avec Amazon Pinpoint

Vous pouvez utiliser des informations d’identification temporaires pour vous connecter à l’aide de la fédération, endosser un rôle IAM ou encore pour endosser un rôle intercompte. Vous obtenez des informations d'identification de sécurité temporaires en appelant AWS Security Token Service (AWS STS) des opérations d'API telles que AssumeRoleou GetFederationToken.

Amazon Pinpoint est compatible avec l'utilisation des informations d'identification temporaires.

Service-linked rôles

Service-linked les rôles permettent aux AWS services d'accéder aux ressources d'autres services pour effectuer une action en votre nom. Service-linked les rôles apparaissent dans votre compte IAM et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Amazon Pinpoint n'utilise pas de rôles liés à un service.

Rôles du service

Cette fonction permet à un service d’endosser une fonction du service en votre nom. Ce rôle autorise le service à accéder à des ressources d’autres services pour effectuer une action en votre nom. Les

rôles de service s'affichent dans votre compte IAM et sont la propriété du compte. Cela signifie qu'un administrateur IAM peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

Amazon Pinpoint prend en charge l'utilisation de rôles de service.

Actions Amazon Pinpoint pour les politiques IAM

Pour gérer l'accès aux ressources Amazon Pinpoint dans votre AWS compte, vous pouvez ajouter des actions Amazon Pinpoint Gestion des identités et des accès AWS aux politiques (IAM). En utilisant des actions dans les stratégies, vous pouvez contrôler ce que les utilisateurs peuvent faire sur la console Amazon Pinpoint. Vous pouvez également contrôler ce que les utilisateurs peuvent faire par programmation en utilisant directement les AWS SDK, le AWS Command Line Interface (AWS CLI) ou les API Amazon Pinpoint.

Dans une stratégie, vous spécifiez chaque action avec l'espace de noms Amazon Pinpoint approprié suivi d'un signe deux-points et du nom de l'action, par exemple `GetSegments`. La plupart des actions correspondent à une demande adressée à l'API Amazon Pinpoint à l'aide d'un URI et d'une méthode HTTP spécifiques. Par exemple, si vous autorisez l'action `mobiletargeting:GetSegments` dans la stratégie d'un utilisateur, l'utilisateur est autorisé à récupérer des informations sur tous les segments d'un projet en soumettant une demande HTTP GET à l'URI `/apps/projectId/segments`. Cette politique permet également à l'utilisateur de visualiser ces informations sur la console et de les récupérer à l'aide d'un AWS SDK ou du AWS CLI.

Chaque action est effectuée sur une ressource Amazon Pinpoint spécifique, que vous identifiez dans une déclaration de stratégie par son Amazon Resource Name (ARN). Par exemple, l'action `mobiletargeting:GetSegments` est effectuée sur une application spécifique, que vous identifiez avec l'ARN, `arn:aws:mobiletargeting:region:accountId:apps/projectId`.

Cette rubrique identifie les actions Amazon Pinpoint que vous pouvez ajouter aux politiques IAM pour votre compte AWS . Pour voir des exemples illustrant comment utiliser les actions des politiques pour gérer l'accès aux ressources Amazon Pinpoint, consultez Exemples de politique basée sur l'identité d'Amazon Pinpoint.

Rubriques

- Actions d'API Amazon Pinpoint
- Actions de l'API SMS et Voix Amazon Pinpoint Version 1

Actions d'API Amazon Pinpoint

Cette section identifie les actions pour les fonctionnalités disponibles à partir de l'API Amazon Pinpoint, qui est l'API principale pour Amazon Pinpoint. Pour en savoir plus sur cette API, consultez la Référence de l'API Amazon Pinpoint.

Catégories :

- Analyses et métriques
- Campagnes
- Canaux
- Points de terminaison
- Flux d’événement
- Événements
- Tâches d'exportation
- Tâches d'importation
- Parcours
- Modèles de messages
- Messages
- One-time mots de passe
- Validation du numéro de téléphone
- Projets
- Modèles de recommandation
- Segments
- Étiquettes
- Utilisateurs

Analyses et métriques

Les autorisations suivantes sont liées à l'affichage des données analytiques sur la console Amazon Pinpoint. Elles sont également liées à la récupération (interrogation) des données agrégées pour les métriques standard, également appelées indicateurs de performance clés (KPI), qui s'appliquent aux projets, campagnes et parcours.

`mobiletargeting:GetReports`

Permet d'afficher les données d'analyse sur la console Amazon Pinpoint. Cette autorisation est également requise pour créer des segments contenant des attributs personnalisés à l'aide de la console Amazon Pinpoint. Il est également nécessaire d'obtenir une estimation de la taille d'un segment dans la console Amazon Pinpoint.

- URI – Non applicable
- Méthode – Non applicable
- ARN des ressources – `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:*`

`mobiletargeting:GetApplicationDateRangeKpi`

Récupérer (interroger) des données agrégées pour une métrique d'application standard. Il s'agit d'une métrique qui s'applique à toutes les campagnes ou messages transactionnels associés à un projet.

- URI – `/apps/`*`projectId`*`/kpis/daterange/`*`kpi-name`*
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:apps/`*`projectId`*`/kpis/daterange/`*`kpi-name`*

`mobiletargeting:GetCampaignDateRangeKpi`

Récupérer (interroger) des données agrégées pour une métrique de campagne standard. Il s'agit d'une métrique qui s'applique à une campagne individuelle.

- URI – `/apps/`*`projectId`*`/campaigns/`*`campaignId`*`/kpis/daterange/`*`kpi-name`*
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:apps/`*`projectId`*`/campaigns/`*`campaignId`*`/kpis/daterange/`*`kpi-name`*

`mobiletargeting:GetJourneyDateRangeKpi`

Récupérer (interroger) des données agrégées pour une métrique d'engagement du parcours standard. Il s'agit d'un indicateur d'engagement qui s'applique à un parcours individuel, par exemple le nombre de messages ouverts par les participants pour toutes les activités d'un parcours.

- URI – `/apps/`*`projectId`*`/journeys/`*`journeyId`*`/kpis/daterange/`*`kpi-name`*

- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/kpis/daterange/kpi-name`

`mobiletargeting:GetJourneyExecutionMetrics`

Récupérez (interrogez) des données agrégées pour les mesures d'exécution standard qui s'appliquent à un parcours individuel, par exemple le nombre de participants qui effectuent activement toutes les activités d'un parcours.

- URI – `/apps/projectId/journeys/journeyId/execution-metrics`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/execution-metrics`

`mobiletargeting:GetJourneyExecutionActivityMetrics`

Récupérez (interrogez) des données agrégées pour les mesures d'exécution standard qui s'appliquent à une activité individuelle dans un parcours ; par exemple, le nombre de participants qui ont commencé ou terminé une activité.

- URI – `/apps/projectId/journeys/journeyId/activities/journey-activity-id/execution-metrics`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/activities/journey-activity-id/execution-metrics`

Campagnes

Les autorisations suivantes sont liées à la gestion des campagnes dans votre compte Amazon Pinpoint.

`mobiletargeting:CreateCampaign`

Créez une campagne pour un projet.

- URI – `/apps/projectId/campaigns`

- Méthode – POST
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns

mobiletargeting:DeleteCampaign

Supprime une campagne spécifique.

- URI – /apps/*projectId*/campaigns/*campaignId*
- Méthode – DELETE
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaign

Récupère des informations sur une campagne spécifique.

- URI – /apps/*projectId*/campaigns/*campaignId*
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignActivities

Récupère des informations sur les activités effectuées par une campagne.

- URI – /apps/*projectId*/campaigns/*campaignId*/activities
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaigns

Récupérez des informations sur toutes les campagnes d'un projet.

- URI – /apps/*projectId*/campaigns
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetCampaignVersion

Récupère des informations sur une version de campagne spécifique.

- URI – /apps/*projectId*/campaigns/*campaignId*/versions/*versionId*
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignVersions

Récupère des informations sur les versions actuelles et antérieures d'une campagne.

- URI – /apps/*projectId*/campaigns/*campaignId*/versions
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:UpdateCampaign

Met à jour une campagne spécifique.

- URI – /apps/*projectId*/campaigns/*campaignId*
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

Canaux

Les autorisations suivantes sont liées à la gestion des canaux dans votre compte Amazon Pinpoint. Dans Amazon Pinpoint, les canaux font référence aux méthodes que vous utilisez pour contacter vos clients, par exemple, l'envoi d'un e-mail, de SMS ou de notifications push.

mobiletargeting:DeleteAdmChannel

Désactive le canal Amazon Device Messaging (ADM) pour un projet.

- URI – /apps/*projectId*/channels/adm
- Méthode – DELETE

- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

`mobiletargeting:GetAdmChannel`

Récupérez des informations sur le canal ADM d'un projet.

- URI – /apps/*projectId*/channels/adm
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

`mobiletargeting:UpdateAdmChannel`

Activez ou mettez à jour le canal ADM d'un projet.

- URI – /apps/*projectId*/channels/adm
- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

`mobiletargeting:DeleteApnsChannel`

Supprimez le canal du service de notification en mode push Apple (Apple Push Notification Service, APNS) d'un projet.

- URI – /apps/*projectId*/channels/apns
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

`mobiletargeting:GetApnsChannel`

Récupérez des informations sur le canal des APN d'un projet.

- URI – /apps/*projectId*/channels/apns
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

`mobiletargeting:UpdateApnsChannel`

Activer ou mettre à jour le canal APN d'un projet.

- URI – /apps/*projectId*/channels/apns
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:DeleteApnsSandboxChannel

Mettez à jour le canal d'environnement de test (sandbox) des APN d'un projet.

- URI – /apps/*projectId*/channels/apns_sandbox
- Méthode – DELETE
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:GetApnsSandboxChannel

Récupérez des informations sur le canal d'environnement de test des APN d'un projet.

- URI – /apps/*projectId*/channels/apns_sandbox
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:UpdateApnsSandboxChannel

Activer ou mettre à jour le canal d'environnement de test (sandbox) APN pour un projet.

- URI – /apps/*projectId*/channels/apns_sandbox
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:DeleteApnsVoipChannel

Désactive le canal VoIP des APN d'un projet.

- URI – /apps/*projectId*/channels/apns_voip
- Méthode – DELETE

- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:GetApnsVoipChannel`

Récupérez des informations sur le canal VoIP des APN d'un projet.

- URI – /apps/*projectId*/channels/apns_voip
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:UpdateApnsVoipChannel`

Activer ou mettre à jour le canal VoIP APN d'un projet.

- URI – /apps/*projectId*/channels/apns_voip
- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:DeleteApnsVoipSandboxChannel`

Désactive le canal d'environnement de test (sandbox) VoIP des APN d'un projet.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

`mobiletargeting:GetApnsVoipSandboxChannel`

Récupérez des informations sur le canal d'environnement de test VoIP des APN d'un projet.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

mobiletargeting:UpdateApnsVoipSandboxChannel

Activer ou mettre à jour le canal d'environnement de test (sandbox) VoIP APN d'un projet.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip_sandbox

mobiletargeting:DeleteBaiduChannel

Désactive le canal Baidu Cloud Push d'un projet.

- URI – /apps/*projectId*/channels/baidu
- Méthode – DELETE
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:GetBaiduChannel

Récupérez des informations sur le canal Baidu Cloud Push d'un projet.

- URI – /apps/*projectId*/channels/baidu
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:UpdateBaiduChannel

Activez ou mettez à jour le canal Baidu Cloud Push d'un projet.

- URI – /apps/*projectId*/channels/baidu
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:DeleteEmailChannel

Désactive le canal des e-mails pour un projet

- URI – /apps/*projectId*/channels/email
- Méthode – DELETE

- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/email`

`mobiletargeting:GetEmailChannel`

Récupérez des informations sur le canal des e-mails d'un projet.

- URI – `/apps/projectId/channels/email`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/email`

`mobiletargeting:UpdateEmailChannel`

Activez ou mettez à jour le canal des e-mails d'un projet.

- URI – `/apps/projectId/channels/email`
- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/email`

`mobiletargeting:DeleteGcmChannel`

Supprimer le canal Firebase Cloud Messaging (FCM) d'un projet. Ce canal permet à Amazon Pinpoint d'envoyer des notifications push à une application Android via le service FCM, qui remplace le service Google Cloud Messaging (GCM).

- URI – `/apps/projectId/channels/gcm`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

`mobiletargeting:GetGcmChannel`

Récupérer les informations sur le canal FCM d'un projet. Ce canal permet à Amazon Pinpoint d'envoyer des notifications push à une application Android via le service FCM, qui remplace le service Google Cloud Messaging (GCM).

- URI – `/apps/projectId/channels/gcm`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

mobiletargeting:UpdateGcmChannel

Activer ou mettre à jour le canal FCM d'un projet. Ce canal permet à Amazon Pinpoint d'envoyer des notifications push à une application Android via le service FCM, qui remplace le service Google Cloud Messaging (GCM).

- URI – /apps/*projectId*/channels/gcm
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:DeleteSmsChannel

Désactive le canal SMS d'un projet.

- URI – /apps/*projectId*/channels/sms
- Méthode – DELETE
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetSmsChannel

Récupérez des informations sur le canal des SMS d'un projet.

- URI – /apps/*projectId*/channels/sms
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:UpdateSmsChannel

Activez ou mettez à jour le canal SMS d'un projet.

- URI – /apps/*projectId*/channels/sms
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetChannels

Récupère des informations sur l'historique et le statut de chaque canal pour une application.

- URI – /apps/*application-id*/channels
- Méthode – GET

- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels`

`mobiletargeting:DeleteVoiceChannel`

Désactive le canal vocal d'une application et supprime tous les paramètres existants pour le canal.

- URI – `/apps/application-id/channels/voice`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

`mobiletargeting:GetVoiceChannel`

Récupère des informations sur le statut et les paramètres du canal vocal pour une application.

- URI – `/apps/application-id/channels/voice`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

`mobiletargeting:UpdateVoiceChannel`

Active le canal vocal pour une application ou met à jour l'état et les paramètres du canal vocal pour une application.

- URI – `/apps/application-id/channels/voice`
- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

Points de terminaison

Les autorisations suivantes sont liées à la gestion des points de terminaison dans votre compte Amazon Pinpoint. Dans Amazon Pinpoint, un point de terminaison est une seule destination pour vos messages. Par exemple, un point de terminaison peut être l'adresse e-mail, le numéro de téléphone ou le jeton d'appareil mobile d'un client.

`mobiletargeting:DeleteEndpoint`

Supprime un point de terminaison.

- URI – /apps/*projectId*/endpoints/*endpointId*
- Méthode – DELETE
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:GetEndpoint

Récupère des informations à propos d'un point de terminaison spécifique.

- URI – /apps/*projectId*/endpoints/*endpointId*
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:RemoveAttributes

Supprime un ou plusieurs attributs, du même type d'attribut, de tous les points de terminaison associés à une application.

- URI – apps/*application-id*/attributes/*attribute-type*
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/attributes/*attribute-type*

mobiletargeting:UpdateEndpoint

Crée un point de terminaison ou met à jour les informations pour un point de terminaison.

- URI – /apps/*projectId*/endpoints/*endpointId*
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:UpdateEndpointsBatch

Crée ou met à jour les points de terminaison sous la forme d'une opération par lot.

- URI – /apps/*projectId*/endpoints
- Méthode – PUT

- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Flux d'événement

Les autorisations suivantes sont liées à la gestion des flux d'événements pour votre compte Amazon Pinpoint.

`mobiletargeting:DeleteEventStream`

Supprimez le flux d'événements d'un projet.

- URI – /apps/*projectId*/eventstream/
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

`mobiletargeting:GetEventStream`

Récupérez des informations sur le flux d'événements d'un projet.

- URI – /apps/*projectId*/eventstream/
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

`mobiletargeting:PutEventStream`

Créez ou mettez à jour un flux d'événement pour un projet.

- URI – /apps/*projectId*/eventstream/
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

Événements

Les autorisations suivantes sont liées à la gestion des tâches d'événements dans votre compte Amazon Pinpoint. Dans Amazon Pinpoint, vous créez des tâches d'importation pour créer des segments en fonction des définitions de point de terminaison stockées dans un compartiment Amazon S3.

`mobiletargeting:PutEvents`

Crée un nouvel événement à enregistrer pour les points de terminaison, ou crée ou met à jour les données de point de terminaison auxquelles les événements existants sont associés.

- URI – `/apps/application-id/events`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/events`

Tâches d'exportation

Les autorisations suivantes sont liées à la gestion des tâches d'exportation dans votre compte Amazon Pinpoint. Dans Amazon Pinpoint, vous créez des tâches d'exportation pour envoyer des informations sur les points de terminaison à un compartiment Amazon S3 en vue d'un stockage ou d'une analyse.

`mobiletargeting:CreateExportJob`

Créez une tâche d'exportation pour exporter les définitions de point de terminaison pour Amazon S3.

- URI – `/apps/projectId/jobs/export`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

`mobiletargeting:GetExportJob`

Récupérez des informations sur une tâche d'exportation spécifique d'un projet.

- URI – `/apps/projectId/jobs/export/jobId`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export/jobId`

`mobiletargeting:GetExportJobs`

Récupérez une liste de toutes les tâches d'exportation d'un projet.

- URI – /apps/*projectId*/jobs/export
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/jobs/export

Tâches d'importation

Les autorisations suivantes sont liées à la gestion des tâches d'importation dans votre compte Amazon Pinpoint. Dans Amazon Pinpoint, vous créez des tâches d'importation pour créer des segments en fonction des définitions de point de terminaison stockées dans un compartiment Amazon S3.

mobiletargeting:CreateImportJob

Importer les définitions de point de terminaison à partir d'Amazon S3 pour créer un segment.

- URI – /apps/*projectId*/jobs/import
- Méthode – POST
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetImportJob

Récupérez des informations sur une tâche d'importation spécifique d'un projet.

- URI – /apps/*projectId*/jobs/import/*jobId*
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/jobs/import/*jobId*

mobiletargeting:GetImportJobs

Récupérez des informations sur toutes les tâches d'importation d'un projet.

- URI – /apps/*projectId*/jobs/import
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

Parcours

Les autorisations suivantes sont liées à la gestion des parcours dans votre compte Amazon Pinpoint.

`mobiletargeting:CreateJourney`

Créez un parcours pour un projet.

- URI – `/apps/projectId/journeys`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

`mobiletargeting:GetJourney`

Récupère des informations sur un parcours spécifique.

- URI – `/apps/projectId/journeys/journeyId`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

`mobiletargeting:ListJourneys`

Récupérez des informations sur les parcours d'un projet.

- URI – `/apps/projectId/journeys`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

`mobiletargeting:UpdateJourney`

Mettez à jour la configuration et les autres paramètres pour un parcours spécifique.

- URI – `/apps/projectId/journeys/journeyId`
- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

`mobiletargeting:UpdateJourneyState`

Annulez un parcours actif.

- URI – `/apps/projectId/journeys/journeyId/state`
- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/state`

`mobiletargeting:DeleteJourney`

Supprime un parcours spécifique.

- URI – `/apps/projectId/journeys/journeyId`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

Modèles de messages

Les autorisations suivantes sont liées à la création et à la gestion de modèles de message pour votre compte Amazon Pinpoint. Un modèle de message est un ensemble de contenu que vous pouvez créer, enregistrer, puis réutiliser dans les messages que vous envoyez pour l'un de vos projets Amazon Pinpoint.

`mobiletargeting:ListTemplates`

Récupérer les informations sur tous les modèles de message associés à votre compte Amazon Pinpoint.

- URI – `/templates`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates`

`mobiletargeting:ListTemplateVersions`

Récupère des informations sur toutes les versions d'un modèle de message spécifique.

- URI – /templates/*template-name*/*template-type*/versions
- Méthode – GET
- ARN des ressources – Sans objet

mobiletargeting:UpdateTemplateActiveVersion

Désignez une version spécifique d'un modèle de message comme version active du modèle.

- URI – /templates/*template-name*/*template-type*/active-version
- Méthode – GET
- ARN des ressources – Sans objet

mobiletargeting:GetEmailTemplate

Récupérez les informations sur un modèle de message pour les messages envoyés par le canal de messagerie.

- URI – /templates/*template-name*/email
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:CreateEmailTemplate

Créez un modèle de message pour les messages envoyés par le canal de messagerie.

- URI – /templates/*template-name*/email
- Méthode – POST
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:UpdateEmailTemplate

Mettez à jour un modèle de message existant pour les messages envoyés par le canal de messagerie.

- URI – /templates/*template-name*/email

- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:DeleteEmailTemplate`

Supprimez un modèle de message pour les messages envoyés par le canal de messagerie.

- URI – `/templates/template-name/email`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:GetPushTemplate`

Récupérez les informations sur un modèle de message pour les messages envoyés via un canal de notification push.

- URI – `/templates/template-name/push`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:CreatePushTemplate`

Créez un modèle de message pour les messages envoyés via un canal de notification push.

- URI – `/templates/template-name/push`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:UpdatePushTemplate`

Mettez à jour un modèle de message existant pour les messages envoyés via un canal de notification push.

- URI – `/templates/template-name/push`

- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:DeletePushTemplate`

Supprimez un modèle de message pour les messages envoyés via un canal de notification push.

- URI – `/templates/template-name/push`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:GetSmsTemplate`

Récupérez les informations sur un modèle de message pour les messages envoyés par le canal SMS.

- URI – `/templates/template-name/sms`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:CreateSmsTemplate`

Créez un modèle de message pour les messages envoyés par le canal SMS.

- URI – `/templates/template-name/sms`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:UpdateSmsTemplate`

Mettez à jour un modèle de message existant pour les messages envoyés par le canal SMS.

- URI – `/templates/template-name/sms`
- Méthode – PUT

- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:DeleteSmsTemplate`

Supprimez un modèle de message pour les messages envoyés par le canal SMS.

- URI – `/templates/template-name/sms`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:GetVoiceTemplate`

Récupérez les informations sur un modèle de message pour les messages envoyés par le canal vocal.

- URI – `/templates/template-name/voice`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

`mobiletargeting:CreateVoiceTemplate`

Créez un modèle de message pour les messages envoyés par le canal vocal.

- URI – `/templates/template-name/voice`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

`mobiletargeting:UpdateVoiceTemplate`

Mettez à jour un modèle de message existant pour les messages envoyés par le canal vocal.

- URI – `/templates/template-name/voice`
- Méthode – PUT
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

`mobiletargeting:DeleteVoiceTemplate`

Supprimez un modèle de message pour les messages envoyés par le canal vocal.

- URI – /templates/*template-name*/voice
- Méthode – DELETE
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

Messages

Les autorisations suivantes sont liées à l'envoi de messages SMS et de notifications push à partir de votre compte Amazon Pinpoint. Vous pouvez utiliser les opérations SendUsersMessages et SendMessages pour envoyer des messages à des points de terminaison spécifiques sans créer au préalable des segments et des campagnes.

mobiletargeting:SendMessages

Envoie un message SMS ou une notification push à des points de terminaison spécifiques.

- URI – /apps/*projectId*/messages
- Méthode – POST
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/messages

mobiletargeting:SendUsersMessages

Envoyez un SMS ou une notification push à tous les points de terminaison associés à un ID utilisateur spécifique.

- URI – /apps/*projectId*/users-messages
- Méthode – POST
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/messages

One-time mots de passe

Les autorisations suivantes sont liées à l'envoi et à la validation de mots de passe à usage unique (OTP) dans Amazon Pinpoint.

mobiletargeting:SendOTPMessage

Envoyer un message texte contenant un mot de passe à usage unique.

- URI – /apps/*projectId*/otp
- Méthode – POST
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/otp

mobiletargeting:VerifyOTPMessage

Vérifier la validité d'un mot de passe unique (OTP) généré à l'aide de l'opération SendOTPMessage.

- URI – /apps/*projectId*/verify-otp
- Méthode – POST
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/verify-otp

Validation du numéro de téléphone

Les autorisations suivantes sont liées à l'utilisation du service de validation du numéro de téléphone dans Amazon Pinpoint.

mobiletargeting:PhoneNumberValidate

Récupérez des informations sur un numéro de téléphone.

- URI – /phone/number/validate
- Méthode – POST
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:phone/number/validate

Projets

Les autorisations suivantes sont liées à la gestion des projets dans votre compte Amazon Pinpoint. Initialement, les projets étaient appelées applications. Dans le cadre de ces opérations, une application Amazon Pinpoint est identique à un projet Amazon Pinpoint.

mobiletargeting:CreateApp

Créer un projet Amazon Pinpoint.

- URI – /apps

- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:DeleteApp`

Supprimer un projet Amazon Pinpoint.

- URI – `/apps/projectId`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApp`

Récupérer des informations sur un projet Amazon Pinpoint.

- URI – `/apps/projectId`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApps`

Récupérez des informations sur tous les projets associés à votre compte Amazon Pinpoint.

- URI – `/apps`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:GetApplicationSettings`

Récupérer les paramètres par défaut d'un projet Amazon Pinpoint.

- URI – `/apps/projectId/settings`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:UpdateApplicationSettings`

Mettre à jour les paramètres par défaut d'un projet Amazon Pinpoint.

- URI – `/apps/projectId/settings`
- Méthode – PUT

- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Modèles de recommandation

Les autorisations suivantes sont liées à la gestion des configurations Amazon Pinpoint pour la récupération et le traitement des données de recommandation des modèles de recommandation. Un modèle de recommandation est un type de modèle de machine learning qui prédit et génère des recommandations personnalisées en identifiant des modèles dans les données.

`mobiletargeting:CreateRecommenderConfiguration`

Crée une configuration Amazon Pinpoint pour un modèle de recommandation.

- URI – `/recommenders`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:recommenders`

`mobiletargeting:GetRecommenderConfigurations`

Récupère des informations sur toutes les configurations de modèle de recommandation associées à votre compte Amazon Pinpoint.

- URI – `/recommenders`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:recommenders`

`mobiletargeting:GetRecommenderConfiguration`

Récupère des informations sur une configuration Amazon Pinpoint individuelle pour un modèle de recommandation.

- URI – `/recommenders/recommenderId`
- Méthode – GET
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

`mobiletargeting:UpdateRecommenderConfiguration`

Met à jour une configuration Amazon Pinpoint pour un modèle de recommandation.

- URI – `/recommenders/recommenderId`
- Méthode – PUT

- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

`mobiletargeting:DeleteRecommenderConfiguration`

Supprime une configuration Amazon Pinpoint pour un modèle de recommandation.

- URI – `/recommenders/recommenderId`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

Segments

Les autorisations suivantes sont liées à la gestion des segments dans votre compte Amazon Pinpoint. Dans Amazon Pinpoint, les segments sont des groupes de destinataires pour vos campagnes qui partagent certains attributs que vous définissez.

`mobiletargeting:CreateSegment`

Créez un segment. Pour permettre à un utilisateur de créer un segment en important les données de point de terminaison en dehors d'Amazon Pinpoint, autorisez l'action `mobiletargeting:CreateImportJob`.

- URI – `/apps/projectId/segments`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:DeleteSegment`

Supprimez un segment.

- URI – `/apps/projectId/segments/segmentId`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

`mobiletargeting:GetSegment`

Récupère des informations sur un segment spécifique.

- URI – /apps/*projectId*/segments/*segmentId*
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentExportJobs

Récupérez des informations sur des tâches qui exportent des définitions de point de terminaison pour un segment.

- URI – /apps/*projectId*/segments/*segmentId*/jobs/export
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*/jobs/export

mobiletargeting:GetSegments

Récupérez les informations sur les segments d'un projet.

- URI – /apps/*projectId*/segments
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetSegmentImportJobs

Récupère des informations sur les tâches qui créent des segments en important les définitions de point de terminaison à partir d'Amazon S3.

- URI – /apps/*projectId*/segments/*segmentId*/jobs/import
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersion

Récupère des informations sur une version de segment spécifique.

- URI – /apps/*projectId*/segments/*segmentId*/versions/*versionId*
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersions

Récupère des informations sur les versions actuelles et antérieures d'un segment.

- URI – /apps/*projectId*/segments/*segmentId*/versions
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:UpdateSegment

Met à jour un segment spécifique.

- URI – /apps/*projectId*/segments/*segmentId*
- Méthode – PUT
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

Étiquettes

Les autorisations suivantes sont liées à l'affichage et à la gestion des balises pour les ressources Amazon Pinpoint.

mobiletargeting:ListTagsForResource

Récupérez les informations sur les balises qui sont associées à un projet, une campagne ou un segment.

- URI – /tags/*resource-arn*
- Méthode – GET
- ARN des ressources – arn:aws:mobiletargeting:*region*:*accountId*:*

`mobiletargeting:TagResource`

Ajoutez une ou plusieurs balises à un projet, une campagne, un modèle de message ou un segment.

- URI – `/tags/resource-arn`
- Méthode – POST
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:UntagResource`

Supprimez une ou plusieurs balises d'un projet, d'une campagne, d'un modèle de message ou d'un segment.

- URI – `/tags/resource-arn`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:*`

Utilisateurs

Les autorisations suivantes sont liées à la gestion des utilisateurs. Dans Amazon Pinpoint, les utilisateurs correspondent aux personnes qui reçoivent des messages de votre part. Un utilisateur peut être associé à plusieurs points de terminaison.

`mobiletargeting:DeleteUserEndpoints`

Supprime tous les points de terminaison qui sont associés à un ID utilisateur.

- URI – `/apps/projectId/users/userId`
- Méthode – DELETE
- ARN des ressources – `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

`mobiletargeting:GetUserEndpoints`

Récupérez des informations sur tous les points de terminaison qui sont associés à un ID utilisateur.

- URI – `/apps/projectId/users/userId`
- Méthode – GET

- ARN des ressources – `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:apps/`*`projectId`*`/users/`*`userId`*

Actions de l'API SMS et Voix Amazon Pinpoint Version 1

Cette section identifie les actions pour les fonctions disponibles à partir de l'API SMS et voix Amazon Pinpoint. Il s'agit d'une API supplémentaire qui fournit des options avancées pour l'utilisation et la gestion des canaux SMS et vocaux dans Amazon Pinpoint. Pour en savoir plus sur cette API, consultez la référence de l'API Amazon Pinpoint de messages SMS et vocaux.

`sms-voice:CreateConfigurationSet`

Créez un jeu de configuration pour l'envoi de messages vocaux.

- URI – `/sms-voice/configuration-sets`
- Méthode – POST
- ARN des ressources – Non disponible Utilisez *.

`sms-voice:DeleteConfigurationSet`

Créez un jeu de configuration pour l'envoi de messages vocaux.

- URI — voice/configuration /sms- -sets/ *`ConfigurationSetName`*
- Méthode – DELETE
- ARN des ressources – Non disponible Utilisez *.

`sms-voice:GetConfigurationSetEventDestinations`

Obtenez des informations sur un ensemble de configurations et les destinations d'événement qu'il contient.

- URI — /sms- -sets/ /event-destinations voice/configuration *`ConfigurationSetName`*
- Méthode – GET
- ARN des ressources – Non disponible Utilisez *.

`sms-voice:CreateConfigurationSetEventDestination`

Créez une destination d'événement pour les événements vocaux.

- URI — /sms- -sets/ /event-destinations voice/configuration *`ConfigurationSetName`*

- Méthode – POST
- ARN des ressources – Non disponible Utilisez *.

`sms-voice:UpdateConfigurationSetEventDestination`

Mettez à jour une destination d'événement pour les événements vocaux.

- URI — /sms- -sets/ voice/configuration /event-destinations/ *`ConfigurationSetName`* *`EventDestinationName`*
- Méthode – PUT
- ARN des ressources – Non disponible Utilisez *.

`sms-voice:DeleteConfigurationSetEventDestination`

Supprimez une destination d'événement pour les événements vocaux.

- URI — /sms- -sets/ voice/configuration /event-destinations/ *`ConfigurationSetName`* *`EventDestinationName`*
- Méthode – DELETE
- ARN des ressources – Non disponible Utilisez *.

`sms-voice:SendVoiceMessage`

Créez et envoyez des messages vocaux.

- URI — /sms - /message voice/voice
- Méthode – POST
- ARN des ressources – Non disponible Utilisez *.

Exemples de politique basée sur l'identité d'Amazon Pinpoint

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources Amazon Pinpoint. Ils ne peuvent pas non plus effectuer de tâches à l'aide de AWS Management Console AWS CLI, ou d'une AWS API. Un administrateur IAM doit créer des politiques IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources dont ils ont besoin. Il doit ensuite attacher ces stratégies aux utilisateurs ou aux groupes ayant besoin de ces autorisations.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, veuillez consulter Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console Amazon Pinpoint
- Exemple : accès à un seul projet Amazon Pinpoint
- Exemple : affichage des ressources Amazon Pinpoint en fonction des balises
- Exemple : autoriser des utilisateurs à afficher leurs propres autorisations
- Exemples : octroi de l'accès aux actions d'API Amazon Pinpoint
- Exemples : octroi de l'accès aux actions d'API SMS et voix Amazon Pinpoint
- Exemple : restriction de l'accès d'un projet Amazon Pinpoint à des adresses IP spécifiques
- Exemple : restriction de l'accès Amazon Pinpoint en fonction des balises
- Exemple : autoriser Amazon Pinpoint à envoyer des e-mails en utilisant des identités vérifiées dans Amazon SES

Bonnes pratiques en matière de politiques

Identity-based les politiques déterminent si quelqu'un peut créer, accéder ou supprimer des ressources Amazon Pinpoint 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 Amazon Pinpoint

Pour accéder à la console Amazon Pinpoint, vous devez disposer d'un ensemble minimum d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources Amazon Pinpoint de votre AWS compte. Si vous créez une stratégie basée sur l'identité qui applique des autorisations plus restrictives que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs et rôles) tributaires de cette stratégie. Pour vous assurer que ces entités peuvent utiliser la console Amazon Pinpoint, attachez une stratégie aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

L'exemple de politique suivant fournit un accès en lecture seule à la console Amazon Pinpoint dans une région spécifique. AWS Il inclut un accès en lecture seule à d'autres services dont dépend la console Amazon Pinpoint, tels qu'Amazon Simple Email Service (Amazon SES), IAM et Amazon Kinesis.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "UseConsole",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "kinesis:ListStreams",
                "s3:List*",
                "ses:Describe*",
                "ses:Get*",
                "ses:List*",
                "sns:ListTopics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

Dans l'exemple de politique précédent, remplacez *region* par le nom d'une AWS région et remplacez *accountId* par votre identifiant de AWS compte.

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.

Exemple : accès à un seul projet Amazon Pinpoint

Vous pouvez également créer des stratégies en lecture seule qui permettent d'accéder uniquement à des projets spécifiques. L'exemple de stratégie suivant permet à des utilisateurs de se connecter à la console et d'afficher une liste des projets. Il permet également aux utilisateurs d'afficher des informations sur les ressources liées à d'autres services AWS dont dépend la console Amazon Pinpoint, tels que Amazon SES, IAM et Amazon Kinesis. Toutefois, la stratégie permet aux utilisateurs d'afficher des informations supplémentaires uniquement sur le projet spécifié dans la stratégie. Vous pouvez modifier cette politique pour autoriser l'accès à d'autres projets ou AWS régions.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:reports"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

Dans l'exemple précédent, remplacez *`region`* par le nom d'une AWS région, remplacez *`accountId`* par votre identifiant de AWS compte et remplacez *`projectId`* par l'ID du projet Amazon Pinpoint auquel vous souhaitez donner accès.

De même, vous pouvez créer des politiques qui accordent à un utilisateur de votre AWS compte un accès en écriture limité à l'un de vos projets Amazon Pinpoint, par exemple le projet qui possède l'ID du `810c7aab86d42fb2b56c8c966example` projet. Dans ce cas, vous souhaitez autoriser l'utilisateur à afficher, ajouter et mettre à jour des composants de projet, tels que des segments et des campagnes, mais pas à supprimer des composants.

Outre l'octroi d'autorisations pour les actions `mobiletargeting:Get` et `mobiletargeting:List`, créez une stratégie qui accorde des autorisations pour les actions suivantes : `mobiletargeting:Create`, `mobiletargeting:Update` et `mobiletargeting:Put`. Il s'agit des autorisations supplémentaires requises pour créer et gérer la plupart des composants du projet. Par exemple :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LimitedWriteProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*",
                "mobiletargeting:Create*",
                "mobiletargeting:Update*",
                "mobiletargeting:Put*"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/810c7aab86d42fb2b56c8c966example",
    "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/810c7aab86d42fb2b56c8c966example/*",
                "arn:aws:mobiletargeting:us-east-1:111122223333:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Exemple : affichage des ressources Amazon Pinpoint en fonction des balises

Vous pouvez utiliser des conditions dans une stratégie basée sur l'identité pour contrôler l'accès aux ressources Amazon Pinpoint en fonction des balises. Cet exemple de stratégie montre comment vous pouvez créer ce type de stratégie pour autoriser l'affichage des ressources Amazon Pinpoint. Toutefois, l'autorisation est accordée uniquement si la balise de ressource `Owner` a la valeur du nom d'utilisateur de cet utilisateur. Cette politique accorde également les autorisations nécessaires pour réaliser cette action sur la console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewResourceIfOwner",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:*:*:*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
```

```
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:*"
                }
            }
        }
    ]
}
```

Vous pouvez attacher ce type de stratégie aux utilisateurs de votre compte. Si un utilisateur nommé `richard-roe` tente d'afficher une ressource Amazon Pinpoint, la ressource doit être balisée `0wner=richard-roe` ou `owner=richard-roe`. Dans le cas contraire, l'utilisateur se voit refuser l'accès. La clé de condition d'étiquette `0wner` correspond à la fois à `0wner` et à `owner`, car les noms de clé de condition ne sont pas sensibles à la casse. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemple : autoriser des utilisateurs à 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": "*"
        }
    ]
}
```

Exemples : octroi de l'accès aux actions d'API Amazon Pinpoint

Cette section fournit des exemples de stratégie permettant d'accéder aux fonctionnalités disponibles à partir de l'API Amazon Pinpoint, API principale pour Amazon Pinpoint. Pour en savoir plus sur cette API, consultez la Référence de l'API Amazon Pinpoint.

Read-only accès

L'exemple de politique suivant autorise l'accès en lecture seule à toutes les ressources de votre compte Amazon Pinpoint dans une région spécifique. AWS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewAllResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
```

```
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        }
    ]
}
```

Dans l'exemple précédent, remplacez *region* par le nom d'une AWS région et remplacez *accountId* par votre identifiant de AWS compte.

Accès administrateur

L'exemple de stratégie suivant autorise un accès complet à toutes les actions et ressources Amazon Pinpoint de votre compte Amazon Pinpoint :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        }
    ]
}
```

Dans l'exemple précédent, remplacez *accountId* par votre identifiant de AWS compte.

Exemples : octroi de l'accès aux actions d'API SMS et voix Amazon Pinpoint

Cette section fournit des exemples de stratégie permettant d'accéder aux fonctionnalités disponibles à partir de l'API SMS et voix Amazon Pinpoint. Il s'agit d'une API supplémentaire qui fournit des options avancées pour l'utilisation et la gestion des canaux SMS et vocaux dans Amazon Pinpoint. Pour en savoir plus sur cette API, consultez la référence de l'API Amazon Pinpoint de messages SMS et vocaux.

Read-only accès

L'exemple de politique suivant autorise l'accès en lecture seule à toutes les actions et ressources de l'API Amazon Pinpoint SMS et Voice de votre compte : AWS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SMSVoiceReadOnly",
            "Effect": "Allow",
            "Action": [
                "sms-voice:Get*",
                "sms-voice:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:us-east-1:111122223333:*"
                }
            }
        }
    ]
}
```

Accès administrateur

L'exemple de politique suivant permet un accès complet à toutes les actions et ressources de l'API Amazon Pinpoint SMS et Voice de votre AWS compte :

JSON

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

```
        {
            "Sid": "SMSVoiceFullAccess",
            "Effect": "Allow",
            "Action": [
                "sms-voice:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:us-east-1:111122223333:*"
                }
            }
        }
    ]
}
```

Exemple : restriction de l'accès d'un projet Amazon Pinpoint à des adresses IP spécifiques

L'exemple de politique suivant accorde à tout utilisateur l'autorisation d'effectuer une action Amazon Pinpoint sur un projet spécifique ()*projectId*. Toutefois, la demande doit provenir de la plage d'adresses IP spécifiée dans la condition.

La condition dans cette instruction identifie la plage `54.240.143.*` d'adresses Internet Protocol version 4 (IPv4) autorisées, avec une exception : `54.240.143.188`. Le `Condition` bloc utilise les `NotIpAddress` conditions `IpAddress` et et la clé de `aws:SourceIp` condition, qui est une clé AWS de condition étendue. Pour plus d'informations sur ces clés de condition, consultez Spécification des conditions dans une stratégie dans le Guide de l'utilisateur IAM. Les valeurs IPv4 `aws:SourceIp` font appel à la notation CIDR standard. Pour en savoir plus, consultez Opérateurs de condition d'adresse IP dans le guide de l'utilisateur IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Id":"AMZPinpointPolicyId1",
    "Statement":[
```

```
        {
            "Sid":"IPAllow",
            "Effect":"Allow",
            "Principal":"*",
            "Action":"mobiletargeting:*",
            "Resource":[
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*"
            ],
            "Condition":{
                "IpAddress":{
                    "aws:SourceIp":"54.240.143.0/24"
                },
                "NotIpAddress":{
                    "aws:SourceIp":"54.240.143.188/32"
                }
            }
        }
    ]
}
```

Exemple : restriction de l'accès Amazon Pinpoint en fonction des balises

L'exemple de politique suivant accorde des autorisations pour effectuer n'importe quelle action Amazon Pinpoint sur un projet spécifique ()*projectId*. Toutefois, les autorisations ne sont accordées que si la demande provient d'un utilisateur dont le nom est une valeur dans la balise de ressource `Owner` pour le projet, comme spécifié dans la condition.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ModifyResourceIfOwner",
            "Effect": "Allow",
            "Action": "mobiletargeting:*",
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                }
            }
        }
    ]
}
```

Exemple : autoriser Amazon Pinpoint à envoyer des e-mails en utilisant des identités vérifiées dans Amazon SES

Lorsque vous vérifiez une identité e-mail (telle qu'une adresse e-mail ou un domaine) via la console Amazon Pinpoint, cette identité est automatiquement configurée afin qu'elle puisse être utilisée à la fois par Amazon Pinpoint et Amazon SES. Toutefois, si vous vérifiez une identité e-mail via Amazon SES et que vous souhaitez utiliser cette identité avec Amazon Pinpoint, vous devez appliquer une politique à cette identité.

L'exemple de politique suivant accorde à Amazon Pinpoint l'autorisation d'envoyer des e-mails en utilisant une identité e-mail vérifiée via Amazon SES.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws:ses:us-east-1:111122223333:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"111122223333"
                },
                "ArnLike":{
```

```
                    "aws:SourceArn":"arn:aws:mobiletargeting:us-
east-1:111122223333:apps/*"
                }
            }
        }
    ]
}
```

Si vous utilisez Amazon Pinpoint dans la région AWS GovCloud (US-West), utilisez plutôt l'exemple de politique suivant :

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws-us-gov:ses:us-gov-
west-1:111122223333:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"111122223333"
                },
                "ArnLike":{
                    "aws:SourceArn":"arn:aws-us-gov:mobiletargeting:us-gov-
west-1:111122223333:apps/*"
                }
            }
        }
    ]
}
```

Rôles IAM pour les tâches Amazon Pinpoint courantes

Un rôle IAM est une identité Gestion des identités et des accès AWS (IAM) que vous pouvez créer dans votre AWS compte et accorder des autorisations spécifiques. Un rôle IAM est une AWS identité dotée de politiques d'autorisation qui déterminent ce que l'identité peut et ne peut pas y faire. AWS En revanche, au lieu d'être associé de manière unique à une personne, un rôle est conçu pour être assumé par tout utilisateur qui en a besoin.

En outre, aucune information d'identification à long terme standard n'est associée à celui-ci. Au lieu de cela, il fournit des informations d'identification de sécurité temporaires pour une session. Vous pouvez utiliser les rôles IAM pour déléguer l'accès à des utilisateurs, à des applications, à des applications ou à des services qui n'ont normalement pas accès à vos AWS ressources.

Pour ces raisons, vous pouvez utiliser des rôles IAM pour intégrer Amazon Pinpoint à certains services et ressources AWS pour votre compte. Par exemple, vous pouvez autoriser Amazon Pinpoint à accéder aux définitions de point de terminaison que vous stockez dans un compartiment Amazon Simple Storage Service (Amazon S3) et que vous souhaitez utiliser pour les segments. Vous pouvez également autoriser Amazon Pinpoint à diffuser des données d'événement vers un flux Amazon Kinesis pour votre compte. De même, vous pouvez utiliser les rôles IAM pour permettre aux applications Web ou mobiles d'enregistrer des points de terminaison ou de signaler des données d'utilisation pour des projets Amazon Pinpoint, sans AWS intégrer de clés dans les applications (où elles peuvent être difficiles à faire pivoter et les utilisateurs peuvent éventuellement les extraire).

Pour ces scénarios, vous pouvez déléguer l'accès à Amazon Pinpoint via les rôles IAM. Cette section explique et fournit des exemples de tâches Amazon Pinpoint courantes qui utilisent des rôles IAM pour travailler avec d'autres services AWS . Pour plus d'informations sur l'utilisation des rôles IAM avec des applications Web et mobiles, consultez Octroi d'accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.

Rubriques

- Rôle IAM pour l'importation de points de terminaison ou de segments
- Rôle IAM pour l'exportation de points de terminaison ou de segments
- Rôle IAM pour récupérer les recommandations d'Amazon Personalize
- Rôle IAM pour la diffusion des événements vers Kinesis
- Rôle IAM pour l'envoi d'e-mails avec Amazon SES

Rôle IAM pour l'importation de points de terminaison ou de segments

Avec Amazon Pinpoint, vous pouvez définir un segment d'utilisateurs en important les définitions des points de terminaison depuis un compartiment Amazon Simple Storage Service (Amazon S3) de votre compte. AWS Avant d'importer, vous devez déléguer les autorisations requises à Amazon Pinpoint. Pour ce faire, vous créez un rôle Gestion des identités et des accès AWS (IAM) et associez les politiques suivantes au rôle :

- La politique gérée `AmazonS3ReadOnlyAccess` AWS . Cette politique est créée et gérée par AWS, et elle accorde un accès en lecture seule à votre compartiment Amazon S3.
- Une stratégie d'approbation qui permet à Amazon Pinpoint d'assumer le rôle.

Après avoir créé le rôle, vous pouvez utiliser Amazon Pinpoint pour importer des segments à partir d'un compartiment Amazon S3. Pour plus d'informations sur la création du compartiment, la création de fichiers de point de terminaison et l'importation d'un segment à l'aide de la console, consultez Importation de segments dans le Guide de l'utilisateur Amazon Pinpoint. Pour un exemple d'importation d'un segment par programmation à l'aide du AWS SDK pour Java, consultez ce Importer des segments dans Amazon Pinpoint guide.

Création du rôle IAM (AWS CLI)

Procédez comme suit pour créer le rôle IAM à l'aide du AWS Command Line Interface (AWS CLI). Si vous ne l'avez pas encore installé AWS CLI, consultez la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour créer le rôle IAM à l'aide du AWS CLI

1. Créez un fichier JSON qui contient la stratégie d'approbation de votre rôle et enregistrez le fichier en local. Vous pouvez utiliser la politique de confiance suivante.

   JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
          {
               "Action": "sts:AssumeRole",
               "Effect": "Allow",
```

```
                "Principal": {
                     "Service": "pinpoint.amazonaws.com"
                },
                "Condition": {
                     "StringEquals": {
                          "aws:SourceAccount": "111122223333"
                     },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/application-id"
                     }
                }
           }
      ]
}
```

Dans l'exemple précédent, procédez comme suit :

- Remplacez *region* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
- *accountId*Remplacez-le par l'identifiant unique de votre AWS compte.
- Remplacez *application-id* par l'identifiant unique du projet.

2. Sur la ligne de commande, utilisez `create-role` pour créer le rôle et attachez la stratégie d'approbation :

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document
 file://PinpointImportTrustPolicy.json
```

Après le préfixe `file://`, spécifiez le chemin d'accès au fichier JSON qui contient la politique d'approbation.

Après avoir exécuté cette commande, vous voyez une sortie similaire à la suivante dans votre terminal :

3. Utilisez la `attach-role-policy`commande pour associer la politique `AmazonS3ReadOnlyAccess` AWS gérée au rôle :

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport
```

Rôle IAM pour l'exportation de points de terminaison ou de segments

Vous pouvez obtenir une liste des points de terminaison en créant une tâche d'exportation. Lorsque vous créez une tâche d'exportation, vous devez spécifier un ID de projet et vous pouvez, si vous le souhaitez, spécifier un ID de segment. Amazon Pinpoint exporte ensuite la liste des points de terminaison associés au projet ou au segment vers un compartiment Amazon Simple Storage Service (Amazon S3). Le fichier obtenu contient une JSON-formatted liste de points de terminaison et de leurs attributs, tels que le canal, l'adresse, le statut de désinscription, in/opt la date de création et l'ID du point de terminaison.

Pour créer une tâche d'exportation, vous devez configurer un rôle IAM qui autorise Amazon Pinpoint à écrire dans un compartiment Amazon S3. Le processus de configuration du rôle se compose de deux étapes :

1. Créez une politique IAM qui autorise une entité (dans le cas présent, Amazon Pinpoint) à écrire dans un compartiment Amazon S3 spécifique.
2. Créez un rôle IAM et attachez-lui la stratégie.

Cette rubrique contient les procédures de réalisation de ces deux étapes. Ces procédures supposent que vous avez déjà créé un compartiment Amazon S3 et un dossier au sein de ce compartiment pour stocker les segments exportés. Pour plus d'informations sur la création de compartiments, consultez Création d'un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service.

Ces procédures supposent aussi que vous avez déjà installé et configuré l' AWS Command Line Interface (AWS CLI). Pour plus d'informations sur la configuration du AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Étape 1 : créer la politique IAM

Une politique IAM définit les autorisations d'une entité, telle qu'une identité ou une ressource. Pour créer un rôle pour exporter les points de terminaison Amazon Pinpoint, vous devez créer une politique qui accorde l'autorisation d'écrire dans un dossier spécifique d'un compartiment Amazon S3 donné. L'exemple de stratégie suivant respecte la pratique en matière de sécurité d'octroi du moindre privilège, autrement dit elle n'accorde que les autorisations requises pour exécuter une seule tâche.

Pour créer la stratégie IAM

1. Dans un éditeur de texte, créez un fichier. Collez le code suivant dans le fichier :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::*" ]
        },
        {
            "Sid": "AllowRootAndHomeListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket" ],
            "Condition": {
                "StringEquals": {
                    "s3:delimiter": [ "/" ],
                    "s3:prefix": [
                        "",
                        "Exports/"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket" ],
            "Condition": {
                "StringLike": {
```

```
                    "s3:prefix": [
                        "Exports/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
            "Action": [ "s3:*" ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket/Exports/*" ]
        }
    ]
}
```

Dans le code précédent, remplacez toutes les instances de *amzn-s3-demo-bucket-example-bucket* par le nom du compartiment Amazon S3 contenant le dossier dans lequel vous souhaitez exporter les informations du segment. Remplacez également toutes les instances *Exports* de par le nom du dossier lui-même.

Lorsque vous avez terminé, enregistrez le fichier sous `s3policy.json`.

2. À l'aide de AWS CLI, naviguez jusqu'au répertoire dans lequel se trouve le `s3policy.json` fichier. Ensuite, entrez la commande suivante pour créer la stratégie :

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
 file://s3policy.json
```

Si la stratégie a été créée avec succès, vous obtenez une sortie similaire à ce qui suit :

```
{
    "Policy": {
        "CreateDate": "2018-04-11T18:44:34.805Z",
        "IsAttachable": true,
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PolicyId": "ANPAJ2YJQRJCG3EXAMPLE",
        "UpdateDate": "2018-04-11T18:44:34.805Z",
        "Arn": "arn:aws:iam::123456789012:policy/s3ExportPolicy",
        "PolicyName": "s3ExportPolicy",
```

```
        "Path": "/"
    }
}
```

Copiez l'Amazon Resource Name (ARN) de la stratégie (`arn:aws:iam::123456789012:policy/s3ExportPolicy` dans l'exemple précédent). Dans la section suivante, vous devez fournir cet ARN lorsque vous créez le rôle.

> **Note**
>
> Si vous voyez un message indiquant que votre compte n'est pas autorisé à effectuer l'opération `CreatePolicy`, vous devez attacher une politique à votre utilisateur vous permettant de créer des politiques et des rôles IAM. Pour plus d'informations, consultez Ajout et suppression d'autorisations basées sur l'identité IAM dans le Guide de l'utilisateur IAM.

Étape 2 : création du rôle IAM

Maintenant que vous avez créé une politique IAM, vous pouvez créer un rôle et lui attacher la politique. Chaque rôle IAM contient une stratégie d'approbation, à savoir un ensemble de règles qui spécifie les entités autorisées à endosser le rôle. Dans cette section, vous allez créer une stratégie d'approbation qui autorise Amazon Pinpoint à assumer le rôle. Ensuite, vous créez le rôle lui-même, puis attachez la stratégie que vous avez créée dans la section précédente.

Pour créer le rôle IAM

1. Dans un éditeur de texte, créez un fichier. Collez le code suivant dans le fichier :

   JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
```

```
            "Action":"sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/applicationId"
                }
            }
        }
    ]
}
```

Enregistrez le fichier sous le nom `trustpolicy.json`.

2. À l'aide de AWS CLI, naviguez jusqu'au répertoire dans lequel se trouve le `trustpolicy.json` fichier. Entrez la commande suivante pour créer un nouveau rôle :

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
 file://trustpolicy.json
```

3. Sur la ligne de commande, entrez la commande suivante pour attacher la stratégie que vous avez créée dans la section précédente au rôle que vous venez de créer :

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
s3ExportPolicy --role-name s3ExportRole
```

Dans la commande précédente, remplacez *arn:aws:iam::123456789012:policy/s3ExportPolicy* par l'ARN de la politique que vous avez créée dans la section précédente.

Rôle IAM pour récupérer les recommandations d'Amazon Personalize

Vous pouvez configurer Amazon Pinpoint pour récupérer les données de recommandation à partir d'une solution Amazon Personalize déployée en tant que campagne Amazon Personalize. Vous pouvez utiliser ces données pour envoyer des recommandations personnalisées aux destinataires de messages en fonction des attributs et du comportement de chaque destinataire. Pour en savoir plus, consultez les modèles de machine learning dans le guide de l'utilisateur Amazon Pinpoint.

Avant de pouvoir récupérer les données de recommandation d'une campagne Amazon Personalize, vous devez créer un rôle Gestion des identités et des accès AWS (IAM) qui autorise Amazon Pinpoint à récupérer les données de la campagne. Amazon Pinpoint peut créer ce rôle automatiquement pour vous lorsque vous utilisez la console pour configurer un modèle de recommandation dans Amazon Pinpoint. Vous pouvez également créer ce rôle manuellement.

Pour créer le rôle manuellement, utilisez l'API IAM pour effectuer les étapes suivantes :

1. Créez une politique IAM qui autorise une entité (dans ce cas, Amazon Pinpoint) à récupérer les données de recommandation d'une campagne Amazon Personalize.
2. Créez un rôle IAM et attachez-lui la politique IAM.

Cette rubrique explique comment effectuer ces étapes à l'aide du AWS Command Line Interface (AWS CLI). Cela suppose la création préalable de la solution Amazon Personalize et son déploiement en tant que campagne Amazon Personalize. Pour plus d'informations sur la création et le déploiement d'une campagne, consultez Création d'une campagne dans le Guide du développeur Amazon Personalize.

Cette rubrique suppose que vous avez déjà installé et configuré l' AWS CLI. Pour plus d'informations sur la configuration du AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Étape 1 : créer la politique IAM

Une politique IAM définit les autorisations d'une entité, telle qu'une identité ou une ressource. Pour créer un rôle autorisant Amazon Pinpoint à récupérer les données de recommandation d'une campagne Amazon Personalize, vous devez d'abord créer une politique IAM pour le rôle. Cette politique doit permettre à Amazon Pinpoint de :

- récupérer les informations de configuration de la solution déployée par la campagne (`DescribeSolution`) ;
- vérifiez l'état de la campagne (`DescribeCampaign`) ;
- récupérer les données de recommandation de la campagne (`GetRecommendations`).

Dans la procédure suivante, l'exemple de politique autorise cet accès pour une solution Amazon Personalize particulière qui a été déployée par une campagne Amazon Personalize particulière.

Pour créer la stratégie IAM

1. Dans un éditeur de texte, créez un fichier. Collez le code suivant dans le fichier :

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "RetrieveRecommendationsOneCampaign",
            "Effect": "Allow",
            "Action": [
                "personalize:DescribeSolution",
                "personalize:DescribeCampaign",
                "personalize:GetRecommendations"
            ],
            "Resource": [
                "arn:aws:personalize:us-
east-1:111122223333:solution/solutionId",
                "arn:aws:personalize:us-
east-1:111122223333:campaign/campaignId"
            ]
        }
    ]
}
```

   Dans l'exemple précédent, remplacez le *italicized* texte par vos informations :

   - *region*— Le nom de la AWS région qui héberge la solution et la campagne Amazon Personalize.
   - *accountId*— Votre Compte AWS identifiant.
   - *solutionId*— L'identifiant de ressource unique pour la solution Amazon Personalize déployée par la campagne.
   - *campaignId*— L'identifiant de ressource unique pour la campagne Amazon Personalize à partir duquel récupérer les données de recommandation.

2. Lorsque vous avez terminé, enregistrez le fichier sous `RetrieveRecommendationsPolicy.json`.

3. À l'aide de l'interface de ligne de commande, accédez au répertoire dans lequel vous avez enregistré le fichier `RetrieveRecommendationsPolicy.json`.
4. Entrez la commande suivante pour créer une stratégie et nommez-la `RetrieveRecommendationsPolicy`. Pour utiliser un autre nom, remplacez-le *`RetrieveRecommendationsPolicy`* par le nom de votre choix.

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document
 file://RetrieveRecommendationsPolicy.json
```

> **Note**
>
> Si vous recevez un message indiquant que votre compte n'est pas autorisé à effectuer l'opération `CreatePolicy`, vous devez attacher une stratégie à votre compte d'utilisateur qui vous permet de créer des politiques et des rôles IAM pour votre compte. Pour plus d'informations, consultez Ajout et suppression d'autorisations basées sur l'identité IAM dans le Guide de l'utilisateur IAM.

5. Copiez l'Amazon Resource Name (ARN) de la stratégie (`arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy` dans l'exemple précédent). Dans la section suivante, vous aurez besoin de cet ARN pour créer le rôle IAM.

Étape 2 : création du rôle IAM

Une fois que vous avez créé la politique IAM, vous pouvez créer un rôle IAM et lui attacher la politique.

Chaque rôle IAM contient une stratégie d'approbation, à savoir un ensemble de règles qui spécifie les entités autorisées à endosser le rôle. Dans cette section, vous allez créer une stratégie d'approbation qui autorise Amazon Pinpoint à assumer le rôle. Ensuite, vous créez le rôle lui-même. Vous attachez ensuite cette stratégie au rôle.

Pour créer le rôle IAM

1. Dans un éditeur de texte, créez un fichier. Collez le code suivant dans le fichier :

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                "AWS:SourceAccount": "111122223333"
                },
                "ArnLike": {
                "AWS:SourceArn": "arn:aws:mobiletargeting:us-
east-1:444455556666:apps/*"
                }
            }
        }
    ]
}
```

2. Enregistrez le fichier sous le nom `RecommendationsTrustPolicy.json`.
3. À l'aide de l'interface de ligne de commande, accédez au répertoire dans lequel vous avez enregistré le fichier `RecommendationsTrustPolicy.json`.
4. Entrez la commande suivante pour créer un nouveau rôle et nommez-le `PinpointRoleforPersonalize`. Pour utiliser un autre nom, remplacez-le *`PinpointRoleforPersonalize`* par le nom de votre choix.

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-
document file://RecommendationsTrustPolicy.json
```

5. Entrez la commande suivante pour attacher la stratégie que vous avez créée dans la section précédente au rôle que vous venez de créer :

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

Dans la commande précédente, remplacez *arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy* par l'ARN de la politique que vous avez créée dans la section précédente. Remplacez-le également *PinpointRoleforPersonalize* par le nom du rôle que vous avez spécifié à l'étape 4, si vous avez spécifié un autre nom pour le rôle.

Rôle IAM pour la diffusion des événements vers Kinesis

Amazon Pinpoint peut envoyer automatiquement des données d'utilisation des applications, ou des données d'événements, depuis votre application vers un flux de données Amazon Kinesis ou un flux de diffusion Amazon Data Firehose sur votre compte. AWS Avant qu'Amazon Pinpoint ne puisse commencer la diffusion en continu des données d'événements, vous devez déléguer les autorisations requises à Amazon Pinpoint.

Si vous utilisez la console pour configurer la diffusion en continu d'événements, Amazon Pinpoint crée automatiquement un rôle Gestion des identités et des accès AWS (IAM) avec les autorisations requises. Pour plus d'informations, consultez la section Diffusion d'événements Amazon Pinpoint vers Kinesis dans le guide de l'utilisateur Amazon Pinpoint.

Si vous souhaitez créer le rôle manuellement, attachez les stratégies suivantes au rôle :

- Une stratégie d'autorisations qui permet à Amazon Pinpoint d'envoyer des données d'événement à votre flux.
- Une stratégie d'approbation qui permet à Amazon Pinpoint d'assumer le rôle.

Après avoir créé le rôle, vous pouvez configurer Amazon Pinpoint pour envoyer automatiquement les événements à votre flux. Pour plus d'informations, consultez Diffusez les données d'événements des applications via Kinesis et Firehose à l'aide d'Amazon Pinpoint dans ce guide.

Création du rôle IAM (AWS CLI)

Effectuez les étapes suivantes pour créer un rôle IAM avec AWS Command Line Interface (AWS CLI). Pour savoir comment créer le rôle à l'aide de la console Amazon Pinpoint, consultez Diffusion d'événements Amazon Pinpoint vers Kinesis dans le Guide de l'utilisateur Amazon Pinpoint.

Si vous ne l'avez pas encore installé AWS CLI, consultez la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur. Vous devez également avoir créé un flux Kinesis ou un flux Firehose. Pour plus d'informations sur la création de ces ressources, consultez Creating

and Managing Streams dans le guide du développeur Amazon Kinesis Data Streams ou Creating an Amazon Data Firehose delivery stream dans le manuel du développeur Amazon Data Firehose.

Pour créer le rôle IAM à l'aide du AWS CLI

1. Créez un nouveau fichier. Collez la politique suivante dans le document et apportez les modifications suivantes :

   - Remplacez *region* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
   - *accountId*Remplacez-le par l'identifiant unique de votre AWS compte.
   - Remplacez *applicationId* par l'identifiant unique du projet.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/applicationId"
                }
            }
        }
    ]
}
```

   Lorsque vous avez terminé, enregistrez le fichier sous `PinpointEventStreamTrustPolicy.json`.

2. Utilisez la commande `create-role` pour créer le rôle et attachez la stratégie d'approbation :

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-
document file://PinpointEventStreamTrustPolicy.json
```

3. Créez un fichier qui contient la stratégie d'autorisations de votre rôle.

   Si vous configurez Amazon Pinpoint pour envoyer des données vers un flux Kinesis, collez la politique suivante dans le fichier et remplacez la suivante :

   - Remplacez *region* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
   - *accountId*Remplacez-le par l'identifiant unique de votre AWS compte.
   - *streamName*Remplacez-le par le nom de votre flux Kinesis.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": {
           "Action": [
               "kinesis:PutRecords",
               "kinesis:DescribeStream"
           ],
           "Effect": "Allow",
           "Resource": [
               "arn:aws:kinesis:us-east-1:111122223333:stream/streamName"
           ]
       }
   }
   ```

   Sinon, si vous configurez Amazon Pinpoint pour envoyer des données vers un flux Firehose, collez la politique suivante dans le fichier et remplacez la suivante :

   - Remplacez *region* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
   - *accountId*Remplacez-le par l'identifiant unique de votre AWS compte.
   - *delivery-stream-name*Remplacez-le par le nom de votre stream Firehose.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "firehose:PutRecordBatch",
            "firehose:DescribeDeliveryStream"
        ],
        "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/delivery-stream-name"
        ]
    }
}
```

Lorsque vous avez terminé, enregistrez le fichier sous `PinpointEventStreamPermissionsPolicy.json`.

4. Utilisez la commande `put-role-policy` pour attacher la stratégie d'autorisations au rôle :

```
aws iam put-role-policy --role-name PinpointEventStreamRole --policy-
name PinpointEventStreamPermissionsPolicy --policy-document file://
PinpointEventStreamPermissionsPolicy.json
```

Rôle IAM pour l'envoi d'e-mails avec Amazon SES

Amazon Pinpoint utilise vos ressources Amazon SES pour envoyer des e-mails relatifs à votre campagne ou à votre parcours. Avant qu'Amazon Pinpoint puisse utiliser vos ressources Amazon SES pour envoyer des e-mails, vous devez accorder les autorisations requises à Amazon Pinpoint. Votre compte doit disposer des `iam:UpdateAssumeRolePolicy` autorisations `iam:PutRolePolicy` et pour mettre à jour ou créer des rôles IAM.

La console Amazon Pinpoint peut créer automatiquement un rôle Gestion des identités et des accès AWS (IAM) avec les autorisations requises. Pour plus d'informations, consultez la section Création d'un rôle d'envoi d'orchestration d'e-mails dans le guide de l'utilisateur Amazon Pinpoint.

Si vous souhaitez créer le rôle manuellement, attachez les stratégies suivantes au rôle :

- Une politique d'autorisation qui accorde à Amazon Pinpoint l'accès à vos ressources Amazon SES.
- Une stratégie d'approbation qui permet à Amazon Pinpoint d'assumer le rôle.

Après avoir créé le rôle, vous pouvez configurer Amazon Pinpoint pour qu'il utilise vos ressources Amazon SES.

Vous pouvez tester les politiques IAM à l'aide du simulateur de politiques IAM. Pour plus d'informations, consultez Test des politiques IAM avec le simulateur de politiques IAM dans le Guide de l'utilisateur IAM.

Création du rôle IAM (AWS Management Console)

Procédez comme suit pour créer manuellement un rôle IAM pour votre campagne ou votre parcours d'envoi d'e-mails.

1. Créez une nouvelle politique d'autorisation en suivant les instructions de la section Création de politiques à l'aide de l'éditeur JSON du guide de l'utilisateur IAM.

   - À l'étape 5, appliquez la politique d'autorisation suivante pour le rôle IAM.

     - Remplacez *partition* par la partition dans laquelle se trouve la ressource. En standard Régions AWS, la partition est `aws`. Si vous avez des ressources dans d'autres partitions, la partition est `aws-partitionname`. Par exemple, la partition des ressources dans le AWS GovCloud (US-West) est `aws-us-gov`.
     - Remplacez *region* par le nom de celui Région AWS qui héberge le projet Amazon Pinpoint.
     - *accountId*Remplacez-le par l'identifiant unique de votre Compte AWS.

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PinpointUsesSESForEmailSends",
            "Effect": "Allow",
```

```
            "Action": [
                "ses:SendEmail",
                "ses:SendRawEmail"
            ],
            "Resource": [
            "arn:aws:ses:us-east-1:111122223333:identity/*",
    "arn:aws:ses:us-east-1:111122223333:configuration-set/*"
            ]
        }
    ]
}
```

2. Créez une nouvelle politique de confiance en suivant les instructions de la section Création d'un rôle à l'aide de politiques de confiance personnalisées du Guide de l'utilisateur IAM.

   a. À l'étape 4, appliquez la politique de confiance suivante.

   - *accountId*Remplacez-le par l'identifiant unique de votre Compte AWS.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPinpoint",
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

   b. À l'étape 11, ajoutez la politique d'autorisation que vous avez créée à l'étape précédente.

Dépannage de la gestion des identités et des accès Amazon Pinpoint

Utilisez les informations suivantes pour identifier et résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec Amazon Pinpoint et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans Amazon Pinpoint
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte pour accéder à mes ressources Amazon Pinpoint

Je ne suis pas autorisé à effectuer une action dans Amazon Pinpoint

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur vous a fourni vos informations de connexion.

L'exemple d'erreur suivant se produit lorsque l'utilisateur `mateojackson` tente d'utiliser la console pour afficher des informations détaillées concernant un projet mais ne dispose pas des autorisations `mobiletargeting:GetApp`.

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

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource *`my-example-project`* à l'aide de l'action `mobiletargeting:GetApp`.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter l'action `iam:PassRole`, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon Pinpoint.

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 Amazon Pinpoint. 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 à mon AWS compte pour accéder à mes ressources Amazon Pinpoint

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 (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si Amazon Pinpoint est compatible avec ces fonctionnalités, consultez Fonctionnement d'Amazon Pinpoint 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.

Journalisation et surveillance dans Amazon Pinpoint

La journalisation et la surveillance sont un élément important du maintien de la fiabilité, de la disponibilité et des performances de vos projets Amazon Pinpoint et d'autres types de ressources Amazon Pinpoint. Vous devez enregistrer et collecter les données de surveillance de toutes les parties de vos projets et ressources Amazon Pinpoint afin de corriger plus facilement une panne multipoint, le cas échéant. AWS fournit plusieurs outils qui peuvent vous aider à enregistrer et à collecter ces données, et à répondre aux incidents potentiels :

AWS CloudTrail

Amazon Pinpoint s'intègre AWS CloudTrailà un service qui fournit un enregistrement des actions effectuées dans Amazon Pinpoint par un utilisateur, un rôle ou un autre service. AWS Cela inclut les actions de la console Amazon Pinpoint et les appels par programmation aux opérations d'API Amazon Pinpoint. En utilisant les informations collectées par CloudTrail, vous pouvez déterminer quelles demandes ont été adressées à Amazon Pinpoint. Pour chaque demande, vous pouvez identifier le moment où elle a été faite, l'adresse IP à partir de laquelle elle a été faite, qui l'a faite, ainsi que des détails supplémentaires. Pour plus d’informations, consultez Enregistrez les appels d'API Amazon Pinpoint avec AWS CloudTrail dans ce guide.

Amazon CloudWatch

Vous pouvez utiliser Amazon CloudWatch pour collecter, consulter et analyser plusieurs statistiques importantes liées à votre compte Amazon Pinpoint et à vos projets. Vous pouvez également l'utiliser CloudWatch pour créer des alarmes qui vous avertissent si la valeur d'une métrique répond à certaines conditions et est inférieure ou supérieure à un seuil que vous définissez. Si vous créez une alarme, CloudWatch envoie une notification à une rubrique Amazon Simple Notification Service (Amazon SNS) que vous spécifiez. Pour plus d'informations, consultez la section Surveillance d'Amazon Pinpoint avec Amazon CloudWatch dans le guide de l'utilisateur Amazon Pinpoint.

AWS Health Tableaux de bord

À l'aide de AWS Health tableaux de bord, vous pouvez vérifier et surveiller l'état de votre environnement Amazon Pinpoint. Pour vérifier l'état général du service Amazon Pinpoint, utilisez le AWS Service Health Dashboard. Pour vérifier, surveiller et consulter les données historiques relatives à tout événement ou problème susceptible d'affecter plus spécifiquement votre AWS environnement, utilisez le AWS Personal Health Dashboard. Pour en savoir plus sur ces tableaux de bord, consultez le Guide de l'utilisateur AWS Health.

AWS Trusted Advisor

AWS Trusted Advisor inspecte votre AWS environnement et fournit des recommandations sur les possibilités de remédier aux failles de sécurité, d'améliorer la disponibilité et les performances du système et de réaliser des économies. Tous les AWS clients ont accès à un ensemble de Trusted Advisor chèques de base. Les clients disposant d'un plan de support Business ou Enterprise ont accès à des Trusted Advisor contrôles supplémentaires.

Nombre de ces contrôles peuvent vous aider à évaluer le niveau de sécurité de vos ressources Amazon Pinpoint dans le cadre de l'ensemble de votre AWS compte. Par exemple, l'ensemble des contrôles Trusted Advisor de base comprend les éléments suivants :

- Configurations de journalisation pour votre AWS compte, pour chaque AWS région prise en charge.
- Autorisations d'accès pour vos compartiments Amazon Simple Storage Service (Amazon S3), qui peuvent contenir des fichiers que vous importez dans Amazon Pinpoint pour créer des segments.
- Utilisation d' Gestion des identités et des accès AWS utilisateurs, de groupes et de rôles pour contrôler l'accès aux ressources Amazon Pinpoint.
- Configurations IAM et paramètres de politique susceptibles de compromettre la sécurité de votre AWS environnement et des ressources Amazon Pinpoint.

Pour plus d’informations, consultez AWS Trusted Advisor dans le Guide de l’utilisateur Support .

Validation de la conformité pour Amazon Pinpoint

Third-party les auditeurs évaluent la sécurité et la conformité d'Amazon Pinpoint dans le cadre de plusieurs programmes de AWS conformité. Il s'agit notamment des contrôles AWS du système et de l'organisation (SOC), FedRAMP, HIPAA, 27001:2013 pour les contrôles de gestion de la sécurité ISO/IEC , 27017:2015 pour les contrôles spécifiques au cloud ISO/IEC , 27018:2014 pour la protection des données personnelles, 9001:2015 pour les systèmes de gestion de la qualité ISO/IEC , etc. ISO/IEC

Pour obtenir la liste des AWS services concernés par des programmes de conformité spécifiques, voir AWS Services concernés par programme de conformitéAWS . Pour des informations générales, consultez les programmes de AWS conformité, les programmes AWS .

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

Lorsque vous utilisez Amazon Pinpoint, votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides de démarrage rapide sur la sécurité et la conformité Guides sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur. AWS
- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA — Ce livre blanc décrit comment les entreprises peuvent créer des applications. AWS HIPAA-compliant
- AWS ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- Évaluation des ressources à l'aide des règles énoncées dans le guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub CSPM— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Amazon Pinpoint est un service éligible à la AWS loi HIPAA lorsque les clients utilisent les canaux de communication appropriés. Si vous souhaitez utiliser Amazon Pinpoint pour exécuter des charges de travail contenant des données de santé protégées (PHI) telles que définies par la loi HIPAA et les lois et règlements connexes, vous devez utiliser le canal de messagerie, le canal de notification push ou le canal SMS pour envoyer des messages contenant des données de santé protégées (PHI) Si vous utilisez le canal SMS pour envoyer des messages contenant des PHI, vous devez envoyer ces messages à partir d'un code court dédié que vous avez demandé pour votre AWS compte dans le but explicite d'envoyer des messages contenant ou pouvant contenir des PHI. Le canal vocal n'est pas éligible à la loi AWS HIPAA ; ne l'utilisez pas pour envoyer des messages contenant des PHI.

Résilience dans Amazon Pinpoint

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions 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, consultez le Guide d'architecture résiliente Amazon Pinpoint sur les architectures de référence.

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

Sécurité de l'infrastructure dans Amazon Pinpoint

En tant que service géré, Amazon Pinpoint est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Amazon Pinpoint via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Des suites de chiffrement dotées d'un secret de transmission parfait (PFS), telles que DHE (Ephemeral) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

Bien que vous puissiez effectuer ces appels d'API à partir de n'importe quel emplacement réseau, Amazon Pinpoint prend en charge les stratégies d'accès basées sur les ressources. Ces stratégies peuvent inclure des restrictions basées sur l'adresse IP source. Pour en savoir plus sur ce type de stratégie, veuillez consulter Gestion de l'accès à l'aide de politiques.

En outre, vous pouvez configurer et utiliser diverses fonctionnalités de AWS sécurité pour contrôler l'accès aux ressources Amazon Pinpoint à partir de toutes les applications mobiles ou Web que vous intégrez à Amazon Pinpoint. Cela inclut des restrictions sur les appels d'API pour des tâches telles que l'ajout de points de terminaison, la mise à jour de données de point de terminaison, la soumission de données d'événement et la création de rapports sur les données d'utilisation.

Pour utiliser ces fonctionnalités, nous vous recommandons d'utiliser les SDK ou AWS Amplify JavaScript bibliothèques AWS mobiles pour intégrer des applications mobiles et Web à Amazon

Pinpoint. Pour les applications Android ou iOS, nous vous recommandons d'utiliser respectivement le AWS Mobile SDK pour Android ou le Applications mobiles AWS SDK for iOS. Pour les applications JavaScript-based mobiles ou Web, nous vous recommandons d'utiliser la AWS Amplify JavaScript bibliothèque pour le Web ou la AWS Amplify JavaScript bibliothèque pour React Native. Pour en savoir plus sur ces ressources, consultez Getting started with the AWS mobile SDK, Getting started with the AWS Amplify library for the web and Getting started with the AWS Amplify library for React Native.

Configuration et analyse des vulnérabilités dans Amazon Pinpoint

En tant que service géré, Amazon Pinpoint 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 maintenir l'infrastructure sous-jacente de votre compte et de vos ressources Amazon Pinpoint. Ces procédures ont été vérifiées et certifiées par les tiers appropriés.

Pour plus d’informations, consultez les ressources suivantes :

- Validation de la conformité pour Amazon Pinpoint
- Modèle de responsabilité partagée
- Amazon Web Services : présentation des procédures de sécurité (livre blanc)

Bonnes pratiques de sécurité pour Amazon Pinpoint

Utilisez les comptes AWS Identity and Access Management (IAM) pour contrôler l'accès aux opérations d' Amazon Pinpoint API, en particulier aux opérations qui créent, modifient ou suppriment des ressources. Amazon Pinpoint Pour l' Amazon Pinpoint API, ces ressources incluent les projets, les campagnes et les voyages. Pour l'API Amazon Pinpoint SMS et voix, ces ressources incluent les numéros de téléphone, les groupes et les jeux de configuration.

- Créez un utilisateur individuel pour chaque personne qui gère les Amazon Pinpoint ressources, y compris vous-même. N'utilisez pas les informations d'identification AWS root pour gérer les ressources Amazon Pinpoint.
- 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 Amazon Pinpoint la sécurité, consultez la section Sécurité dans Amazon Pinpoint. 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.

Quotas Amazon Pinpoint

Les sections suivantes énumèrent et décrivent les quotas, anciennement appelés limites, qui s'appliquent aux ressources et aux opérations Amazon Pinpoint. Certains quotas peuvent être augmentés, mais pas tous. Pour déterminer si vous pouvez demander l'augmentation d'un quota, reportez-vous à l'instruction ou à la colonne Admissible à l'augmentation de chaque section.

Rubriques

- Quotas de projet
- Quotas de demande d’API
- Quotas de campagne
- Quotas d’e-mails
- Quotas de points de terminaison
- Quotas d’importation de points de terminaison
- Quotas d’ingestion d’événements
- Quotas de parcours
- Quotas Lambda
- Quotas de machine learning
- Quotas de modèles de message
- Quotas de notifications push
- Quotas de messages dans l'application
- Quotas de segments
- Quotas de SMS
- Quotas de 10DLC
- Quotas de messages vocaux
- Demande d’augmentation de quota

Quotas de projet

Le tableau suivant répertorie les quotas liés aux projets dans Amazon Pinpoint.

Ressource	Quota par défaut	Éligible à une augmentation
Projets	Dans chacun d'eux Région AWS, vous pouvez avoir jusqu'à 100 projets.	Non

Quotas de demande d’API

Amazon Pinpoint implémente des quotas qui limitent la taille et le nombre de demandes que vous pouvez envoyer à l'API Amazon Pinpoint depuis votre compte. AWS

La taille maximale d’une charge utile d’appel (demande et réponse) est de 7 Mo, sauf indication contraire pour un type particulier de ressource. Pour déterminer si une ressource a un quota différent, consultez la section appropriée de cette rubrique pour ce type de ressource.

Le nombre maximal de demandes varie en fonction du type de quota et du fonctionnement de l'API. Amazon Pinpoint implémente deux types de quotas pour les demandes d'API :

- Quotas de taux : également appelés limites de taux, ce type de quota définit le nombre maximal de demandes que vous pouvez effectuer par seconde pour une opération particulière. Il contrôle le taux de demandes envoyées ou reçues par compte.
- Quotas de débordement : également appelés limites de débordement ou capacité de débordement, ce type de quota définit le nombre maximum de demandes simultanément en cours pour un compte.

Le tableau suivant répertorie les quotas de taux et de débit pour l'API Amazon Pinpoint.

Opération	burst/rate Quota par défaut (demandes par seconde)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10
CreateImportJob	300

Opération	burst/rate Quota par défaut (demandes par seconde)
CreatePushTemplate	10
CreateSegment	25
CreateSmsTemplate	10
CreateVoiceTemplate	10
DeleteCampaign	25
DeleteEndpoint	5
DeleteSegment	25
GetEndpoint	10
PhoneNumberValidate	20
PutEvents	15
SendMessages	4 000
SendUsersMessages	6 000
UpdateCampaign	25
UpdateEmailTemplate	10
UpdateEndpoint	10
UpdateEndpointsBatch	2
UpdateInAppTemplate	10
UpdatePushTemplate	10
UpdateSegment	25
UpdateSmsTemplate	10

Opération	burst/rate Quota par défaut (demandes par seconde)
UpdateVoiceTemplate	10
Toutes les autres opérations	300

Le tableau suivant répertorie les quotas d'importation de fichiers pour `CreateImportJob`.

Opération	Quota par défaut	Éligible à une augmentation
Nombre maximal de fichiers d'importation	10 000 fichiers par tâche d'importation	Non

Si vous dépassez l'un de ces quotas, Amazon Pinpoint limite la demande, c'est-à-dire qu'il rejette une demande par ailleurs valide et renvoie une erreur `TooManyRequests`. La limitation est basée sur le nombre total de demandes que vous effectuez à partir de votre compte pour une opération spécifique dans une Région AWS spécifique. En outre, les décisions de limitation sont calculées indépendamment pour chaque opération. Par exemple, si Amazon Pinpoint limite une demande pour l'opération `SendMessages`, une demande simultanée pour l'opération `UpdateEndpoint` peut aboutir.

Quotas de campagne

Les quotas suivants s'appliquent à la ressource Campaigns de l'API Amazon Pinpoint.

Les quotas suivants s'appliquent chaque Région AWS année et certains peuvent être augmentés. Pour plus d'informations, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

Ressource	Quota par défaut	Éligible à une augmentation
Campagnes actives	200 par compte	Non

Ressource	Quota par défaut	Éligible à une augmentation
	Note Une campagne active est une campagne qui n'est pas terminée ou n'a pas échoué. Les campagnes actives ont le statut `SCHEDULED` , `EXECUTING` ou `PENDING_NEXT_RUN` .	
Taille maximum du segment	Pour les segments importés : 100 000 000 par campagne Pour les segments dynamiques : nombre illimité	Non

Ressource	Quota par défaut	Éligible à une augmentation
Campagnes basées sur des événements	Chaque projet peut comporter jusqu'à 25 campagnes envoyées lorsque des événements se produisent. Les campagnes qui utilisent des déclencheurs basés sur des événements doivent employer des segments dynamiques. Elles ne peuvent pas utiliser de segments importés. Si vous intégrez votre application à Amazon Pinpoint à l'aide d'un SDK AWS mobile, les messages issus de campagnes basées sur des événements sont envoyés uniquement aux clients dont les applications exécutent la AWS Mobile SDK pour Android version 2.7.2 ou ultérieure, ou la version 2.6.30 ou ultérieure. Applications mobiles AWS SDK for iOS Si Amazon Pinpoint ne parvient pas à remettre un message à partir d'une campagne basée sur des événements dans un délai de cinq minutes, il supprime le message et ne tente pas de le transmettre à nouveau.	Non

Quotas d’e-mails

Les quotas indiqués dans les sections suivantes s’appliquent au canal des e-mails.

Quotas de messages électroniques

Ressource	Quota par défaut	Éligible à une augmentation
Taille de message maximale (pièces jointes incluses)	10 Mo par message.	Non
Nombre d’identités vérifiées	10 000 identités **Note** Le terme Identités fait référence aux adresses e-mail ou aux domaines, ou à une combinaison des deux. Chaque e-mail que vous envoyez à l'aide d'Amazon Pinpoint doit être envoyé à partir d'une identité vérifiée.	Non

Quotas d’expéditeurs et de destinataires d’e-mails

Ressource	Quota par défaut	Éligible à une augmentation
Adresse de l’expéditeur	L’ensemble des adresses ou domaines d’envoi doivent être vérifiés.	Non

Ressource	Quota par défaut	Éligible à une augmentation
Adresse du destinataire	Si votre compte se trouve encore dans l'environnement de test (sandbox), l'ensembl e des adresses ou domaines de messagerie doivent être vérifiés. Si votre compte se trouve en dehors de l'environnement de test, vous pouvez envoyer un e-mail à n'importe quelle adresse valide.	Oui
Nombre de destinataires par message	50 destinataires par message	Non
Nombre d'identités que vous pouvez vérifier	10 000 identités par AWS région **Note** Le terme Identités fait référence aux adresses e-mail ou aux domaines, ou à une combinaison des deux. Chaque e-mail que vous envoyez à l'aide d'Amazon Pinpoint doit être envoyé à partir d'une identité vérifiée.	Non

Quotas d’envoi d’e-mails

Le quota d'envoi, le taux d'envoi et les limites de l'environnement de test (sandbox) sont partagés entre les deux services de la même région. Si vous utilisez Amazon SES dans us-east-1, que vous avez été retiré du sandbox et que vos quota/rate envois ont augmenté, ces modifications s'appliquent toutes à votre compte Pinpoint dans us-east-1.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre d’emails pouvant être envoyés par période de 24 heures (quota d’envoi)	Si votre compte se trouve dans l’environnement de test : 200 e-mails par période de 24 heures. Si votre compte est en dehors de l’environnement de test, le quota varie en fonction de votre cas d’utilisation. **Note** Ce quota est basé sur le nombre de destinataires et non sur le nombre de messages uniques envoyés. Un destinataire correspond à toute adresse e-mail insérée dans le champ À :.	Oui
Nombre d’e-mails pouvant être envoyés chaque seconde (taux d’envoi)	Si votre compte est dans l’environnement de test : 1 e-mail par seconde. Si votre compte est en dehors de l’environnement de test, le	Oui

Ressource	Quota par défaut	Éligible à une augmentation
	taux varie en fonction de votre cas d'utilisation. **Note** Ce taux est basé sur le nombre de destinataires et non sur le nombre de messages uniques envoyés. Un destinataire correspond à toute adresse e-mail insérée dans le champ À :.	

Quotas de points de terminaison

Les quotas suivants s'appliquent à la ressource Endpoints de l'API Amazon Pinpoint.

Le nombre maximum d'attributs pris en charge par point de terminaison est de 250, et la taille maximale de chaque point de terminaison est de 15 Ko. Ce nombre d'attributs peut toutefois être limité par la taille totale d'un point de terminaison, qui inclut tous les attributs. Si vous rencontrez des erreurs lors de l'ajout d'attributs à votre modèle, pensez à diminuer la quantité de données dans chaque attribut ou à diminuer le nombre d'attributs.

Ressource	Quota par défaut	Éligible à une augmentation
Taille du point de terminaison	Taille maximale : 15 Ko	Non
Attributs assignés aux paramètres `Attributes` , `Metrics` et `UserAttributes` , collectivement	250 pour tous les paramètres d'attribut par application	Non

Ressource	Quota par défaut	Éligible à une augmentation
Attributs assignés au paramètre `Attributes`	250 pour tous les paramètres d'attribut par application	Non
Attributs assignés au paramètre `Metrics`	250 pour tous les paramètres d'attribut par application	Non
Attributs assignés au paramètre `UserAttributes`	250 pour tous les paramètres d'attribut par application	Non
Longueur du nom d’attribut	50 caractères	Non
Longueur de la valeur d’attribut	100 caractères	Non
Objets `EndpointBatchItem` dans une charge utile `EndpointBatchRequest`	100 par charge utile. La taille de la charge utile ne doit pas dépasser 7 Mo.	Non
Points de terminaison avec le même ID utilisateur	15 points de terminaison uniques par ID utilisateur	Non
Valeurs assignées aux attributs de paramètre `Attributes`	50 par attribut	Non
Valeurs assignées aux attributs de paramètre `UserAttributes`	50 par attribut	Non

Quotas d’importation de points de terminaison

Les quotas suivants s'appliquent à l'importation de points de terminaison dans Amazon Pinpoint.

Ressource	Quota par défaut	Éligible à une augmentation
Tâches d'importation actives	10 par compte Les tâches d'importation ne sont prises en compte dans ce quota que si elles sont en cours d'exécution. Une fois la tâche d'importation terminée, elle n'est plus prise en compte dans ce quota.	Non
Taille d’importation	1 Go par tâche d’importation Par exemple, si chaque point de terminaison est inférieur ou égal à 4 Ko, vous pouvez importer 250 000 points de terminaison.	Non

Quotas d’ingestion d’événements

Les quotas suivants s'appliquent à l'ingestion d'événements à l'aide de la ressource AWS Mobile SDKs et Events de l'API Amazon Pinpoint.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre maximal de types d’événements personnalisés	1 500 par application	Non
Nombre maximal de clés d’attributs personnalisés	500 par application	Non
Nombre maximal de valeurs d’attributs personnalisés par clé d’attribut	100 000. Tout nombre supérieur à 100 000 est toujours enregistré, mais ne sera pas disponible dans la	Non

Ressource	Quota par défaut	Éligible à une augmentation
	console d'analyse Amazon Pinpoint.	
Nombre maximal de caractères par clé d’attribut	50	Non
Nombre maximal de caractères par valeur d’attribut	200. Si le nombre de caractères est supérieur à 200, l'événement est annulé.	Non
Nombre maximal de clés métriques personnalisées	500 par application	Non
Nombre maximal d’événements dans une demande.	100 par demande	Non
Taille maximale d’une demande	4 Mo	Non
Taille maximale d’un événement individuel	1 000 Ko	Non
Nombre maximal de clés d’attributs et de clés métriques pour chaque événement	40 par demande	Non

Quotas de parcours

Les quotas suivants s’appliquent aux parcours.

Les quotas suivants s'appliquent chaque Région AWS année et certains peuvent être augmentés. Pour plus d'informations, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre maximum de parcours actifs	50 par compte	Non
Nombre maximum d'actifs EventTriggeredJourneys	20 par compte	Non
Nombre maximum d’activités du parcours	40 par parcours	Non
Taille maximum du segment	Pour les segments importés : 100 000 000 par parcours Pour les segments dynamiques : nombre illimité	Non
Activités maximales du centre de contact	3 par parcours	Non
Nombre maximal de règles de jours fermés	20 par canal	Non
Longueur maximale du nom de la règle des jours fermés	150 caractères	Non
Nombre maximum de jours entre les heures de début et de fin pour une règle des jours fermés	7 jours	Non
Nombre maximal de règles d’heures d’ouverture	4 par jour	Non

Quotas Lambda

Les quotas suivants s'appliquent aux configurations Amazon Pinpoint pour la récupération et le traitement des données à partir de Lambda

Ressource	Quota par défaut	Éligible à une augmentation
Taille maximale de la charge utile d'une invocation (demande et réponse) pour une fonction Lambda	6 Mo	Non
Durée maximale d'attente pour qu'une fonction Lambda traite les données	15 secondes	Non
Nombre maximal d'attributs d'événements par point de terminaison	5	Non
Nombre maximal de caractères pour le nom d'attribut d'événement	128 caractères	Non
Nombre maximal de caractères pour la valeur d'attribut d'événement	128 caractères	Non
Nombre maximum de jours qu'un parcours peut durer	540 jours	Non

Quotas de machine learning

Les quotas suivants s'appliquent aux configurations Amazon Pinpoint pour la récupération et le traitement des données à partir de modèles de machine learning (ML).

Ressource	Quota par défaut	Éligible à une augmentation
Nombre maximal de configurations de modèle	1 par modèle de message 100 par compte	Non

Ressource	Quota par défaut	Éligible à une augmentation
Nombre maximal de recommandations	5 par point de terminaison ou utilisateur	Non
Nombre maximal d’attributs recommandés par point de terminaison ou utilisateur	1, si les valeurs d’attribut ne sont pas traitées par une fonction AWS Lambda 10, si les valeurs d’attribut sont traitées par une fonction AWS Lambda	Non
Longueur maximale d’un nom d’attribut recommandé	50 caractères pour un nom d’attribut 25 caractères pour un nom d'affichage d'attribut (nom qui apparaît dans le localisateur d'attributs de la console)	Non
Longueur maximale d'une valeur d'attribut recommandée récupérée à partir d'Amazon Personalize	100 caractères	Non
Taille maximale de la charge utile d'une invocation (demande et réponse) pour une fonction Lambda	6 Mo	Non
Durée maximale d'attente pour qu'une fonction Lambda traite les données	15 secondes	Non
Nombre maximal de tentative s d'invocation d'une fonction Lambda	3 tentatives	Non

En fonction de la manière dont vous configurez Amazon Pinpoint pour utiliser un modèle de machine learning, des quotas supplémentaires peuvent s'appliquer. Pour en savoir plus sur les quotas Amazon Personalize, consultez Quotas dans le Guide du développeur Amazon Personalize. Pour en savoir plus sur les quotas AWS Lambda , veuillez consulter Quotas dans le Guide du développeur AWS Lambda .

Quotas de modèles de message

Les quotas suivants s'appliquent aux modèles de message de votre compte Amazon Pinpoint.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre maximal de modèles de message	20 000 par compte	Non
Nombre maximal de versions	5 000 par modèle	Non
Nombre maximal de caractères dans un modèle d’e-mail	600 000 caractères	Non
Nombre maximal de caractères dans un modèle dans l'application	200 000 caractères	Non
Nombre maximal de caractères dans les parties de modèle par défaut d’un modèle de notification push	4 000 caractères	Non
Nombre maximal de caractères dans les parties de modèle spécifiques à ADM d’un modèle de notification push	6 000 caractères	Non
Nombre maximum de caractères dans les parties APNs spécifiques du modèle	4 000 caractères	Non

Ressource	Quota par défaut	Éligible à une augmentation
d'un modèle de notification push		
Nombre maximal de caractères dans les parties de modèle spécifiques à Baidu d'un modèle de notification push	4 000 caractères	Non
Nombre maximal de caractères dans les parties de modèle spécifiques à FCM d'un modèle de notification push	4 000 caractères	Non
Nombre maximal de caractères dans un modèle SMS	1 600 caractères	Non
Nombre maximum de caractères dans un modèle vocal	10 000 caractères	Non

Quotas de notifications push

Les quotas suivants s'appliquent aux messages qu'Amazon Pinpoint envoie via les canaux de notification push.

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
Taille de la charge utile des messages Amazon Device Messaging (ADM)	6 Ko par message	Non

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

Quotas de messages dans l'application

Le quota suivant s'applique aux messages dans l'application que vous gérez avec Amazon Pinpoint.

Les quotas suivants s'appliquent chaque Région AWS année et certains peuvent être augmentés. Pour plus d'informations, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre maximum de fois que vous pouvez appeler l'API `GetInAppMessages` par seconde.	5 000 demandes par seconde	Oui
Campagnes de messagerie dans l'application	Chaque projet peut inclure jusqu'à 25 campagnes utilisant le canal de messagerie dans l'application.	Oui, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

Quotas de segments

Le quota suivant s'applique à la ressource Segments de l'API Amazon Pinpoint API.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre maximal de dimensions pouvant être utilisées pour créer un segment	100 par segment	Non
Nombre maximum de groupes de segments par segment	5	Non
Nombre maximum de segments source par segment	5	Non
Profondeur maximale des segments source. Par exemple, si un segment possède un segment source qui possède également un segment source, la chaîne de profondeur n'est pas plus longue que cette limite.	5	Non

Quotas de SMS

Pour les quotas de SMS, consultez la section Quotas de SMS dans le guide de l'utilisateur pour la messagerie SMS destinée aux utilisateurs AWS finaux.

Quotas de 10DLC

Pour les quotas de 10 DLC, consultez la section Quotas de 10 DLC dans le guide de l'utilisateur AWS final pour la messagerie SMS.

Quotas de messages vocaux

Pour les quotas vocaux, consultez la section Quotas vocaux dans le guide de l'utilisateur pour la messagerie SMS destinée aux utilisateurs AWS finaux.

Demande d’augmentation de quota

Si la valeur indiquée dans la colonne Admissible à une augmentation d’un des tableaux ci-dessus est Oui, vous pouvez demander un relèvement de ce quota.

Pour demander une augmentation de quota

1. Connectez-vous à l' AWS Management Console adresse https://console.aws.amazon.com/.
2. Créez un nouveau dossier de AWS support à l'adresse https://console.aws.amazon.com/support/home#/case/create.
3. Dans le panneau Vos cas de support, choisissez Créer un cas.
4. Choisissez le lien Vous recherchez des augmentations de la limite de service ?.
5. Sous Augmentation du quota de service, pour Service, choisissez l'une des options suivantes :
   - Pour demander une augmentation de quota concernant le canal de messagerie, choisissez Pinpoint Email (E-mail Pinpoint).
   - Pour demander une augmentation du quota pour les limites de dépenses ou les taux d'envoi de SMS, choisissez Pinpoint SMS. Pour toutes les autres augmentations du quota de SMS, choisissez Pinpoint
   - Pour demander une augmentation de quota liée au canal vocal, choisissez Pinpoint Voice (Voix Pinpoint).
   - Pour demander une augmentation de quota concernant toute autre fonctionnalité Amazon Pinpoint, choisissez Pinpoint.
6. Selon le service que vous choisissez, il peut vous être demandé de saisir les informations suivantes :
   - (Facultatif) Pour Fournir un lien vers le site ou l’application qui enverra les SMS, entrez les informations sur le site Web, l’application ou le service qui enverra des SMS.
   - (Facultatif) Pour Quel type de message envisagez-vous d’envoyer, choisissez le type de message que vous prévoyez d’envoyer à l’aide de votre code long :

- Mot de passe ponctuel – Messages fournissant des mots de passe que vos clients utilisent pour s'authentifier sur votre site web ou votre application.
- Promotionnel – Messages non stratégiques faisant la promotion de votre entreprise ou de votre service, comme des offres spéciales ou des annonces.
- Transactionnel – Messages d'information importants qui prennent en charge les transactions clients, comme des confirmations de commande ou des alertes de compte. Les messages transactionnels ne peuvent pas comporter de contenu promotionnel ni marketing.

- (Facultatif) Pour AWS la région à partir de laquelle vous allez envoyer des messages, choisissez la région à partir de laquelle vous allez envoyer des messages.
- (Facultatif) Pour Vers quels pays comptez-vous envoyer des messages, entrez le pays ou la région dans lesquels vous souhaitez acheter des codes courts.
- (Facultatif) Dans Comment vos clients acceptent-ils de recevoir messages de votre part, fournissez des détails sur votre processus d'acceptation.
- (Facultatif) Dans le champ Veuillez fournir le modèle de message que vous comptez utiliser pour envoyer des messages à vos clients, incluez le modèle que vous allez utiliser.

7. Sous Demandes, procédez comme suit :

   - Pour Région, choisissez votre Région AWS.
   - Pour Type de ressource, choisissez Limites générales. Le champ Type de ressource n'est présent que pour certains services.
   - Pour Quota, choisissez le quota à modifier.
   - Pour Nouvelle valeur de quota, entrez une nouvelle valeur pour le quota.
   - Pour demander une augmentation du même quota dans une demande supplémentaire Région AWS, choisissez Ajouter une autre demande, puis choisissez la demande supplémentaire Région AWS et remplissez la nouvelle demande.
8. Choisissez le quota que vous souhaitez augmenter, puis entrez la nouvelle valeur souhaitée.
9. Sous Description du cas, expliquez pourquoi vous demandez l'augmentation du quota.
10. Sous Options de contact, dans Langue de contact préférée, choisissez la langue que vous préférez utiliser lorsque vous communiquez avec l'équipe de AWS support.
11. Pour la méthode de contact, choisissez votre méthode préférée pour communiquer avec l'équipe de AWS support.
12. Sélectionnez Soumettre.

L'équipe de AWS Support fournit une première réponse à votre demande dans les 24 heures.

Pour empêcher que nos systèmes soient utilisés pour envoyer des contenus indésirables ou malveillants, chaque demande est traitée avec soin. Si nous sommes en mesure de le faire, nous répondons à votre demande dans ce délai de 24 heures. En revanche, si nous avons besoin que vous nous fournissiez de plus amples informations, le traitement de votre demande peut prendre plus de temps.

Nous ne serons peut-être pas en mesure d’accéder à votre demande si votre cas d’utilisation n’est pas conforme à nos politiques.

Historique du document pour Amazon Pinpoint

Le tableau ci-après décrit les modifications importantes dans chaque édition du Guide du développeur Amazon Pinpoint à compter de décembre 2018. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

- Dernière date de mise à jour de la documentation : 16 novembre 2023

Modification	Description	Date
Avis de fin de support	Avis de fin de prise en charge : le 30 octobre 2026, AWS cessera de prendre en charge Amazon Pinpoint. Après le 30 octobre 2026, vous ne pourrez plus accéder à la console Amazon Pinpoint ni aux ressources Amazon Pinpoint (points de terminaison, segments, campagnes, parcours et analytique). Pour plus d'informations, consultez Fin de la prise en charge d'Amazon Pinpoint. Remarque : en ce qui concerne les SMS, la voix, le push mobile, l'OTP et la validation des numéros de téléphone ne sont pas concernés par cette modification et sont pris en charge par AWS la messagerie utilisateur final.	20 mai 2025
Ajout du CloudTrail support pour l'envoi de messages	Ajout de la prise en charge de la CloudTrail journalisation	17 février 2025

	pour PutEvents SendUserMessages, et des appels d'SendMessagesAPI. Voir Actions d'API Amazon Pinpoint prises en charge dans les fichiers CloudTrail journaux.	
Amazon Pinpoint a mis à jour la documentation de son guide de l'utilisateur	Pour obtenir les dernières informations concernant la création, la configuration et la gestion de vos ressources Push, consultez le nouveau Guide de l'utilisateur final pour la messagerie push.	22 juillet 2024
En-têtes de courrier électronique	Vous pouvez ajouter des en-têtes d'e-mail à vos e-mails. Pour plus d'informations, consultez la section Envoi d'un e-mail avec des en-têtes de désabonnement.	7 mai 2024
Orchestration du courrier électronique	Amazon Pinpoint a mis à jour la façon dont il utilise vos ressources Amazon SES pour envoyer des e-mails. Pour plus d'informations, consultez le rôle IAM pour l'envoi d'e-mails avec Amazon SES.	30 avril 2024

Amazon Pinpoint a mis à jour la documentation de son guide de l’utilisateur	Les rubriques relatives à la gestion des ressources SMS et vocales sont désormais redirigées vers le Guide de l'utilisateur AWS final pour la messagerie SMS. Pour plus d'informations, consultez le Guide de l'utilisateur pour la messagerie SMS destinée aux utilisateurs AWS finaux.	8 février 2024
Quotas Amazon Pinpoint	Quotas ajoutés pour les règles relatives au nombre maximum de jours fermés, longueur maximale du nom de la règle relative aux jours fermés, nombre maximum de jours entre le début et la fin pour une règle relative aux jours fermés et nombre maximum de règles pour les heures d’ouverture. Pour plus d’informations, consultez Quotas Amazon Pinpoint.	19 décembre 2023
Amazon Pinpoint a mis à jour la documentation de son guide de l’utilisateur	Pour obtenir les informations les plus récentes concernan t la création, la configuration et la gestion de vos ressource s vocales et SMS pour les utilisateurs AWS finaux, consultez le nouveau guide de l'utilisateur de messagerie SMS pour les utilisateurs AWS finaux.	16 novembre 2023

Quotas Amazon Pinpoint	A mis à jour les quotas pour UpdateEndpointsBatch UpdateEndpoint PutEvents,, DeleteEndpoint, et GetEndpoint. Pour plus d'informations, consultez Quotas Amazon Pinpoint.	22 septembre 2023
Quotas Amazon Pinpoint	A mis à jour les quotas pour CreateEmailTemplate CreateSmsTemplate CreatePushTemplate, CreateInAppTemplate, CreateVoiceTemplate, UpdateEmailTemplate, UpdateSmsTemplate, , UpdatePushTemplate , UpdateInAppTemplate, UpdateVoiceTemplate et CreateImportJob. Pour plus d'informations, consultez Quotas Amazon Pinpoint.	12 septembre 2023
Métriques relatives à l'exécution des parcours et des campagnes	De nouvelles mesures analytiques ont été ajoutées pour les parcours et les campagnes. Pour obtenir plus d'informations, consultez Métriques d'exécution du parcours et de la campagne.	25 avril 2023

Création d'un point de terminaison de VPC d'interface pour l'API Amazon Pinpoint	Amazon Pinpoint prend désormais en charge les points de terminaison VPC d'interface. Pour plus d'informations, consultez Création d'une interface VPC pour Amazon Pinpoint	11 avril 2023
Chiffrement en transit	À compter du 22 mars 2023, Amazon Pinpoint ne prendra plus en charge TLS 1.0, mais vous pourrez toujours utiliser TLS 1.2 ou une version ultérieure. Pour de plus amples informations, veuillez consulter Chiffrement en transit.	20 mars 2023
Quotas Amazon Pinpoint	Mise à jour du processus de demande d'augmentation du quota pour les campagnes, les parcours et les messages dans l'application. Pour plus d'informations, consultez Quotas Amazon Pinpoint.	16 décembre 2022
Disponibilité par région	Amazon Pinpoint est désormais disponible dans cette région : USA Est (Ohio).	5 octobre 2022
Mises à jour des exemples de rôles IAM	Plusieurs exemples de rôles IAM ont été mis à jour dans le document afin de mieux les aligner sur les bonnes pratiques en matière de sécurité.	27 mai 2022

API SMS et Voix, version 2	Amazon Pinpoint inclut désormais une API dédiée pour l'envoi de SMS et de messages vocaux. Cette API inclut de nouvelles fonctionnalités, telles que des ensembles de configuration, des groupes et des listes de désabonnement, qui sont utiles aux clients qui envoient des SMS et des messages vocaux de manière transactionnelle. Pour plus d'informations, consultez Utiliser l'API SMS et voix Amazon Pinpoint.	1er avril 2022
Création et validation d'un mot de passe à usage unique	Amazon Pinpoint inclut désormais une fonctionnalité qui génère des mots de passe à usage unique (OTPs) et les envoie à vos utilisateurs sous forme de SMS. Il inclut également une API pour valider les codes OTP lorsque vos utilisateurs les saisissent dans votre application ou votre site. Pour plus d'informations, consultez la section Envoi et validation de mots de passe à usage unique (OTPs).	26 novembre 2021
Messages dans l'application	Ajout d'informations sur l'intégration de la fonctionnalité de messagerie dans l'application d'Amazon Pinpoint à vos applications.	10 novembre 2021

Exemples de code	Ajout d'une bibliothèque d'exemples de code pour les opérations courantes d'Amazon Pinpoint.	03 novembre 2021
Quotas de projets	Le nombre maximum de projets Amazon Pinpoint reste fixé à 100, mais ce quota peut désormais être augmenté en ouvrant une demande d'augmentation de limite de service auprès de. Support	11 octobre 2021
Mises à jour de la politique Lambda.	Certaines politiques d'autorisation Lambda doivent désormais inclure une condition `AWS:SourceAccount` . Nous avons mis à jour les exemples de politiques présentés dans les rubriques Création de chaînes personnalisées dans Amazon Pinpoint et Personnalisation des segments avec AWS Lambda pour répondre à cette exigence.	7 octobre 2021
UpdateEndpoint	L'UpdateEndpoint API Amazon Pinpoint est désormais connectée. CloudTrail	16 novembre 2020
Custom attributes (Attributs personnalisés)	Amazon Pinpoint prend désormais en charge 250 attributs dans les modèles de messagerie électronique. Consultez Quotas.	18 septembre 2020

Disponibilité par région	Amazon Pinpoint est désormais disponible dans les régions suivantes : Asie Pacifique (Tokyo), Europe (Londres) et Canada (Centre). Notez que l'API Amazon Pinpoint SMS et voix n'est pas disponible dans ces régions.	10 septembre 2020
Disponibilité par région	Amazon Pinpoint est maintenant disponible dans la région d'Asie-Pacifique (Tokyo). Notez que l'API Amazon Pinpoint SMS et voix ne prend pas en charge la voix dans cette région.	2 septembre 2020
Événements de campagne	Ajout d'informations sur un nouveau paramètre `delivery_type` d'événement de campagne dans les événements de campagne.	2 août 2020
Disponibilité par région	Amazon Pinpoint est maintenant disponible dans la région d'Asie-Pacifique (Séoul). Notez que l'API Amazon Pinpoint ne prend pas en charge la voix dans cette région.	31 juillet 2020
Disponibilité par région	Amazon Pinpoint est désormais disponible dans la AWS GovCloud (US) région.	30 avril 2020

Canaux personnalisés	Mise à jour des informations sur la création de canaux personnalisés à l'aide de fonctions Lambda ou de webhooks.	23 avril 2020
Machine learning	Ajout d'informations sur la récupération de recommandations personnalisées à partir de modèles de recommandation, et éventuellement sur l'amélioration de ces recommandations à l'aide AWS Lambda de fonctions.	4 mars 2020
Sécurité	Ajout d'un chapitre Sécurité, qui fournit des informations sur les différents contrôles de sécurité et fonctionnalités d'Amazon Pinpoint.	4 février 2020
Parcours	Ajout d'informations sur l'utilisation de parcours Amazon Pinpoint pour développer des flux de travail automatisés qui exécutent des activités de messagerie pour des projets. Ajout d'informations sur l'interrogation des données d'analyse pour un sous-ensemble de métriques qui s'appliquent aux parcours.	31 octobre 2019

Analyse	Ajout de procédures expliquant comment interroger les données d'analyse pour les campagnes et les messages transactionnels, et ajout d'informations sur l'utilisation des résultats de requête.	17 octobre 2019
Analyse	Ajout d'informations sur l'interrogation des données d'analyse pour un sous-ensemble de métriques qui s'appliquent aux e-mails transactionnels et aux SMS.	6 septembre 2019
Exemples de code	Ajout d'exemples de code que vous pouvez utiliser pour envoyer des notifications push transactionnelles à l'aide de tous les services pris en charge par Amazon Pinpoint.	30 juillet 2019
Analyse	Ajout d'informations sur l'interrogation des données d'analyse pour un sous-ensemble de métriques qui s'appliquent aux projets (applications) et aux campagnes.	24 juillet 2019
Segments	Ajout d'un didacticiel qui décrit une solution pour importer les données client dans Amazon Pinpoint à partir de systèmes externes tels que Salesforce ou Marketo.	14 mai 2019

Disponibilité par région	Amazon Pinpoint est désormais disponible dans les régions AWS Asie-Pacifique (Mumbai) et Asie-Pacifique (Sydney).	25 avril 2019
Utiliser Postman avec Amazon Pinpoint	Ajout d'un didacticiel qui explique comment utiliser Postman pour interagir avec l'API Amazon Pinpoint.	8 avril 2019
Balisage	Ajout d'informations sur le balisage des ressources Amazon Pinpoint.	27 février 2019
Inscription par SMS	Ajout d'un chapitre Didacticiels et ajout d'un didacticiel qui explique comment créer une solution qui gère l'inscription par SMS des utilisateurs.	27 février 2019
Exemples de code	Ajout d'exemples de code dans plusieurs langages de programmation qui montrent comment envoyer des e-mails, des SMS et des messages vocaux par programmation.	6 février 2019

Mises à jour antérieures

Le tableau ci-après décrit les modifications importantes apportées dans chaque version du Guide du développeur Amazon Pinpoint avant décembre 2018.

Modifier	Description	Date
Disponibilité par région	Amazon Pinpoint est désormais disponible dans les	21 décembre 2018

Modifier	Description	Date
	régions de l'ouest des AWS États-Unis (Oregon) et de l'Europe (Francfort).	
Canal vocal	Vous pouvez utiliser le nouveau canal vocal Amazon Pinpoint pour créer des messages vocaux et les envoyer à vos clients par téléphone. Actuellement, vous ne pouvez envoyer des messages vocaux qu'en utilisant l'API Amazon Pinpoint de messages SMS et vocaux.	15 novembre 2018
Disponibilité Europe (Irlande)	Amazon Pinpoint est désormais disponible dans la région AWS Europe (Irlande).	le 25 octobre 2018
API d’événements	Utilisez l'API Amazon Pinpoint pour enregistrer des événements et les associer à des points de terminaison.	7 août 2018

Modifier	Description	Date
Exemples de code pour définir et rechercher des points de terminaison	Des exemples de code sont ajoutés pour vous montrer comment définir, mettre à jour, supprimer et recherche r des points de terminais on. Des exemples sont fournis pour le AWS CLI AWS SDK pour Java, et l'API Amazon Pinpoint. Pour de plus amples informations, veuillez consulter Utilisez des points de terminaison pour représenter votre public dans Amazon Pinpoint.	7 août 2018
Autorisations d’exportation de point de terminaison	Configurer une politique IAM qui vous permet d'exporter les points de terminaison Amazon Pinpoint vers un compartiment Amazon S3.	1er mai 2018
Vérification du numéro de téléphone pour SMS	Utilisez l'API Amazon Pinpoint pour vérifier un numéro de téléphone et déterminer s'il s'agit d'une destination valide pour les messages SMS.	23 avril 2018
Rubriques mises à jour pour l'intégration d'Amazon Pinpoint	Intégrez Amazon Pinpoint à votre appareil Android, iOS ou à votre JavaScript application à l'aide de nos bibliothèques AWS SDKs .	23 mars 2018

Modifier	Description	Date
AWS CloudTrail journalisation	Ajout d'informations sur la journalisation des appels d'API Amazon Pinpoint avec. CloudTrail	6 février 2018
Quotas de service mis à jour	Mise à jour des Quotas avec des informations supplémen taires sur les quotas d'e-mails.	19 janvier 2018
Bêta publique pour les extensions Amazon Pinpoint	Utilisez AWS Lambda les fonctions pour personnal iser les segments ou créer des canaux de messagerie personnalisés.	28 novembre 2017
Quotas de charges applicati ves de notification push	Les quotas incluent les tailles de charges applicatives pour les messages push mobiles.	25 octobre 2017
Quotas de service mis à jour	Ajout d'informations sur les canaux SMS et e-mail à Quotas.	9 octobre 2017
Push mobile ADM et Baidu	Mettez à jour le code de votre application de manière à gérer les notifications push à partir des canaux push mobiles Baidu et ADM.	27 septembre 2017

Modifier	Description	Date
Événements relatifs IDs aux utilisateurs et à l'authent ification avec les groupes d'utilisateurs Amazon Cognito.	Si vous utilisez les groupes d'utilisateurs Amazon Cognito pour gérer la connexion des utilisateurs dans vos applications mobiles, Amazon Cognito attribue l'utilisateur IDs aux points de terminais on et signale les événement s d'authentification à Amazon Pinpoint.	26 septembre 2017
Utilisateur IDs	Attribuez IDs un utilisateur à des points de terminaison pour surveiller l'utilisation des applications par les utilisate urs individuels. Des exemples sont fournis pour le AWS Mobile SDKs et le SDK for Java.	31 août 2017
Événements d’authentification	Signalez les événements d’authentification pour savoir à quelle fréquence les utilisate urs s’authentifient auprès de votre application. Des exemples sont fournis dans Signalez les événements Amazon Pinpoint dans votre application.	31 août 2017
Exemples d’événements mis à jour	Les événements exemples incluent les événements qu'Amazon Pinpoint diffuse pour les activités par e-mail et SMS.	08 juin 2017

Modifier	Description	Date
Gestion de session Android	Gérez les sessions dans les applications Android à l'aide d'une classe fournie par l'exemple d'application AWS Mobile Hub .	le 20 avril 2017
Exemples d'événements de monétisation mis à jour	L'exemple de code est mis à jour pour le rapport des événements de monétisation.	31 mars 2017
Flux d'événement	Vous pouvez configurer Amazon Pinpoint pour envoyer les événements d'application et de campagne vers un flux Kinesis.	24 mars 2017
Permissions	Consultez Fonctionnement d'Amazon Pinpoint avec IAM pour en savoir plus sur l'octroi de l'accès à Amazon Pinpoint aux AWS utilisateurs de votre compte et aux utilisateurs de votre application mobile.	12 janvier 2017
Disponibilité générale d'Amazon Pinpoint	Cette version présente Amazon Pinpoint.	1er décembre 2016