



Guide du développeur

Amazon Pinpoint



Amazon Pinpoint: Guide du développeur

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Qu'est-ce qu'Amazon Pinpoint ?	1
Fonctionnalités d'Amazon Pinpoint	1
Définir des segments de public ciblé	1
Atteindre votre public avec des campagnes de communication	2
Envoyer des messages transactionnels	2
Analyser le comportement des utilisateurs	2
Disponibilité par région	2
Didacticiels	4
Utiliser Postman avec Amazon Pinpoint	4
À propos de ce didacticiel	4
Prérequis	5
Étape 1 : créer des politiques et des rôles IAM	6
Étape 2 : Configurer Postman	10
Étape 3 : Envoi de demandes supplémentaires	18
Configuration d'un système d'inscription par SMS	26
À propos de la double inscription	26
À propos de cette solution	27
Prérequis	29
Étape 1 : configurer Amazon Pinpoint	30
Étape 2 : créer des politiques et des rôles IAM	36
Étape 3 : Créer des fonctions Lambda	39
Étape 4 : Configuration d'Amazon API Gateway	51
Étape 5 : Créer et déployer le formulaire web	56
Étapes suivantes	64
Intégration à votre application	68
Utilisation des AWS SDK	69
Connecter votre application frontale avec AWS Amplify	70
Étape suivante	70
Enregistrement des points de terminaison	70
Avant de commencer	71
Kits SDK AWS Mobile	71
AWS Amplify	71
Étapes suivantes	72
Rapport des événements	72

Avant de commencer	73
Kits SDK AWS Mobile	73
Web et React Native	71
API d'événements Amazon Pinpoint	74
Étape suivante	74
Gestion des notifications push	74
Configuration des notifications push	75
Gestion des notifications push	77
Définition de votre public	78
Ajout de points de terminaison	79
Exemples	80
Informations connexes	85
Association des utilisateurs aux points de terminaison	86
Exemples	87
Informations connexes	91
Ajout d'un lot de points de terminaison	92
Exemples	92
Informations connexes	100
Importation de points de terminaison	100
Avant de commencer	100
Exemples	101
Informations connexes	114
Suppression de points de terminaison	114
Exemples	114
Gestion des points de terminaison des membres de l'audience	117
Accès aux données du public	119
Recherche de points de terminaison	120
Exemples	120
Informations connexes	126
Exportation de points de terminaison	126
Avant de commencer	126
Exemples	127
Informations connexes	138
Répertorier les ID de points de terminaison	138
Création de segments	141
Préparation des segments	141

Préparation de segments avec le kit AWS SDK for Java	141
Importation des segments	145
Importation d'un segment	145
Personnalisation de segments avec AWS Lambda	148
Données d'événement	149
Création d'une fonction Lambda	150
Attribution d'une stratégie de fonction Lambda	152
Attribution d'une fonction Lambda à une campagne	155
Création de campagnes	157
Création de campagnes standard	157
Création de campagnes à l'aide du AWS SDK for Java	157
Création de campagnes de tests A/B	160
Création de campagnes de tests A/B avec AWS SDK for Java	160
Utilisation de l'API SMS et voix	163
Envoi et validation de mots de passe uniques	165
Envoi d'un message OTP	165
Réponse d'SendOtpMessage	168
Validation d'un message OTP	169
Réponse d'VerifyOtpMessage	169
Exemples de code	170
Génération d'un ID de référence	170
Envoi de codes OTP	171
Validation des codes OTP	172
Envoyer et récupérer des messages intégrés à l'application	174
Récupération de messages dans l'application pour un point de terminaison	174
Comprendre les réponses de l'API GetInAppMessages	177
Objet InAppMessageCampaigns	178
Objet InAppMessage	180
Objet HeaderConfig	181
Objet BodyConfig	182
Objet InAppMessageContent	182
Objet Schedule	184
Objet InAppMessageButton	184
Objet DefaultButtonConfig	185
Objet OverrideButtonConfig	187
Validation des numéros de téléphone	188

Cas d'utilisation de la validation de numéro de téléphone	188
Utilisation du service de validation de numéro de téléphone	189
Réponses du service de validation de numéro de téléphone	190
Envoi de messages	194
Send email (Envoyer un e-mail)	194
Choix d'une méthode d'envoi d'e-mail	195
Choix entre Amazon Pinpoint et Amazon Simple Email Service (SES)	195
Utilisation de l'API	195
Envoi d'un e-mail avec des en-têtes de désabonnement	210
Envoi de SMS	212
Envoi de messages vocaux	225
Envoi de notifications push	233
Création de canaux personnalisés	243
Création d'une campagne qui envoie des messages à travers un canal personnalisé	243
Comprendre les données d'événement	244
Configuration de webhooks	246
Configuration des fonctions Lambda	246
Exemple de fonction Lambda	246
Format de réponse de fonction Lambda pour Amazon Pinpoint	250
Accorder une autorisation Amazon Pinpoint pour invoquer la fonction Lambda	252
Diffusion en continu d'événements	255
Configuration de la diffusion en continu des événements	256
Prérequis	256
AWS CLI	257
AWS SDK for Java	257
Désactivation de la diffusion en continu des événements	258
Événements d'application	258
Exemple	258
Attributs d'événement d'application	260
Événements de campagne	265
Exemple d'événement	265
Attributs d'événement de campagne	266
Événements de parcours	273
Exemple d'événement	273
Attributs de l'événement de parcours	274
Événements d'e-mail	279

Exemples d'événements	279
Attributs d'événement d'e-mail	286
Événements SMS	293
Exemple	293
Attributs d'événement SMS	294
Interrogation des données d'analyse	305
Métriques prises en charge	305
Principes de base de la requête	307
Politiques IAM	308
Métriques standard	312
Métriques d'application pour les campagnes	313
Métriques d'application pour les e-mails transactionnels	320
Métriques d'application pour les SMS transactionnels	332
Métriques de campagne	339
Métriques d'engagement du parcours	352
Métriques d'exécution du parcours	360
Métriques d'exécution de l'activité du parcours	362
Métriques relatives à l'exécution des parcours et des campagnes	367
Interrogation des données de campagne	371
Prérequis	371
Interrogation de données pour une campagne	373
Interrogation de données pour plusieurs campagnes	378
Interrogation des données de messagerie transactionnelle	384
Prérequis	385
Interrogation de données pour les messages électroniques transactionnels	386
Interrogation de données pour les messages SMS transactionnels	391
Utilisation des résultats des requêtes	397
Structure JSON	398
Objets et champs JSON	403
Journalisation des appels d'API	405
Informations Amazon Pinpoint dans CloudTrail	405
Actions de l'API Amazon Pinpoint qui peuvent être enregistrées par CloudTrail	407
Actions de l'API de messagerie Amazon Pinpoint qui peuvent être enregistrées par CloudTrail	411
Actions de la version 1 de l'API SMS et voix Amazon Pinpoint qui peuvent être enregistrées par CloudTrail	412

Exemples : entrées de fichier journal Amazon Pinpoint	412
Étiquetage des ressources	418
Gestion des balises	418
Utilisation de balises dans les politiques IAM	419
Ajout de balises à des ressources	420
Ajout de balises à l'aide de l'API	420
Ajout de balises à l'aide de AWS CLI	421
Affichage des balises pour des ressources	423
Afficher les balises à l'aide de l'API	423
Afficher les balises à l'aide de AWS CLI	424
Mise à jour des balises pour des ressources	425
Suppression de balises de ressources	425
Suppression de balises à l'aide de l'API	425
Suppression de balises à l'aide de AWS CLI	426
Informations connexes	427
Personnalisation des recommandations avec AWS Lambda	428
Utilisation des recommandations dans les messages	428
Création de la fonction Lambda	431
Données d'événement en entrée	431
Données et exigences de réponse	434
Attribution d'une stratégie de fonction Lambda	438
Autorisation accordée à Amazon Pinpoint d'invoquer la fonction	440
Configuration du modèle de recommandation	441
Suppression des données	442
Suppression de points de terminaison	442
Suppression de données de segment et de point de terminaison d'Amazon S3	443
Suppression de toutes les données du projet	443
Supprimer toutes les AWS données	444
Exemples de code	446
Amazon Pinpoint	447
Actions	448
API de messages SMS et vocaux Amazon Pinpoint	534
Actions	535
Sécurité	544
Protection des données	545
Chiffrement des données	547

Confidentialité du trafic inter-réseau	548
Création d'un point de terminaison de VPC d'interface pour l'API Amazon Pinpoint	549
Gestion des identités et des accès	551
Public ciblé	551
Authentification par des identités	552
Gestion des accès à l'aide de politiques	555
Fonctionnement d'Amazon Pinpoint avec IAM	558
Actions liées aux politiques Amazon Pinpoint	566
Exemples de politiques basées sur l'identité	599
Rôles IAM pour les tâches courantes	613
Résolution des problèmes	631
Journalisation et surveillance	633
Validation de conformité	634
Résilience	636
Sécurité de l'infrastructure	636
Analyse de la configuration et des vulnérabilités	637
Bonnes pratiques de sécurité	638
Quotas	639
Quotas de projet	639
Quotas de demande d'API	640
Quotas de campagne	642
Quotas d'e-mails	645
Quotas de messages électroniques	645
Quotas d'expéditeurs et de destinataires d'e-mails	645
Quotas d'envoi d'e-mails	647
Quotas de points de terminaison	648
Quotas d'importation de points de terminaison	649
Quotas d'ingestion d'événements	650
Quotas de parcours	651
Quotas Lambda	652
Quotas de machine learning	653
Quotas de modèles de message	655
Quotas de notifications push	656
Quotas de messages dans l'application	657
Quotas de segments	658
Quotas de SMS	658

Quotas de 10DLC	661
Quotas de messages vocaux	661
Demande d'augmentation de quota	664
Historique de la documentation	667
Mises à jour antérieures	676
.....	dclxxxii

Qu'est-ce qu'Amazon Pinpoint ?

Amazon Pinpoint est un service AWS que vous pouvez utiliser pour échanger avec vos clients sur 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](#).

Plusieurs autres documents accompagnent ce document. Les documents suivants fournissent des informations de référence relatives à l'API Amazon Pinpoint :

- [Référence de l'API Amazon Pinpoint](#)
- [API SMS et voix Amazon Pinpoint](#)

Si vous n'êtes pas encore familiarisé avec Amazon Pinpoint, il peut être utile de lire le [Guide de l'utilisateur Amazon Pinpoint](#) avant de poursuivre la lecture du présent document.

Fonctionnalités d'Amazon Pinpoint

Cette section décrit les principales fonctionnalités d'Amazon Pinpoint, ainsi que les tâches qu'elles vous permettent d'effectuer.

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.

Atteindre votre public avec des campagnes de communication

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.

Analyser le comportement des utilisateurs

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.

Disponibilité par région

Amazon Pinpoint est disponible dans plusieurs régions AWS en Amérique du Nord, en Europe, en Asie et en Océanie. Dans chaque région, AWS dispose de plusieurs zones de disponibilité. Ces zones de disponibilité sont physiquement isolées mais sont reliées par des connexions réseau privées, à latence faible, à débit élevé et à forte redondance. Ces zones de disponibilité nous permettent de fournir des niveaux très élevés de disponibilité et de redondance, tout en réduisant au minimum la latence.

Pour en apprendre davantage sur les Régions AWS, consultez [Gestion des 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](#).

Didacticiels

Les tutoriels de cette section expliquent aux utilisateurs d'Amazon Pinpoint comment effectuer plusieurs tâches importantes. Si vous ne connaissez pas encore Amazon Pinpoint ou si certaines fonctionnalités vous sont encore inconnues, ces tutoriels sont un bon point de départ.

Les didacticiels de ce guide incluent des tâches destinées à un public de développeurs ou d'administrateurs systèmes. Ces tutoriels montrent comment exécuter des tâches à l'aide de l'API Amazon Pinpoint, des kits SDK AWS et de l'AWS CLI. Si vous interagissez principalement avec Amazon Pinpoint à l'aide de la console web, veuillez consulter la section Tutoriels du Guide de l'utilisateur Amazon Pinpoint.

Didacticiels

- [Tutoriel : utilisation de Postman avec l'API Amazon Pinpoint](#)
- [Didacticiel : configuration d'un système d'inscription par SMS](#)

Tutoriel : utilisation de Postman avec l'API Amazon Pinpoint

Postman est un outil couramment utilisé pour tester les API dans un environnement graphique simple à utiliser. Vous pouvez utiliser Postman pour envoyer des demandes d'API à n'importe quelle API REST et recevoir des réponses à vos demandes. Utiliser Postman est un moyen pratique pour tester et dépanner les appels à l'API Amazon Pinpoint que vous effectuez. Ce tutoriel inclut les procédures de configuration et d'utilisation de Postman avec Amazon Pinpoint.

Note

Postman est développé par une entreprise tierce. Il n'est ni développé ni pris en charge par Amazon Web Services (AWS). Pour en savoir plus sur l'utilisation de Postman ou pour obtenir de l'aide sur des problèmes liés à Postman, consultez le [Centre de support](#) sur le site web de Postman.

À propos de ce didacticiel

Cette section contient une vue d'ensemble de ce didacticiel.

Public visé

Ce didacticiel est destiné aux développeurs et aux personnes chargées de l'implémentation du système. Vous n'avez pas besoin d'être familiarisé avec Amazon Pinpoint ou Postman pour effectuer les étapes de ce tutoriel. Vous devez être à l'aise avec la gestion des politiques IAM et la modification des exemples de code JSON.

Les procédures décrites dans ce tutoriel ont été conçues pour empêcher les nouveaux utilisateurs d'utiliser les opérations d'API qui peuvent supprimer définitivement les ressources Amazon Pinpoint. Les utilisateurs avancés peuvent supprimer cette restriction en modifiant la stratégie associée à leurs utilisateurs .

Fonctionnalités utilisées

Ce tutoriel inclut des exemples d'utilisation pour les fonctionnalités Amazon Pinpoint suivantes :

- Interaction avec l'API Amazon Pinpoint à l'aide de Postman

Temps requis

Ce didacticiel dure environ 15 minutes.

Restrictions régionales

Il n'y a pas de restrictions régionales associées à l'utilisation de cette solution.

Coûts d'utilisation des ressources

Il n'y a pas de frais pour la création d'un compte AWS. Toutefois, lorsque vous mettez en œuvre cette solution, vous pouvez encourir des frais d'utilisation AWS si vous utilisez Postman pour effectuer l'une des opérations suivantes :

- Envoyer des e-mails, des SMS, des push mobile ou des messages vocaux
- Créer et envoyer des campagnes
- Utiliser la fonction de validation de numéro de téléphone

Pour plus d'informations sur les frais associés à l'utilisation d'Amazon Pinpoint, consultez [Tarification d'Amazon Pinpoint](#).

Prérequis

Avant de commencer le didacticiel, suivez les conditions préalables requises :

- Vous devez posséder un compte AWS. Pour créer un compte AWS, rendez-vous sur <https://console.aws.amazon.com/> et choisissez Créer un compte AWS.
- Assurez-vous que le compte que vous utilisez pour vous connecter à AWS Management Console est en mesure de créer de nouvelles politiques et de nouveaux rôles IAM.
- Assurez-vous d'avoir créé au moins un exemple de projet dans lequel les e-mails sont activés et dont l'identité e-mail est vérifiée. Consultez [Création d'un projet Amazon Pinpoint avec assistance par e-mail](#) dans le Guide de l'utilisateur Amazon Pinpoint.
- Assurez-vous d'avoir un ID de compte AWS. Votre ID de compte AWS se trouve dans le coin supérieur droit de la console ou vous pouvez utiliser l'interface de ligne de commande (CLI). Consultez [Rechercher votre ID de compte AWS](#).
- Vous devez télécharger et installer Postman sur votre ordinateur. Vous pouvez télécharger Postman à partir du [site web de Postman](#).
- Une fois que vous avez installé Postman sur votre ordinateur, créez un compte Postman. Lorsque vous démarrez l'application Postman pour la première fois, vous êtes invité à vous connecter ou à créer un nouveau compte. Suivez les instructions fournies par Postman pour vous connecter à votre compte ou pour créer un compte si vous n'en avez pas déjà un.

Étape 1 : créer des politiques et des rôles IAM

Lorsque vous utilisez Postman pour tester l'API Amazon Pinpoint, la première étape consiste à créer un utilisateur. Dans cette section, vous créez une politique qui permet aux utilisateurs d'interagir avec toutes les ressources Amazon Pinpoint. Ensuite, vous créez un utilisateur et attachez la politique directement à l'utilisateur.

Créer une politique IAM

Découvrez comment créer une politique IAM. Les utilisateurs et les rôles qui utilisent cette politiques peuvent interagir avec toutes les ressources de l'API Amazon Pinpoint. Elle permet également d'accéder aux ressources associées à l'API Amazon Pinpoint Email, ainsi qu'à l'API Amazon Pinpoint SMS et voix.

Pour créer la politique

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le volet de navigation, sélectionnez Politiques, puis Créer une politique.

3. Dans l'éditeur de politiques, sélectionnez JSON. Supprimez tout JSON actuel dans l'éditeur de politiques afin qu'il soit vide. Copiez et collez le code JSON suivant dans l'éditeur de stratégie, puis remplacez toutes les instances de `123456789012` par votre identifiant dans l'éditeur de stratégie. Compte AWS

Votre Compte AWS identifiant se trouve dans le coin supérieur droit de la console, ou vous pouvez utiliser la CLI, voir [Trouver l'identifiant de votre AWS compte](#).

Note

Pour protéger les données de votre compte Amazon Pinpoint, cette politique inclut uniquement les autorisations qui vous permettent de consulter, créer et modifier des ressources. Elle n'inclut pas les autorisations qui vous permettent de supprimer les ressources. Vous pouvez modifier cette politique à l'aide de l'éditeur visuel dans la console IAM. Pour plus d'informations, consultez [Gestion des politiques IAM](#) dans le Guide de l'utilisateur IAM. Vous pouvez également utiliser l'[CreatePolicyVersion](#) opération dans l'API IAM pour mettre à jour cette politique.

Notez également que cette politique inclut les autorisations qui vous permettent d'interagir avec les services `ses` et `sms-voice`, en plus du service `mobiletargeting`. Les autorisations `ses` et `sms-voice` vous permettent d'interagir avec l'API Amazon Pinpoint Email et l'API Amazon Pinpoint SMS et voix, respectivement. Les autorisations `mobiletargeting` vous permettent d'interagir avec l'API Amazon Pinpoint.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Update*",
        "mobiletargeting:Get*",
        "mobiletargeting:Send*",
        "mobiletargeting:Put*",
        "mobiletargeting:Create*"
      ],
      "Resource": [
        "arn:aws:mobiletargeting:*:123456789012:apps/*",

```

```

        "arn:aws:mobiletargeting:*:123456789012:apps/*/campaigns/*",
        "arn:aws:mobiletargeting:*:123456789012:apps/*/segments/*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "mobiletargeting:TagResource",
        "mobiletargeting:PhoneNumberValidate",
        "mobiletargeting:ListTagsForResource",
        "mobiletargeting:CreateApp"
    ],
    "Resource": "arn:aws:mobiletargeting:*:123456789012:*"
},
{
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": [
        "ses:TagResource",
        "ses:Send*",
        "ses:Create*",
        "ses:Get*",
        "ses:List*",
        "ses:Put*",
        "ses:Update*",
        "sms-voice:SendVoiceMessage",
        "sms-voice:List*",
        "sms-voice:Create*",
        "sms-voice:Get*",
        "sms-voice:Update*"
    ],
    "Resource": "*"
}
]
}

```

Choisissez Suivant.

4. Dans Nom de la stratégie, entrez un nom pour la stratégie, tel que **PostmanAccessPolicy**. Choisissez Créer une politique.
5. (Facultatif) Vous pouvez ajouter des balises à la politique en sélectionnant Ajouter une balise.
6. Choisissez Suivant : vérification.

Créer un utilisateur IAM

Warning

Les utilisateurs IAM disposent d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces utilisateurs lorsqu'ils ne sont plus nécessaires.

Une fois que vous avez créé la politique, vous pouvez créer un utilisateur et lui attacher la politique. Lorsque vous créez l'utilisateur, IAM fournit un ensemble d'informations d'identification pour autoriser Postman à exécuter des opérations d'API Amazon Pinpoint.

Pour créer l'utilisateur

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Sur la console IAM, dans le volet de navigation, choisissez Users, puis Create users.
3. Sous Informations utilisateur, pour Nom utilisateur, saisissez un nom qui identifie l'utilisateur, par exemple **PostmanUser**. Ensuite, sélectionnez Suivant.
4. Sous Définir les autorisations, pour les Options d'autorisations, choisissez Attacher directement les politiques.
5. Sous Politiques d'autorisations, choisissez la stratégie (**PostmanAccessPolicy**) que vous avez créée dans [Créer une stratégie IAM](#). Ensuite, sélectionnez Suivant.
6. Sur la page Examiner et créer, ajoutez, si vous le souhaitez, des balises qui vous aident à identifier l'utilisateur. Pour plus d'informations sur l'utilisation des balises, consultez la rubrique [Balisage des ressources IAM](#) dans le Guide de l'utilisateur IAM.
7. Lorsque vous êtes prêt à créer l'utilisateur, choisissez Create user (Créer un utilisateur).

Création de clés d'accès

Warning

Ce scénario nécessite que les utilisateurs IAM disposent d'un accès programmatique et d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces utilisateurs

lorsqu'ils ne sont plus nécessaires. Les clés d'accès peuvent être mises à jour si nécessaire. Pour plus d'informations, consultez la section [Mise à jour des clés d'accès](#) dans le guide de l'utilisateur IAM.

IAM fournit un ensemble d'informations d'identification que vous pouvez utiliser pour autoriser Postman à exécuter des opérations d'API Amazon Pinpoint.

Pour créer l'utilisateur

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation de la console IAM, sélectionnez Utilisateurs. Sélectionnez l'utilisateur (**PostmanUser**) créé dans [Créer un utilisateur IAM](#), puis sélectionnez l'onglet Informations d'identification de sécurité.
3. Dans la section Clés d'accès, choisissez Créer une clé d'accès.
4. Sur la page Meilleures pratiques et alternatives relatives aux clés d'accès, sélectionnez Application exécutée en dehors AWS.

Ensuite, sélectionnez Suivant.

5. (Facultatif) Vous pouvez ajouter une balise de description à la politique.
6. Choisissez Create access key (Créer une clé d'accès).
7. Sur la page Récupérer les clés d'accès, copiez les informations d'identification qui sont affichées dans les colonnes Clé d'accès et Clé d'accès secrète.

Note

Vous devez fournir l'ID de clé d'accès et la clé d'accès secrète ultérieurement dans ce tutoriel. C'est la seule fois que vous pourrez afficher la clé d'accès secrète. Nous vous recommandons de la copier et de l'enregistrer dans un emplacement sûr.

8. Après avoir enregistré les deux clés, choisissez Terminé.

Étape 2 : Configurer Postman

Maintenant que vous avez créé un utilisateur en mesure d'accéder à l'API Amazon Pinpoint, vous pouvez configurer Postman. Dans cette section, vous allez créer un ou plusieurs environnements

dans Postman. Ensuite, vous importez une collection qui contient un modèle de demande pour chacune des opérations de l'API Amazon Pinpoint.

Créer un espace de travail Postman

Dans Postman, un espace de travail est un conteneur organisationnel pour les projets et les environnements. Dans cette section, vous créez au moins un espace de travail à utiliser avec Amazon Pinpoint.

Création d'un espace de travail

Dans Postman, choisissez le plus d'actions, choisissez Fichier, puis Nouveau.

1. Dans la fenêtre Créer nouveau, choisissez Espace de travail.
2. Entrez un nom, un résumé et définissez la visibilité sur Personnel. Choisissez ensuite Créer un espace de travail.

Créer des environnements Postman


Dans Postman, un environnement est un ensemble de variables qui sont stockées en tant que paires clé-valeur. Vous pouvez utiliser des environnements pour modifier la configuration des demandes que vous effectuez via Postman, sans avoir à modifier les demandes d'API elles-mêmes.

Dans cette section, vous allez créer au moins un environnement à utiliser avec Amazon Pinpoint. Chaque environnement que vous créez contient un ensemble de variables qui sont spécifiques à votre compte dans une seule Région AWS. Si vous utilisez les procédures de cette section pour créer plusieurs environnements, vous pouvez passer d'une région à l'autre en choisissant un autre environnement dans le menu Environnement de Postman.

Pour créer un environnement .

1. Dans Postman, choisissez le menu plus d'actions, choisissez Fichier, puis Nouveau.
2. Dans la fenêtre Create New (Créer), choisissez Environment (Environnement).
3. Dans la fenêtre MANAGE ENVIRONNEMENTS (GÉRER LES ENVIRONNEMENTS), pour Environment Name (Nom de l'environnement), saisissez **Amazon Pinpoint - *Region Name***. Remplacez *Region Name (Nom de la région)* par l'une des valeurs suivantes :
 - USA Est (Virginie du Nord)
 - USA Ouest (Oregon)

- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Sydney)
- Europe (Francfort)
- Europe (Irlande)

 Note

Au minimum, vous ne devez créer qu'un seul environnement pour un seul Région AWS, qui Région AWS doit contenir un seul projet. Si vous n'avez pas créé de projet dans l'une des listes précédentes Régions AWS, consultez la section [Création d'un projet Amazon Pinpoint avec assistance par e-mail](#) dans le guide de l'utilisateur Amazon Pinpoint.

4. Créez six nouvelles variables: `endpoint`, `region`, `serviceName`, `accountId`, `accessKey` et `secretAccessKey`. Utilisez le tableau suivant pour déterminer quelle valeur saisir dans les colonnes Valeur initiale et Valeur actuelle pour chaque variable.

Région	Variable	Valeur initiale et actuelle
USA Est (Virginie du Nord)	<code>endpoint</code>	pinpoint.us-east-1 .amazonaws.com
	<code>region</code>	us-east-1
	<code>serviceName</code>	mobiletargeting
	<code>accountId</code>	(votre identifiant de AWS compte)
	<code>accessKey</code>	(Votre ID de clé d'accès IAM)
	<code>secretAccessKey</code>	(votre clé d'accès secrète IAM)
USA Ouest (Oregon)	<code>endpoint</code>	pinpoint.us-west-2 .amazonaws.com

Région	Variable	Valeur initiale et actuelle
	region	us-west-2
	serviceName	mobiletargeting
	accountId	(votre identifiant de AWS compte)
	accessKey	(Votre ID de clé d'accès IAM)
	secretAccessKey	(votre clé d'accès secrète IAM)
Asie-Pacifique (Mumbai)	endpoint	pinpoint.ap-south-1.amazonaws.com
	region	ap-south-1
	serviceName	mobiletargeting
	accountId	(votre identifiant de AWS compte)
	accessKey	(Votre ID de clé d'accès IAM)
	secretAccessKey	(votre clé d'accès secrète IAM)
Asie-Pacifique (Sydney)	endpoint	pinpoint.ap-southeast-2.amazonaws.com
	region	ap-southeast-2
	serviceName	mobiletargeting

Région	Variable	Valeur initiale et actuelle
	accountId	(votre identifiant de AWS compte)
	accessKey	(Votre ID de clé d'accès IAM)
	secretAccessKey	(votre clé d'accès secrète IAM)
Europe (Francfort)	endpoint	pinpoint.eu-central-1-1.amazonaws.com
	region	eu-central-1
	serviceName	mobiletargeting
	accountId	(votre identifiant de AWS compte)
	accessKey	(Votre ID de clé d'accès IAM)
	secretAccessKey	(votre clé d'accès secrète IAM)
Europe (Irlande)	endpoint	pinpoint.eu-west-1.amazonaws.com
	region	eu-west-1
	serviceName	mobiletargeting
	accountId	(votre identifiant de AWS compte)
	accessKey	(Votre ID de clé d'accès IAM)

Région	Variable	Valeur initiale et actuelle
	secretAccessKey	(votre clé d'accès secrète IAM)

Une fois que vous avez créé ces variables, la fenêtre MANAGE ENVIRONNEMENTS (GÉRER LES ENVIRONNEMENT) ressemble à l'exemple présenté dans l'image suivante.

US East (N. Virginia) Fork 0 Save Share ...

	VARIABLE	TYPE ⓘ	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	endpoint	default ▾	pinpoint.us-east-1.amazonaws.com	pinpoint.us-east-1.amazonaws.com			🗑️ ...
<input checked="" type="checkbox"/>	region	default ▾	us-east-1	us-east-1			
<input checked="" type="checkbox"/>	serviceName	default ▾	mobiletargeting	mobiletargeting			
<input checked="" type="checkbox"/>	accountId	default ▾	123456789012	123456789012			
<input checked="" type="checkbox"/>	accessKey	default ▾	AKIAIOSFODNN7EXAMPLE	AKIAIOSFODNN7EXAMPLE			
<input checked="" type="checkbox"/>	secretAccessKey	default ▾	wJairXUtnFEMI/K7MDENG/bPxrFiCY...	wJairXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY			

Lorsque vous avez terminé, choisissez Enregistrer.

Important

Les clés d'accès illustrées dans l'image précédente sont fictives. Ne partagez pas vos clés d'accès IAM avec d'autres personnes.

Postman inclut des fonctionnalités qui vous permettent de partager et d'exporter des environnements. Si vous utilisez ces fonctionnalités, veillez à ne pas partager votre ID de clé d'accès et votre clé d'accès secrète avec une personne qui ne devrait pas avoir accès à ces informations d'identification.

Pour plus d'informations, consultez [Bonnes pratiques IAM](#) dans le Guide de l'utilisateur IAM.

- (Facultatif) Répétez les étapes 1 à 4 pour chaque environnement supplémentaire que vous souhaitez créer.

Tip

Dans Postman, vous pouvez créer autant d'environnements que vous le souhaitez. Vous pouvez utiliser les environnements des manières suivantes :

- Créer un environnement distinct pour chaque région dans laquelle vous devez tester l'API Amazon Pinpoint.
- Créer des environnements qui sont associés à d'autres Comptes AWS.
- Créer des environnements qui utilisent les informations d'identification associées à un autre utilisateur.

6. Lorsque vous avez fini de créer des environnements, passez à la section suivante.

Créer une collection Amazon Pinpoint dans Postman

Dans Postman, une collection est un groupe de demandes d'API. Les demandes d'une collection ont généralement un objectif commun qui les unit. Dans cette section, vous créez une nouvelle collection qui contient un modèle de demande pour chaque opération dans l'API Amazon Pinpoint.

Pour créer la collection Amazon Pinpoint

1. Dans Postman, choisissez le menu plus d'actions, choisissez Fichier, puis Importer.
2. Dans la fenêtre d'importation, choisissez Importer depuis un lien, puis entrez l'URL suivante : https://raw.githubusercontent.com/awsdocs/amazon-pinpoint-developer-guide/master/Amazon%20Pinpoint.postman_collection.json.

Choisissez Import (Importer). Postman importe la collection Amazon Pinpoint, qui contient 120 exemples de demande.

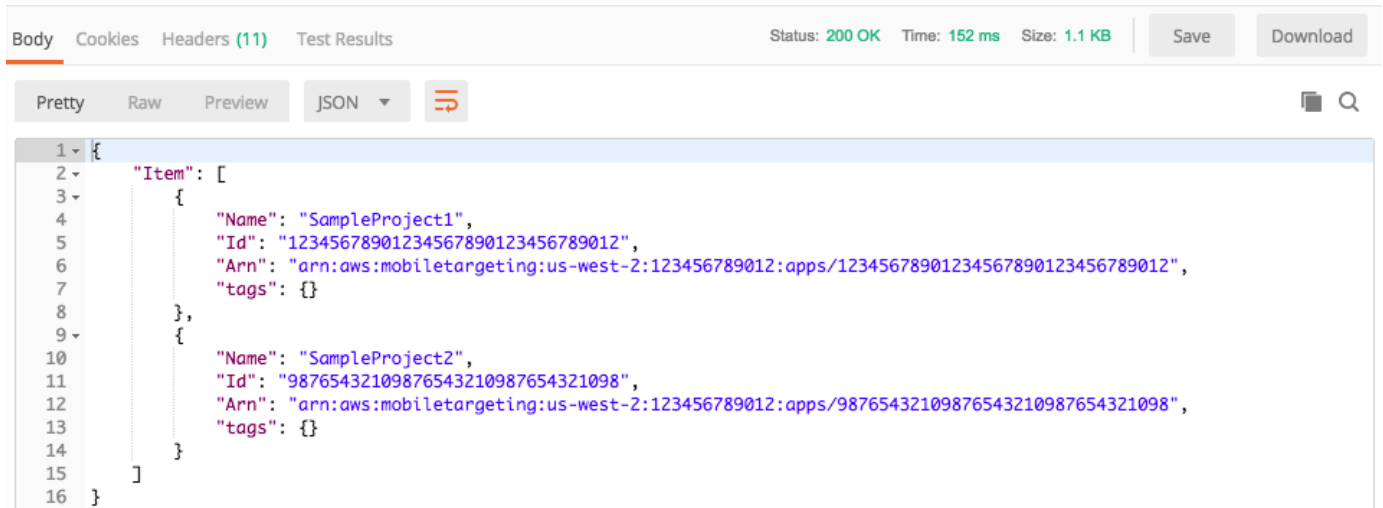
Test de la configuration de Postman

Une fois que vous avez importé la collection Amazon Pinpoint, nous recommandons d'effectuer un test rapide pour vérifier que tous les composants sont correctement configurés. Vous pouvez tester votre configuration en soumettant une demande GetApps. Cette demande renvoie une liste de tous les projets qui existent dans votre compte Amazon Pinpoint dans la région Région AWS actuelle. Cette demande ne nécessite aucune configuration supplémentaire. C'est donc un bon moyen de tester votre configuration.

Pour tester la configuration de la collection Amazon Pinpoint

1. Dans le volet de navigation de gauche, sélectionnez Collections, développez la collection Amazon Pinpoint, puis développez le dossier Apps.

2. Dans la liste des demandes, sélectionnez GetApps.
3. Utilisez le sélecteur d'environnement pour choisir l'environnement que vous avez créé dans les environnements [Create Postman](#).
4. Sélectionnez Send (Envoyer). Si la demande est envoyée avec succès, le volet de réponse indique le statut 200 OK. Une réponse semblable à l'exemple présenté dans l'image ci-dessous s'affiche.



```
Body Cookies Headers (11) Test Results Status: 200 OK Time: 152 ms Size: 1.1 KB Save Download
Pretty Raw Preview JSON ↻
1 {
2   "Item": [
3     {
4       "Name": "SampleProject1",
5       "Id": "12345678901234567890123456789012",
6       "Arn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/12345678901234567890123456789012",
7       "tags": {}
8     },
9     {
10      "Name": "SampleProject2",
11      "Id": "98765432109876543210987654321098",
12      "Arn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/98765432109876543210987654321098",
13      "tags": {}
14    }
15  ]
16 }
```

Note

Si aucun projet n'a été créé dans la Région AWS, Amazon Pinpoint renvoie `{ "Item": [] }`

Cette réponse montre une liste de tous les projets Amazon Pinpoint qui existent dans votre compte dans la région que vous avez sélectionnée à l'étape 3.

Résolution des problèmes

Lorsque vous soumettez votre demande, vous pouvez voir un message d'erreur. La liste suivante contient plusieurs erreurs courantes que vous pouvez rencontrer et les étapes que vous pouvez effectuer pour les résoudre.

Message d'erreur	Problème	Résolution
<p>Could not get any response</p> <p>There was an error connecting to https://%7B%7Bendpoint%7D%7D/v1/apps.</p>	<p>Il n'y a pas de valeur actuelle pour la <code>{{endpoint}}</code> variable qui est définie lorsque vous choisissez un environnement.</p>	<p>Utilisez le sélecteur d'environnements pour choisir un environnement.</p>
<p>The security token included in the request is invalid.</p>	<p>Postman n'a pas été en mesure d'identifier la valeur actuelle de votre ID de clé d'accès ou de clé d'accès secrète.</p>	<p>Choisissez l'icône d'engrenage située près du sélecteur d'environnements, puis sélectionnez l'environnement actuel. Vérifiez que les valeurs <code>accessKey</code> et <code>secretAccessKey</code> s'affichent dans les colonnes VALEUR INITIALE et VALEUR ACTUELLE et que vous avez saisi les informations d'identification correctement.</p>
<p>« Message » : « L'utilisateur : arn:aws:iam : :123456789012:user/ n'PinpointPostmanUserest pas autorisé à effectuer : mobiletargeting : on resource : arn:aws:mobiletargeting:us-west-2:123456789012 :* » GetApps</p>	<p>La politique IAM associée à votre utilisateur n'inclut pas les autorisations appropriées.</p>	<p>Vérifiez que votre utilisateur dispose des autorisations décrites dans Créer une politique IAM et que vous avez fourni les informations d'identification correctes lorsque vous avez créé l'environnement dans l'espace de travail Create Postman.</p>

Étape 3 : Envoi de demandes supplémentaires

Lorsque vous avez terminé la configuration et le test de Postman, vous pouvez commencer à envoyer des demandes supplémentaires à l'API Amazon Pinpoint. Cette section présente les informations que vous devez connaître avant de commencer à envoyer des demandes. Elle inclut

également deux exemples de demandes qui décrivent comment utiliser la collection Amazon Pinpoint.

Important

Lorsque vous effectuez les procédures de cette section, vous soumettez des demandes à l'API Amazon Pinpoint. Ces demandes permettent de créer de nouvelles ressources dans votre compte Amazon Pinpoint, de modifier des ressources existantes, d'envoyer des messages, de modifier la configuration de vos projets Amazon Pinpoint et d'utiliser d'autres fonctionnalités d'Amazon Pinpoint. Soyez prudent lorsque vous exécutez ces demandes.

À propos des exemples dans la collection Postman d'Amazon Pinpoint

Vous devez configurer la plupart des opérations dans la collection Postman d'Amazon Pinpoint avant de pouvoir les utiliser. Pour les opérations GET et DELETE, vous devez généralement modifier uniquement les variables qui sont définies dans l'onglet Pre-request Script (Script de pré-demande).

Note

Lorsque vous utilisez la politique IAM présentée dans [Créer une stratégie IAM](#), vous ne pouvez exécuter aucune des DELETE demandes incluses dans cette collection.

Par exemple, l'opération GetCampaign nécessite que vous spécifiez un `projectId` et un `campaignId`. Dans l'onglet Pre-request Script (Script de pré-demande), ces deux variables sont présentes et sont renseignées avec des exemples de valeurs. Supprimez les exemples de valeurs et remplacez-les par les valeurs applicables pour votre projet et votre campagne Amazon Pinpoint.

Parmi ces variables, la plus couramment utilisée est la variable `projectId`. Sa valeur doit être l'identifiant unique du projet concerné par votre demande. Pour obtenir la liste de ces identifiants pour vos projets, vous pouvez vous référer à la réponse à la demande GetApps que vous avez envoyée à l'étape précédente de ce tutoriel. Dans cette réponse, le champ `Id` fournit l'identifiant unique d'un projet. Pour en savoir plus sur l'opération GetApps et la signification de chaque champ de la réponse, consultez [Applications](#) dans la Référence de l'API Amazon Pinpoint.

Note

Dans Amazon Pinpoint, un « projet » est identique à une « application » ou à une « application ».

Pour les opérations POST et PUT, vous devez également modifier le corps de la demande afin d'inclure les valeurs que vous souhaitez envoyer à l'API. Par exemple, lorsque vous soumettez une demande `CreateApp`, qui est une demande POST, vous devez spécifier un nom pour le projet que vous créez. Vous pouvez modifier la demande sur l'onglet Body (Corps). Dans cet exemple, remplacez la valeur située à côté de "Name" par le nom du projet. Si vous souhaitez ajouter des balises au projet, vous pouvez les spécifier dans l'objet `tags`. Si vous ne souhaitez pas ajouter de balises, vous pouvez supprimer l'intégralité de l'objet `tags`.

Note

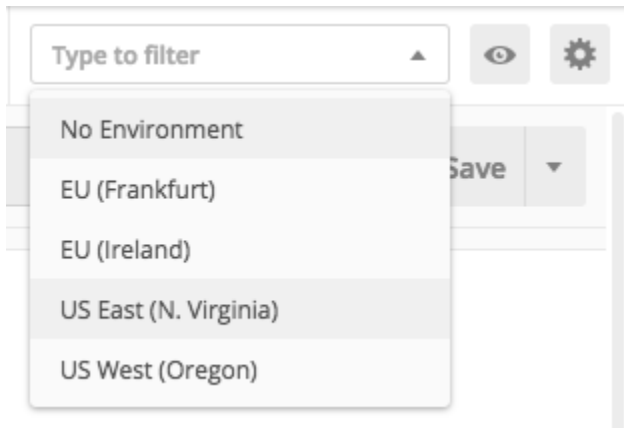
L'opération `UntagResource` nécessite également que vous spécifiez les paramètres d'URL. Vous pouvez spécifier ces paramètres dans l'onglet Params (Paramètres). Remplacez les valeurs de la colonne VALUE (VALEUR) par les balises que vous souhaitez supprimer pour la ressource spécifiée.

Exemple de demande : création d'un projet à l'aide de l'opération `CreateApp`

Avant de créer des segments et des campagnes dans Amazon Pinpoint, vous devez d'abord créer un projet. Dans Amazon Pinpoint, un projet se compose de segments, campagnes, configurations et données unis par un objectif commun. Par exemple, vous pouvez utiliser un projet pour contenir l'ensemble du contenu qui est lié à une application particulière ou à une initiative marketing ou marque spécifique. Lorsque vous ajoutez les informations client à Amazon Pinpoint, ces informations sont associées à un projet.

Pour créer un projet en envoyant une demande d' `CreateApp` API

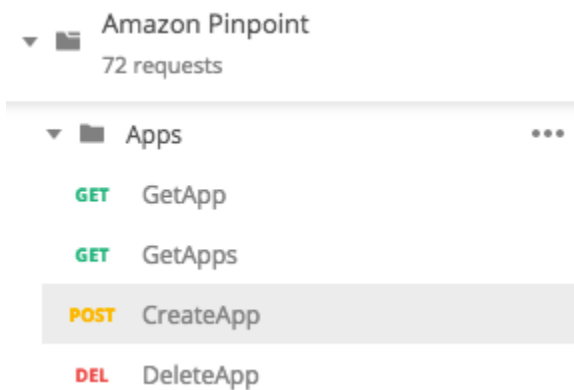
1. Dans le menu Environnements, choisissez Région AWS celui dans lequel vous souhaitez créer le projet.



Dans cet exemple, Postman a été configuré de telle sorte que le menu Environments (Environnements) affiche les quatre options suivantes :

- USA Est (Virginie du Nord)
- USA Ouest (Oregon)
- Europe (Francfort)
- Europe (Irlande)

2. Dans le dossier Apps, choisissez l'CreateAppopération>.



Le dossier Apps de la collection Amazon Pinpoint Postman est agrandi et affiche les demandes suivantes :

- GetApp
- GetApps
- CreateApp
- DeleteApp

3. Dans l'onglet Body (Corps), en regard de "Name", remplacez la valeur d'espace réservé ("string") par un nom pour la campagne, tel que **"MySampleProject"**.
4. Supprimez la virgule après le nom de la campagne, puis supprimez l'intégralité de l'objet tags sur les lignes 3 à 5. Lorsque vous avez terminé, votre demande devrait ressembler à l'exemple qui est illustré dans l'extrait de code suivant.

```
{
  "Name": "MySampleProject"
}
```

Postman est configuré pour envoyer la demande en tant que charge utile JSON brute.

5. Sélectionnez Send (Envoyer). Si la campagne est créée avec succès, le volet de réponse indique le statut 201 Created.

```
{
  "Name": "MySampleProject"
  "Id": "12345678901234567890123456789012",
  "Arn": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/12345678901234567890123456789012",
  "tags": {}
}
```

Exemple : envoi d'un e-mail à l'aide de l'opération **SendMessage**

Il est très courant d'utiliser l'API Amazon Pinpoint `SendMessage` pour envoyer des messages transactionnels. Un avantage de l'envoi de messages à l'aide de l'API `SendMessage`, par opposition à la création de campagnes, est que vous pouvez envoyer des messages à n'importe quelle adresse (par exemple, une adresse e-mail, un numéro de téléphone ou un jeton d'appareil). L'adresse à laquelle vous envoyez des messages ne doit pas déjà exister dans votre compte Amazon Pinpoint. Comparons cette méthode à l'envoi de messages en créant des campagnes. Avant d'envoyer une campagne dans Amazon Pinpoint, vous devez ajouter des points de terminaison à votre compte Amazon Pinpoint, créer des segments, créer la campagne et la mener à bien.

L'exemple de cette section vous montre comment envoyer un e-mail transactionnel directement à une adresse e-mail spécifique. Vous pouvez modifier cette demande pour envoyer des messages via d'autres canaux, comme les SMS, push mobile ou la messagerie vocale.

Pour envoyer un e-mail en soumettant une SendMessages demande

1. Vérifiez que le canal de messagerie est activé pour le projet et que l'adresse e-mail ou le domaine que vous souhaitez utiliser pour envoyer et recevoir le message est configuré. Pour plus d'informations, consultez les sections [Activation et désactivation du canal e-mail](#) et [Vérification des identités e-mail](#) dans le Guide de l'utilisateur Amazon Pinpoint.

Note

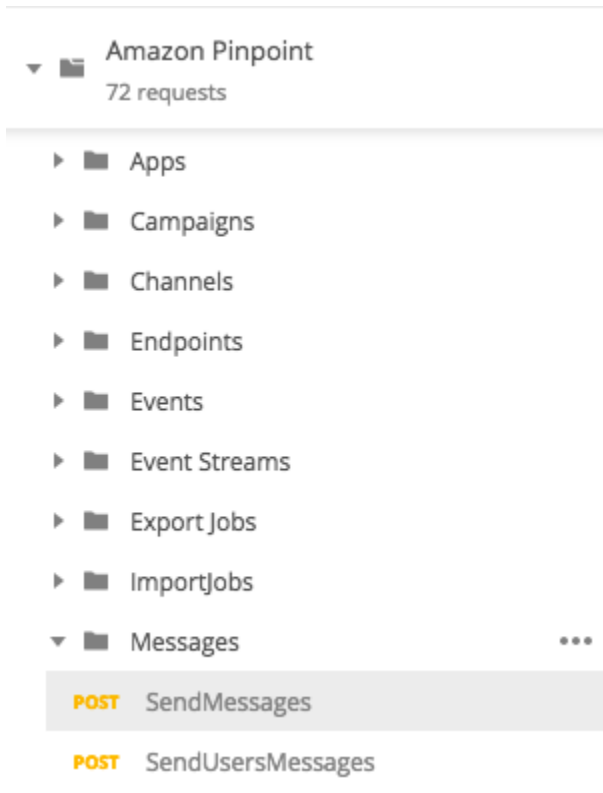
Pour compléter la procédure de cette section, vous devez vérifier votre adresse e-mail

2. Dans le menu Environnements, choisissez Région AWS celui à partir duquel vous souhaitez envoyer le message.

Dans cet exemple, Postman a été configuré de telle sorte que le menu Environments (Environnements) affiche les quatre options suivantes :

- USA Est (Virginie du Nord)
- USA Ouest (Oregon)
- Europe (Francfort)
- Europe (Irlande)

3. Dans le dossier Messages, choisissez l'SendMessagesopération.



4. Dans l'onglet Pre-request Script (Script de pré-demande), remplacez la valeur de la variable `projectId` par l'ID d'un projet qui existe déjà dans la région que vous avez sélectionnée à l'étape 2 de cette section.
5. Dans l'onglet Body (Corps), supprimez l'exemple de demande qui est affiché dans l'éditeur de demande. Collez le code suivant :

```
{
  "MessageConfiguration":{
    "EmailMessage":{
      "FromAddress":"sender@example.com",
      "SimpleEmail":{
        "Subject":{
          "Data":"Sample Amazon Pinpoint message"
        },
        "HtmlPart":{
          "Data":"<h1>Test message</h1><p>This is a sample message sent
from <a href=\"https://aws.amazon.com/pinpoint\">Amazon Pinpoint</a> using the
SendMessages API.</p>"
        },
        "TextPart":{
          "Data":"This is a sample message sent from Amazon Pinpoint
using the SendMessages API."
        }
      }
    }
  }
}
```

```
    }
  }
},
"Addresses":{
  "recipient@example.com": {
    "ChannelType": "EMAIL"
  }
}
}
```

- Dans le code précédent, remplacez *expéditeur@exemple.com* par votre adresse e-mail vérifiée. Remplacez *recipient@example.com* par l'adresse vérifiée à laquelle vous souhaitez envoyer le message.

Note

Si votre compte se trouve encore dans l'environnement de test (sandbox) Amazon Pinpoint, vous ne pouvez envoyer des e-mails qu'aux adresses ou domaines vérifiés dans votre compte Amazon Pinpoint. Pour plus d'informations sur la suppression de votre compte de l'environnement de test (sandbox), veuillez consulter [Demande d'un accès en production pour les e-mails](#) dans le Guide de l'utilisateur d'Amazon Pinpoint.

- Sélectionnez Send (Envoyer). Si le message est envoyé avec succès, le volet de réponse indique le statut 200 OK.

```
{
  "ApplicationId": "12345678901234567890123456789012",
  "RequestId": "<sampleValue>",
  "Result": {
    "recipient@example.com": {
      "DeliveryStatus": "SUCCESSFUL",
      "StatusCode": 200,
      "StatusMessage": "<sampleValue>",
      "MessageId": "<sampleValue>"
    }
  }
}
```

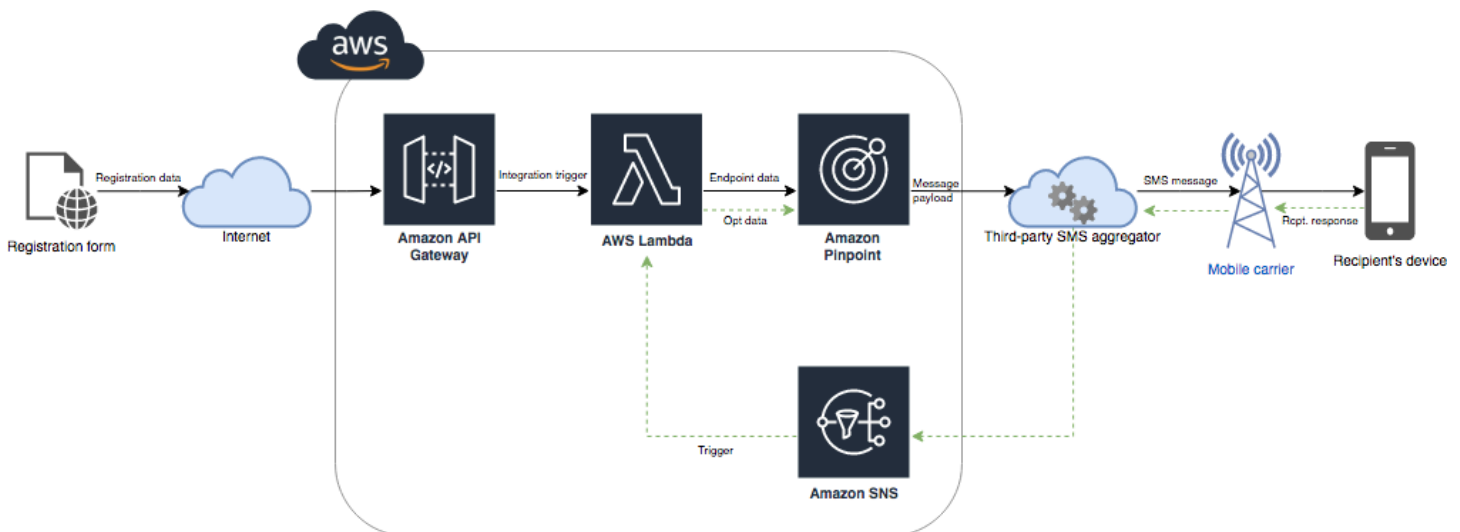
Didacticiel : configuration d'un système d'inscription par SMS

Les messages SMS (messages texte) sont une excellente façon d'envoyer des messages à caractère urgent à vos clients. De nos jours, de nombreuses personnes conservent leur téléphone à portée de main à tout moment. De plus, les messages SMS ont tendance à plus capturer l'attention que les notifications push, les e-mails ou les appels téléphoniques.

Un moyen courant de capturer les numéros des téléphones mobiles des clients consiste à utiliser un formulaire web. Après avoir vérifié le numéro de téléphone du client et confirmé son abonnement, vous pouvez commencer à envoyer des messages SMS promotionnels, transactionnels et informationnels à ce client.

Ce didacticiel explique comment configurer un formulaire web pour capturer les informations de contact des clients. Le formulaire web envoie ces informations à Amazon Pinpoint. Ensuite, Amazon Pinpoint vérifie que le numéro de téléphone est valide et capture les autres métadonnées relatives au numéro de téléphone. Après cela, Amazon Pinpoint envoie au client un message lui demandant de confirmer son abonnement. Une fois que le client a confirmé son abonnement, Amazon Pinpoint l'inscrit pour recevoir vos messages.

Le diagramme d'architecture suivant illustre le flux de données dans cette solution.



À propos de la double inscription

Ce tutoriel montre comment configurer un système de double inscription dans Amazon Pinpoint qui utilise des SMS bidirectionnels.

Dans un système de double inscription par SMS, un client vous fournit son numéro de téléphone dans un formulaire web ou dans votre application. Lorsque vous recevez la demande du client, vous créez un point de terminaison dans Amazon Pinpoint. Le nouveau point de terminaison doit être désinscrit de vos communications. Ensuite, vous envoyez un message à ce numéro de téléphone. Dans votre message, vous demandez au destinataire de confirmer son abonnement en répondant à l'aide d'un mot ou d'une phrase spécifique (par exemple, « Oui » ou « Confirmer »). Si le client répond au message avec le mot ou la phrase que vous avez spécifié, vous remplacez le statut du point de terminaison par Inscrit. Sinon, si le client ne répond pas ou s'il répond avec un autre mot ou une autre phrase, vous pouvez laisser le point de terminaison avec un statut Désinscrit.

À propos de cette solution

Cette section contient des informations sur la solution que vous concevez dans ce didacticiel.

Public visé

Ce didacticiel est destiné aux développeurs et personnes chargées de l'implémentation du système. Vous n'avez pas besoin de connaître Amazon Pinpoint pour effectuer les étapes de ce tutoriel. Cependant, vous devez être à l'aise avec la gestion des politiques IAM, la création de fonctions Lambda dans Node.js, et le déploiement de contenu web.

Fonctionnalités utilisées

Ce tutoriel inclut des exemples d'utilisation pour les fonctionnalités Amazon Pinpoint suivantes :

- Envoi de messages SMS transactionnels
- Obtention d'informations sur les numéros de téléphone à l'aide de la validation de numéro de téléphone
- Réception de messages SMS entrant à l'aide de la messagerie SMS bidirectionnelle
- Création de segments dynamiques
- Création de campagnes
- Interaction avec l'API Amazon Pinpoint en utilisant AWS Lambda

Temps requis

Ce didacticiel dure environ une heure. Une fois que vous avez implémenté cette solution, il existe des étapes supplémentaires que vous pouvez effectuer pour affiner la solution en fonction de vos cas d'utilisation uniques.

Restrictions régionales

Ce tutoriel nécessite de louer un code long à l'aide de la console Amazon Pinpoint. Vous pouvez utiliser la console Amazon Pinpoint pour louer des codes longs dédiés qui sont basés dans plusieurs pays. Cependant, seuls les codes longs basés au Canada peuvent être utilisés pour envoyer des messages SMS. (Vous pouvez utiliser des codes longs qui sont basés dans d'autres pays et régions pour envoyer des messages vocaux.)

Nous avons développé les exemples de code de ce didacticiel avec cette restriction à l'esprit. Par exemple, les exemples de code supposent que le numéro de téléphone du destinataire a toujours 10 chiffres et un code de 1. Si vous implémentez cette solution dans des pays ou des régions autres que les États-Unis ou le Canada, vous devez modifier les exemples de code de façon appropriée.

Coûts d'utilisation des ressources

La création d'un AWS compte est gratuite. Cependant, si vous implémentez cette solution, vous risquez de payer les frais suivants :

- **Frais de location des codes longs** : pour suivre ce tutoriel, vous devez louer un code long. Les codes longs basés au Canada coûtent 1,00 \$ par mois.
- **Utilisation de la validation de numéro de téléphone** : la solution de ce tutoriel utilise la fonctionnalité de validation de numéro de téléphone d'Amazon Pinpoint pour vérifier que chaque numéro que vous recevez est valide et correctement formaté, et pour obtenir des informations supplémentaires sur le numéro de téléphone. Vous payez 0,006 USD pour chaque demande de validation de numéro de téléphone.
- **Coûts d'envoi des messages** : la solution de ce tutoriel envoie des messages SMS sortants. Vous payez pour chaque message que vous envoyez via Amazon Pinpoint. Le prix que vous payez pour chaque message dépend du pays ou de la région du destinataire. Si vous envoyez des messages à des destinataires situés aux États-Unis (à l'exception des territoires des États-Unis), vous payez 0,00645 USD par message. Si vous envoyez des messages à des destinataires situés au Canada, vous payez une somme de 0,00109 \$ à 0,02 \$, en fonction de l'emplacement de l'opérateur et du destinataire.
- **Coûts de réception des messages** : cette solution reçoit et traite aussi les messages SMS entrants. Vous payez pour chaque message entrant envoyé à des numéros de téléphone qui sont associés à votre compte Amazon Pinpoint. Le prix que vous payez dépend de l'endroit où est basé le numéro de téléphone du destinataire. Si votre numéro de destinataire est basé aux États-Unis (à l'exception des territoires des États-Unis), vous payez 0,0075 USD par message entrant. Si votre numéro est basé au Canada, vous payez 0,00155 USD par message entrant.

- Utilisation de Lambda : cette solution utilise deux fonctions Lambda qui interagissent avec l'API Amazon Pinpoint. Lorsque vous appelez une fonction Lambda, vous êtes facturé en fonction du nombre de demandes de vos fonctions, pour le temps qu'il faut à votre code pour s'exécuter, et pour la quantité de mémoire que vos fonctions utilisent. Les fonctions de ce tutoriel utilisent très peu de mémoire, et s'exécutent généralement pendant 1 à 3 secondes. Tout ou partie de votre utilisation de cette solution peut entrer dans le cadre du niveau d'utilisation gratuite de Lambda. Pour en savoir plus, consultez [Tarification Lambda](#).
- Utilisation d'API Gateway : le formulaire web de cette solution appelle une API gérée par API Gateway. Pour chaque million d'appels vers API Gateway, vous payez entre 3,50 et 3,70 dollars, selon la région dans laquelle vous AWS utilisez Amazon Pinpoint. Pour plus d'informations, consultez [Tarification d'API Gateway](#).
- Coûts d'hébergement sur le web : cette solution inclut un formulaire web que vous devez héberger sur votre site web. Le prix que vous payez pour l'hébergement de ce contenu dépend de votre fournisseur d'hébergement web.

Note

Tous les tarifs affichés dans cette liste sont exprimés en dollars US (USD).

Suivant : [Prérequis](#)

Prérequis

Avant de commencer ce didacticiel, vous devez satisfaire les prérequis suivants :

- Vous devez avoir un compte AWS. Pour créer un compte AWS, rendez-vous sur <https://console.aws.amazon.com/> et choisissez Créer un compte AWS.
- Le compte que vous utilisez pour vous connecter à AWS Management Console doit être en mesure d'effectuer les tâches suivantes :
 - Créer des politiques et des rôles IAM
 - Créer des projets Amazon Pinpoint
 - Créer des fonctions Lambda
 - Créer des API dans API Gateway

- Vous devez avoir une méthode d'hébergement de pages web, et vous devez savoir comment publier des pages web. Même si vous pouvez utiliser les services AWS pour héberger vos pages web, vous n'êtes pas tenu de le faire.

 Tip

Pour en savoir plus sur l'hébergement des pages web à l'aide des services AWS, veuillez consulter [Hébergement d'un site web statique](#).

Suivant : [Configurer Amazon Pinpoint](#)

Étape 1 : configurer Amazon Pinpoint

La première étape de l'implémentation de cette solution consiste à configurer Amazon Pinpoint. Dans cette section, vous effectuez les opérations suivantes :

- Créer un projet Amazon Pinpoint
- Activer le canal SMS et louer un numéro de téléphone
- Configurer la messagerie SMS bidirectionnelle

Avant de commencer le didacticiel, vous devez consulter les [conditions préalables requises](#).

Créer un projet Amazon Pinpoint

Pour commencer, vous devez créer un projet Amazon Pinpoint. Dans Amazon Pinpoint, un projet se compose de segments, campagnes, configurations et données unis par un objectif commun. Par exemple, vous pouvez utiliser un projet pour contenir l'ensemble du contenu qui est lié à une application particulière ou à une initiative marketing ou marque spécifique. Lorsque vous ajoutez les informations client à Amazon Pinpoint, ces informations sont associées à un projet.

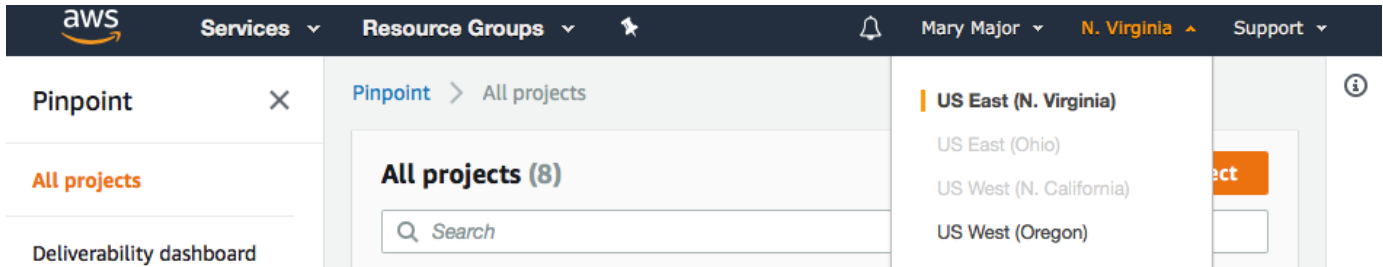
Les étapes impliquées dans la création d'un nouveau projet varient selon que vous avez déjà créé ou pas un projet dans Amazon Pinpoint.

Création d'un projet (nouveaux utilisateurs d'Amazon Pinpoint)

Ces étapes décrivent le processus de création d'un nouveau projet Amazon Pinpoint si vous n'avez jamais créé de projet dans la région actuelle AWS .

Pour créer un projet

1. [Connectez-vous à la console Amazon Pinpoint AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/pinpoint/.](https://console.aws.amazon.com/pinpoint/)
2. Utilisez le sélecteur de région pour choisir la AWS région que vous souhaitez utiliser, comme illustré dans l'image suivante. En cas de doute, choisissez la région la plus proche de vous.



3. Sous Get started (Mise en route), dans Name (Nom), saisissez un nom pour la campagne (comme **SMSRegistration**), puis choisissez Create project (Créer un projet).
4. Sur la page Configure features (Configurer les fonctionnalités), choisissez Skip this step (Ignorer cette étape).
5. Dans le panneau de navigation, choisissez All projects (Tous les projets).
6. Sur la page All projects (Tous les projets), située en regard du projet que vous venez de créer, copiez la valeur illustrée dans la colonne Project ID (ID de projet).

Tip

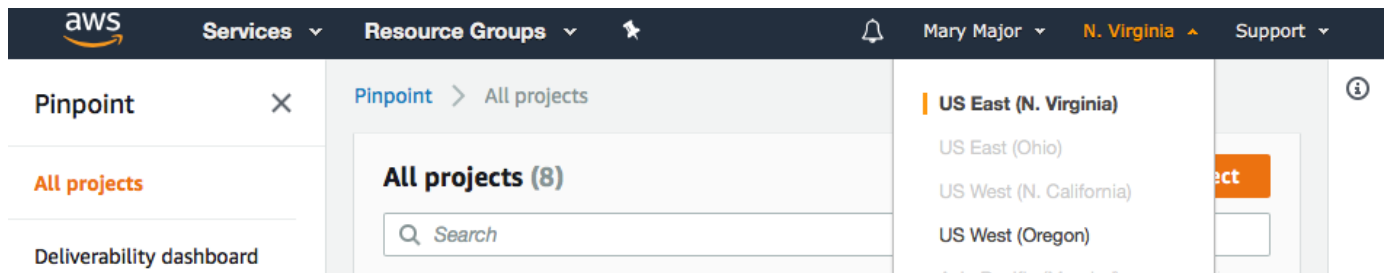
Vous devez utiliser cet ID dans plusieurs endroits différents du didacticiel. Conservez l'ID de projet dans un endroit pratique, afin de pouvoir le copier ultérieurement.

Création d'un projet (utilisateurs Amazon Pinpoint existants)

Ces étapes décrivent le processus de création d'un nouveau projet Amazon Pinpoint si vous avez déjà créé des projets dans la région actuelle AWS .

Pour créer un projet

1. [Connectez-vous à la console Amazon Pinpoint AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/pinpoint/.](https://console.aws.amazon.com/pinpoint/)
2. Utilisez le sélecteur de région pour choisir la AWS région que vous souhaitez utiliser, comme illustré dans l'image suivante. En cas de doute, choisissez la région la plus proche de vous.



3. Sur la page All projects (Tous les projets), sélectionnez Create a project (Créer un projet).
4. Dans la fenêtre Create a project (Créer un projet) , dans Project name (Nom de projet), attribuez un nom au projet (comme **SMSRegistration**). Choisissez Créer.
5. Sur la page Configure features (Configurer les fonctionnalités), choisissez Skip this step (Ignorer cette étape).
6. Dans le panneau de navigation, choisissez All projects (Tous les projets).
7. Sur la page All projects (Tous les projets), située en regard du projet que vous venez de créer, copiez la valeur illustrée dans la colonne Project ID (ID de projet).

Tip


Vous devez utiliser cet ID dans plusieurs endroits différents du didacticiel. Conservez l'ID de projet dans un endroit pratique, afin de pouvoir le copier ultérieurement.

Obtenir un numéro de téléphone dédié

Note

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 Amazon Pinpoint SMS and Voice, consultez le nouveau [Guide de l'utilisateur Amazon Pinpoint SMS](#).

Après que vous avez créé un projet, vous pouvez commencer à configurer les fonctionnalités au sein du projet. Dans cette section, vous activez le canal SMS, et obtenez un numéro de téléphone dédié à utiliser lors de l'envoi de messages SMS.

 Note

Cette section suppose que vous louez un numéro de téléphone 10DLC américain après l'enregistrement de la marque et de la campagne, un numéro gratuit américain ou un code long du Canada. Si vous suivez les procédures décrites dans cette section, mais choisissez un pays autre que les États-Unis ou le Canada, vous ne serez pas en mesure d'utiliser ce numéro pour envoyer des messages SMS. Pour en savoir plus sur la location de codes longs compatibles avec les SMS dans des pays autres que les États-Unis ou le Canada, consultez [Pays et régions pris en charge \(canal SMS\)](#) dans le Guide de l'utilisateur Amazon Pinpoint SMS.

Pour activer le canal SMS à l'aide de la console Amazon Pinpoint, procédez comme suit :


Activer le canal SMS

1. [Connectez-vous à la console Amazon Pinpoint AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/pinpoint/](https://console.aws.amazon.com/pinpoint/).
2. Dans le panneau de navigation, sous Paramètres, choisissez SMS et voix.
3. En regard de SMS settings (Paramètres SMS), choisissez Edit (Modifier).
4. Sous General settings (Paramètres généraux), choisissez Enable the SMS channel for this project (Activer le canal SMS pour ce projet), puis choisissez Save changes (Enregistrer les modifications).

Pour demander un numéro de téléphone à l'aide de la console Amazon Pinpoint SMS, suivez ces étapes :


Demander un numéro de téléphone (console)

1. Ouvrez la console Amazon Pinpoint SMS à l'adresse <https://console.aws.amazon.com/sms-voice/>.

 Note

Assurez-vous de demander le même numéro de téléphone Région AWS que celui dans lequel vous avez créé votre projet Amazon Pinpoint.


2. Dans le volet de navigation, sous Configurations, choisissez Numéros de téléphone, puis Auteur de la demande.
3. Sur la page Sélectionner un pays pour le Pays de destination du message, choisissez les États-Unis ou le Canada. Choisissez Suivant.
4. Dans la section Cas d'utilisation de messagerie, saisissez les informations suivantes :
 - Sous Capacités numériques, choisissez SMS.

 Important

Les fonctionnalités pour SMS et Voix ne peuvent pas être modifiées une fois que le numéro de téléphone a été acheté.

- Pour la Messagerie bidirectionnelle, choisissez Oui.
5. Choisissez Suivant.
 6. Sous Sélectionner le type d'expéditeur, choisissez soit Code long, soit 10DLC.

Si vous choisissez 10DLC et que vous avez déjà une campagne enregistrée, vous pouvez choisir la campagne dans Associer à la campagne enregistrée.
 7. Choisissez Suivant.
 8. Dans Vérifier et demander, vous pouvez vérifier et modifier votre demande avant de l'envoyer. Choisissez Request (Demander).
 9. Une fenêtre Inscription requise peut apparaître en fonction du type de numéro de téléphone que vous avez demandé. Votre numéro de téléphone est associé à cette inscription et vous ne pouvez pas envoyer de messages tant que votre inscription n'a pas été approuvée. Pour plus d'informations sur les exigences d'inscription, consultez [Inscriptions](#).
 - a. Pour Nom du formulaire d'inscription, entrez un nom convivial.
 - b. Choisissez Démarrer l'inscription pour terminer l'inscription du numéro de téléphone, ou choisissez S'inscrire ultérieurement.

 Important

Votre numéro de téléphone ne peut pas envoyer de messages tant que votre inscription n'a pas été approuvée.

La redevance mensuelle pour le numéro de téléphone vous est toujours facturée, quel que soit le statut de l'inscription. Pour plus d'informations sur les exigences d'inscription, consultez [Inscriptions](#).

Activer les SMS bidirectionnels

Maintenant que vous avez un numéro de téléphone dédié, vous pouvez configurer les SMS bidirectionnels. L'activation de la communication bidirectionnelle par SMS permet à vos clients de répondre aux messages SMS que vous leur envoyez. Dans cette solution, vous utilisez les SMS bidirectionnels pour donner à vos clients un moyen de confirmer qu'il souhaite s'abonner à votre programme SMS.

Pour activer les SMS bidirectionnels à l'aide de la console Amazon Pinpoint SMS, procédez comme suit :

Activer les SMS bidirectionnels

1. Ouvrez la console Amazon Pinpoint SMS à l'adresse <https://console.aws.amazon.com/sms-voice/>.
2. Dans le volet de navigation, sous Configurations, choisissez Numéros de téléphone.
3. Sur la page Numéros de téléphone, choisissez un numéro de téléphone.
4. Dans l'onglet SMS bidirectionnels, choisissez le bouton Modifier les paramètres.
5. Sur la page Modifier les paramètres, choisissez Activer les messages bidirectionnels.
6. Pour Type de destination, choisissez Amazon SNS.
 - Nouvelle rubrique Amazon SNS : Amazon Pinpoint SMS crée une rubrique dans votre compte. La rubrique est automatiquement créée avec toutes les autorisations requises. Pour plus d'informations sur les rubriques Amazon SNS, consultez [Configuration d'Amazon SNS](#) dans le Guide du développeur Amazon SNS.
 - Pour la destination du message entrant, entrez un nom de rubrique, tel que **SMSRegistrationFormTopic**.
7. Pour le Rôle de canal bidirectionnel, choisissez Utiliser les politiques de rubrique SNS.
8. Sélectionnez Enregistrer les modifications.

Utilisez la console Amazon Pinpoint SMS pour ajouter des mots clés à votre numéro de téléphone, que les clients vous envoient pour confirmer leurs abonnements (comme **Yes** ou **Confirm**).

Ajouter un mot-clé

1. Ouvrez la console Amazon Pinpoint SMS à l'adresse <https://console.aws.amazon.com/sms-voice/>.
2. Dans le volet de navigation, sous Configurations, choisissez Numéro de téléphone.
3. Sur la page Numéro de téléphone, choisissez le numéro de téléphone auquel ajouter un mot-clé.
4. Dans l'onglet Mots-clés, cliquez sur le bouton Ajouter un mot-clé.
5. Dans le volet Mot-clé personnalisé, ajoutez ce qui suit :
 - Mot-clé : nouveau mot-clé à ajouter (tel que **Yes** ou **Confirm**).
 - Message de réponse : message à renvoyer au destinataire.
 - Action de mot-clé : action à exécuter lorsque le mot-clé est reçu. Choisissez Réponse automatique.
6. Choisissez Ajouter un mot-clé.

Suivant : [Création de politiques et de rôles IAM](#)

Étape 2 : créer des politiques et des rôles IAM

L'étape suivante de la mise en œuvre de la solution d'enregistrement par SMS consiste à configurer une politique et un rôle dans AWS Identity and Access Management (IAM). Pour cette solution, vous devez créer une stratégie qui permet l'accès à certaines ressources qui sont associées à Amazon Pinpoint. Vous créez un rôle et lui attachez la stratégie. Plus loin dans ce didacticiel, vous allez créer une AWS Lambda fonction qui utilise ce rôle pour appeler certaines opérations dans l'API Amazon Pinpoint.

Créer une politique IAM

Cette section vous montre comment créer une politique IAM. Les utilisateurs et les rôles qui utilisent cette stratégie sont en mesure d'effectuer les actions suivantes :

- Utiliser la fonction de validation du numéro de téléphone
- Afficher, créer et mettre à jour les points de terminaison Amazon Pinpoint
- Envoyer des messages aux points de terminaison Amazon Pinpoint

Dans ce tutoriel, vous souhaitez donner à Lambda la possibilité d'effectuer ces tâches. Toutefois, pour plus de sécurité, cette stratégie utilise le principe d'attribution du moindre privilège. En d'autres termes, elle accorde uniquement les autorisations requises pour exécuter la solution, et rien de plus. Cette stratégie est limitée de la manière suivante :

- Vous pouvez uniquement l'utiliser pour appeler l'API Phone Number Validate dans une région spécifique.
- Vous pouvez uniquement utiliser cette stratégie pour afficher, créer ou mettre à jour les points de terminaison associés à un projet Amazon Pinpoint spécifique.
- Vous pouvez uniquement l'utiliser pour envoyer des messages aux points de terminaison qui sont associés à un projet Amazon Pinpoint spécifique.

Pour créer la politique

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le volet de navigation, sélectionnez Politiques, puis Créer une politique.
3. Dans l'onglet JSON, copiez le code suivant.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:CreateLogGroup"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": "mobiletargeting:SendMessages",
      "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "mobiletargeting:GetEndpoint",
        "mobiletargeting:UpdateEndpoint",
        "mobiletargeting:PutEvents"
    ],
    "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/
endpoints/*"
    },
    {
        "Effect": "Allow",
        "Action": "mobiletargeting:PhoneNumberValidate",
        "Resource": "arn:aws:mobiletargeting:region:accountId:phone/number/
validate"
    }
]
}
```


Dans l'exemple précédent, procédez comme suit :

- Remplacez *la région* par la AWS région dans laquelle vous utilisez Amazon Pinpoint, telle que `us-east-1` ou `eu-central-1`

 Tip

Pour obtenir la liste complète des AWS régions dans lesquelles Amazon Pinpoint est disponible, consultez la section [AWS Régions et points de terminaison](#) dans le. Références générales AWS

- Remplacez *AccountID* par l'identifiant unique de votre compte. AWS
- Remplacez *ProjectID* par l'identifiant unique du projet que vous avez créé [dans Créer un projet Amazon Pinpoint de ce didacticiel](#).

 Note

Les logs actions permettent à Lambda de consigner sa sortie dans CloudWatch Logs.

4. Choisissez Suivant.
5. Dans Nom de la stratégie, entrez un nom pour la stratégie, tel que **RegistrationFormPolicy**. Choisissez Créer une politique.

Créer un rôle IAM

Pour créer le rôle

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation de la console IAM, choisissez Rôles, puis Créer un rôle.
3. Sous Type d'entité fiable, choisissez AWS service, puis pour Service ou cas d'utilisateur, choisissez Lambda dans la liste déroulante.
4. Choisissez Suivant.
5. Sous Politiques d'autorisation, choisissez ou recherchez la politique que vous avez créée dans la section précédente, puis choisissez Suivant.
6. Sous Détails du rôle, dans Nom du rôle, entrez un nom pour le rôle, tel que **SMSRegistrationForm**. Sélectionnez Créer un rôle.

Suivant : [Création de fonctions Lambda](#)

Étape 3 : Créer des fonctions Lambda

Cette solution utilise deux fonctions Lambda. Cette section vous montre comment créer et configurer ces fonctions. Ultérieurement, vous allez configurer API Gateway et Amazon Pinpoint pour exécuter ces fonctions lorsque certains événements se produisent. Ces deux fonctions créent et mettent à jour les points de terminaison dans le projet Amazon Pinpoint que vous spécifiez. La première fonction utilise également la fonction de validation de numéro de téléphone.

Créer la fonction qui valide les informations clients et crée les points de terminaison

La première fonction prend les informations de votre formulaire d'inscription, qu'elle reçoit d'Amazon API Gateway. Elle utilise ces informations pour obtenir des informations sur le numéro de téléphone du client en utilisant la fonction de [validation du numéro de téléphone](#) d'Amazon Pinpoint. La fonction utilise alors les données validées pour créer un point de terminaison dans le projet Amazon Pinpoint que vous spécifiez. Par défaut, le point de terminaison que la fonction crée est exclu de vos futures communications, mais cet état peut être modifié par la deuxième fonction. Enfin, cette fonction envoie au client un message leur demandant de vérifier qu'ils souhaitent recevoir des communications SMS de votre part.

Pour créer la fonction Lambda

1. Ouvrez la AWS Lambda console à l'adresse <https://console.aws.amazon.com/lambda/>.

2. Choisissez Créer une fonction.
3. Sous Créer une fonction, choisissez Utiliser un plan.
4. Dans le champ de recherche, entrez **hello**, puis appuyez sur Entrée. Dans la liste des résultats, choisissez la fonction Node.js hello-world, comme illustré dans l'image suivante.

Create function Info

Choose one of the following options to create your function.

- Author from scratch
Start with a simple Hello World example.
- Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
- Container image
Select a container image to deploy for your function.

Basic information Info

Blueprint name
Hello world function nodejs18.x ▼
A starter AWS Lambda function.

Function name
Enter a name that describes the purpose of your function.
RegistrationForm
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime
nodejs18.x

Architecture
x86_64

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).
 Create a new role with basic Lambda permissions
 Use an existing role
 Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
SMSRegistrationForm ↻
[View the SMSRegistrationForm role](#) on the IAM console.

5. Sous Basic information (Informations de base), procédez comme suit :
 - Dans Name (Nom), saisissez un nom pour la fonction, tel que **RegistrationForm**.
 - Pour Rôle, sélectionnez Choisir un rôle existant.
 - Pour Rôle existant, choisissez le RegistrationForm rôle SMS que vous avez créé dans [Créer un rôle IAM](#).

Lorsque vous avez terminé, choisissez Create function (Créer une fonction).

6. Pour Code source, supprimez l'exemple de fonction dans l'éditeur de code, puis collez le code suivant :

```
import { PinpointClient, PhoneNumberValidateCommand, UpdateEndpointCommand,
  SendMessagesCommand } from "@aws-sdk/client-pinpoint"; // ES Modules import
const pinClient = new PinpointClient({region: process.env.region});

// Make sure the SMS channel is enabled for the projectId that you specify.
// See: https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-setup.html
var projectId = process.env.projectId;
```

```
// You need a dedicated long code in order to use two-way SMS.
// See: https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-voice-
manage.html#channels-voice-manage-request-phone-numbers
var originationNumber = process.env.originationNumber;

// This message is spread across multiple lines for improved readability.
var message = "ExampleCorp: Reply YES to confirm your subscription. 2 msgs per "
  + "month. No purchase req'd. Msg&data rates may apply. Terms: "
  + "example.com/terms-sms";

var messageType = "TRANSACTIONAL";

export const handler = async (event, context) => {
  console.log('Received event:', event);
  await validateNumber(event);
};

async function validateNumber (event) {
  var destinationNumber = event.destinationNumber;
  if (destinationNumber.length == 10) {
    destinationNumber = "+1" + destinationNumber;
  }
  var params = {
    NumberValidateRequest: {
      IsoCountryCode: 'US',
      PhoneNumber: destinationNumber
    }
  };
  try{
    const PhoneNumberValidatorresponse = await pinClient.send( new
    PhoneNumberValidateCommand(params));
    console.log(PhoneNumberValidatorresponse);
    if (PhoneNumberValidatorresponse['NumberValidateResponse']['PhoneTypeCode'] ==
    0) {
      await createEndpoint(PhoneNumberValidatorresponse, event.firstName,
      event.lastName, event.source);

    } else {
      console.log("Received a phone number that isn't capable of receiving "
        + "SMS messages. No endpoint created.");
    }
  }catch(err){
    console.log(err);
  }
}
```

```
    }  
  }  
  
  async function createEndpoint(data, firstName, lastName, source) {  
    var destinationNumber = data['NumberValidateResponse']  
    ['CleansedPhoneNumberE164'];  
    var endpointId = data['NumberValidateResponse']  
    ['CleansedPhoneNumberE164'].substring(1);  
  
    var params = {  
      ApplicationId: projectId,  
      // The Endpoint ID is equal to the cleansed phone number minus the leading  
      // plus sign. This makes it easier to easily update the endpoint later.  
      EndpointId: endpointId,  
      EndpointRequest: {  
        ChannelType: 'SMS',  
        Address: destinationNumber,  
        // OptOut is set to ALL (that is, endpoint is opted out of all messages)  
        // because the recipient hasn't confirmed their subscription at this  
        // point. When they confirm, a different Lambda function changes this  
        // value to NONE (not opted out).  
        OptOut: 'ALL',  
        Location: {  
          PostalCode: data['NumberValidateResponse']['ZipCode'],  
          City: data['NumberValidateResponse']['City'],  
          Country: data['NumberValidateResponse']['CountryCodeIso2'],  
        },  
        Demographic: {  
          Timezone: data['NumberValidateResponse']['Timezone']  
        },  
        Attributes: {  
          Source: [  
            source  
          ]  
        },  
        User: {  
          UserAttributes: {  
            FirstName: [  
              firstName  
            ],  
            LastName: [  
              lastName  
            ]  
          }  
        }  
      }  
    }  
  }  
}
```

```
    }
  }
};
try{
  const UpdateEndpointresponse = await pinClient.send(new
UpdateEndpointCommand(params));
  console.log(UpdateEndpointresponse);
  await sendConfirmation(destinationNumber);
}catch(err){
  console.log(err);
}
}

async function sendConfirmation(destinationNumber) {
  var params = {
    ApplicationId: projectId,
    MessageRequest: {
      Addresses: {
        [destinationNumber]: {
          ChannelType: 'SMS'
        }
      },
      MessageConfiguration: {
        SMSMessage: {
          Body: message,
          MessageType: messageType,
          OriginationNumber: originationNumber
        }
      }
    }
  };
  try{
    const SendMessagesCommandresponse = await pinClient.send(new
SendMessagesCommand(params));
    console.log("Message sent! "
      + SendMessagesCommandresponse['MessageResponse']['Result']
[destinationNumber]['StatusMessage']);
  }catch(err){
    console.log(err);
  }
}
```

7. Dans l'onglet Configuration des variables d'environnement, choisissez Modifier puis Ajouter une variable d'environnement, procédez comme suit :

- Dans la première ligne, créez une variable avec la clé **originationNumber**. Ensuite, définissez la valeur sur le numéro de téléphone du code long dédié que vous avez reçu à [l'étape 1.2](#).

Note

N'oubliez pas d'inclure le signe plus (+) et le code du pays pour le numéro de téléphone. N'incluez pas d'autres caractères spéciaux, comme les tirets (-), les points (.) ou les parenthèses.

- Dans la deuxième ligne, créez une variable avec la clé **projectId**. Ensuite, définissez la valeur sur l'ID unique du projet que vous avez créé à [l'étape 1.1](#).
- Dans la troisième ligne, créez une variable avec la clé **region**. Ensuite, définissez la valeur sur la région dans laquelle vous utilisez Amazon Pinpoint, comme **us-east-1** ou **us-west-2**.

Lorsque vous avez terminé, la section Environment Variables (Variables d'environnement) doit ressembler à l'exemple illustré dans l'image suivante.

Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more.](#)

originationNumber	+12065550199	Remove
projectId	33d643d9bexample9a5e726f6c4	Remove
region	us-east-1	Remove
Key	Value	Remove

► Encryption configuration

8. Dans le haut de la page, choisissez Enregistrer.

Tester la fonction

Après avoir créé la fonction, vous devez la tester pour vous assurer qu'elle soit correctement configurée. De plus, assurez-vous que le rôle IAM que vous avez créé possède les autorisations appropriées.

Pour tester la fonction

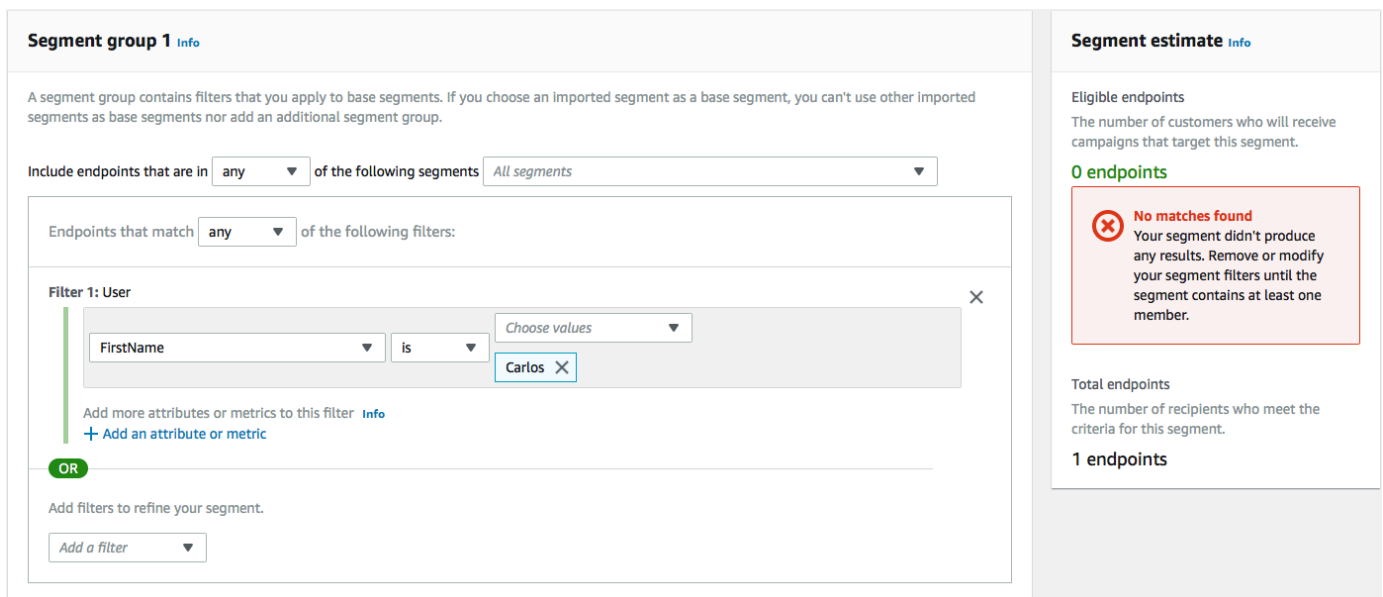
1. Choisissez l'onglet Test.
2. Choisissez Créer un nouvel événement, procédez comme suit :
 - Pour Event name (Nom d'événement), saisissez un nom pour l'événement de test, tel que **MyPhoneNumber**.
 - Effacez l'exemple de code dans l'éditeur de code. Collez le code suivant :

```
{
  "destinationNumber": "+12065550142",
  "firstName": "Carlos",
  "lastName": "Salazar",
  "source": "Registration form test"
}
```

- Dans l'exemple de code précédent, remplacez les valeurs des attributs `destinationNumber`, `firstName` et `lastName` par les valeurs que vous souhaitez utiliser pour les tests, telles que vos coordonnées personnelles. Lorsque vous testez cette fonction, elle envoie un message SMS au numéro de téléphone que vous spécifiez dans l'attribut `destinationNumber`. Assurez-vous que le numéro de téléphone que vous spécifiez est en mesure de recevoir des messages SMS.
 - Sélectionnez Create (Créer).
3. Sélectionnez Test.
 4. Sous Execution result : succeeded (Résultat de l'exécution : succès), choisissez Details (Détails). Dans la section Log output (Sortie de journal), vérifiez le résultat de la fonction. Assurez-vous que la fonction s'est exécutée sans erreurs.
- Vérifiez le périphérique associé au `destinationNumber` que vous avez spécifié pour vous assurer qu'il a reçu le message de test.
5. Ouvrez la console Amazon Pinpoint à l'adresse <https://console.aws.amazon.com/pinpoint/>.

6. Sur la page Tous les projets, choisissez le projet que vous avez créé dans [Créer un projet Amazon Pinpoint](#).
7. Dans le volet de navigation, sélectionnez Segments. Sur la page Segments, choisissez Create a segment (Créer un segment).
8. Dans le groupe Segment group 1 (1er groupe de segments), sous Add filters to refine your segment (Ajouter des filtres pour affiner votre segment), choisissez Filter by user (Filtrer par utilisateur).
9. Pour Choisir un attribut utilisateur, sélectionnez FirstName. Ensuite, pour Choose values (Choisir des valeurs), choisissez le premier nom que vous avez spécifié dans l'événement de test.

La section Segment estimate (Estimation du segment) doit montrer qu'il n'existe aucun point de terminaison éligible et un point de terminaison global, comme illustré dans l'image suivante. Ce résultat est prévu. Lorsque la fonction crée un nouveau point de terminaison, le point de terminaison est désinscrit. Les segments dans Amazon Pinpoint excluent automatiquement les points de terminaison désabonnés.



The screenshot displays the Amazon Pinpoint console interface. On the left, the 'Segment group 1' configuration is shown, including a filter for 'User' with the attribute 'FirstName' and the value 'Carlos'. On the right, the 'Segment estimate' section shows '0 endpoints' and a red warning box stating 'No matches found. Your segment didn't produce any results. Remove or modify your segment filters until the segment contains at least one member.'

Créer la fonction qui inscrit les clients à vos communications

La deuxième fonction est uniquement exécutée lorsqu'un client répond au message envoyé par la première fonction. Si la réponse du client inclut le mot clé que vous avez spécifié dans [Activer les SMS bidirectionnels](#), la fonction met à jour l'enregistrement de son terminal pour l'autoriser à participer aux futures communications. Amazon Pinpoint répond également automatiquement avec le message que vous avez spécifié dans [Activer les SMS bidirectionnels](#).

Si le client ne répond pas ou répond autrement qu'avec le mot-clé défini, rien ne se passe. Le point de terminaison du client reste dans Amazon Pinpoint, mais il ne peut pas être ciblé par segments.

Pour créer la fonction Lambda

1. Ouvrez la AWS Lambda console à l'[adresse https://console.aws.amazon.com/lambda/](https://console.aws.amazon.com/lambda/).
2. Choisissez Créer une fonction.
3. Sous Create a function (Créer une fonction), choisissez Blueprints (Plans).
4. Dans le champ de recherche, entrez **hello**, puis appuyez sur Entrée. Dans la liste des résultats, choisissez la fonction Node.js hello-world, comme illustré dans l'image suivante. Choisissez Configurer.
5. Sous Basic information (Informations de base), procédez comme suit :
 - Dans Name (Nom), saisissez un nom pour la fonction, tel que **RegistrationForm_OptIn**.
 - Pour Rôle, sélectionnez Choisir un rôle existant.
 - Pour Rôle existant, choisissez le RegistrationForm rôle SMS que vous avez créé dans [Créer un rôle IAM](#).

Lorsque vous avez terminé, choisissez Create function (Créer une fonction).

6. Supprimez l'exemple de fonction dans l'éditeur de code, puis collez le code suivant :

```
import { PinpointClient, UpdateEndpointCommand } from "@aws-sdk/client-pinpoint"; // ES Modules import

// Create a new Pinpoint client instance with the region specified in the environment variables
const pinClient = new PinpointClient({ region: process.env.region });

// Get the Pinpoint project ID and the confirm keyword from environment variables
const projectId = process.env.projectId;
const confirmKeyword = process.env.confirmKeyword.toLowerCase();

// This is the main handler function that is invoked when the Lambda function is triggered
export const handler = async (event, context) => {
  console.log('Received event:', event);

  try {
```

```
    // Extract the timestamp, message, and origination number from the SNS
    event
    const timestamp = event.Records[0].Sns.Timestamp;
    const message = JSON.parse(event.Records[0].Sns.Message);
    const originationNumber = message.originationNumber;
    const response = message.messageBody.toLowerCase();

    // Check if the response message contains the confirm keyword
    if (response.includes(confirmKeyword)) {
        // If the confirm keyword is found, update the endpoint's opt-in status
        await updateEndpointOptIn(originationNumber, timestamp);
    }
} catch (error) {
    console.error('An error occurred:', error);
    throw error; // Rethrow the error to handle it upstream
}
};


// This function updates the opt-in status of a Pinpoint endpoint
async function updateEndpointOptIn(originationNumber, timestamp) {
    // Extract the endpoint ID from the origination number
    const endpointId = originationNumber.substring(1);

    // Prepare the parameters for the UpdateEndpointCommand
    const params = {
        ApplicationId: projectId,
        EndpointId: endpointId,
        EndpointRequest: {
            Address: originationNumber,
            ChannelType: 'SMS',
            OptOut: 'NONE',
            Attributes: {
                OptInTimestamp: [timestamp]
            },
        },
    };

    try {
        // Send the UpdateEndpointCommand to update the endpoint's opt-in status
        const updateEndpointResponse = await pinClient.send(new
UpdateEndpointCommand(params));
        console.log(updateEndpointResponse);
        console.log(`Successfully changed the opt status of endpoint ID
${endpointId}`);
    }
}
```

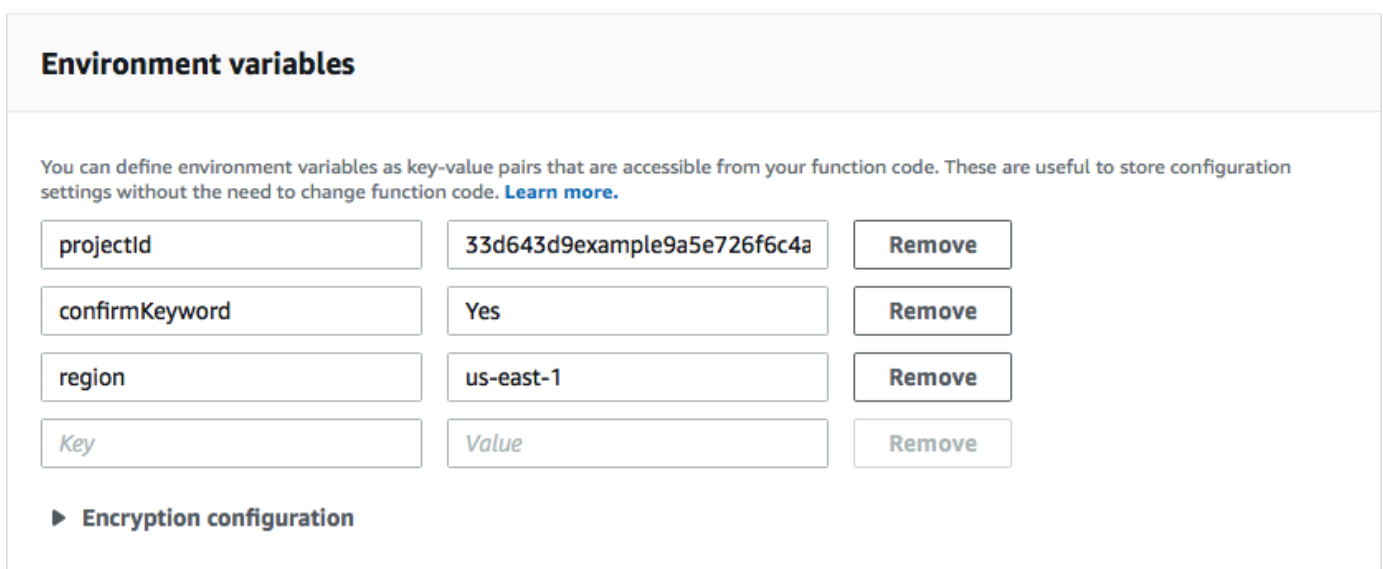
```
    } catch (error) {  
        console.error('An error occurred while updating endpoint:', error);  
        throw error; // Rethrow the error to handle it upstream  
    }  
}
```

7. Sous Environment variables (Variables d'environnement), effectuez les opérations suivantes :
- Dans la première ligne, créez une variable avec la clé **projectId**. Ensuite, définissez la valeur sur l'ID unique du projet que vous avez créé dans [Create an Amazon Pinpoint project](#).
 - Dans la deuxième ligne, créez une variable avec la clé **region**. Ensuite, définissez la valeur sur la région dans laquelle vous utilisez Amazon Pinpoint, comme **us-east-1** ou **us-west-2**.
 - Dans la troisième ligne, créez une variable avec la clé **confirmKeyword**. Ensuite, définissez la valeur sur le mot clé de confirmation que vous avez créé dans [Activer les SMS bidirectionnels](#).

 Note

Le mot-clé n'est pas sensible à la casse. Cette fonction convertit le message entrant en minuscules.

Lorsque vous avez terminé, la section Environment Variables (Variables d'environnement) doit ressembler à l'exemple illustré dans l'image suivante.



Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more.](#)

projectId	33d643d9example9a5e726f6c4a	Remove
confirmKeyword	Yes	Remove
region	us-east-1	Remove
Key	Value	Remove

► Encryption configuration

8. Dans le haut de la page, choisissez Enregistrer.

Tester la fonction

Après avoir créé la fonction, vous devez la tester pour vous assurer qu'elle soit correctement configurée. De plus, assurez-vous que le rôle IAM que vous avez créé possède les autorisations appropriées.

Pour tester la fonction

1. Sélectionnez Tester).
2. Sur la fenêtre Configure test event (Configurer un événement de test), procédez de la façon suivante :
 - a. Choisissez Create new test event.
 - b. Pour Event name (Nom d'événement), saisissez un nom pour l'événement de test, tel que **MyResponse**.
 - c. Effacez l'exemple de code dans l'éditeur de code. Collez le code suivant :

```
{
  "Records": [
    {
      "Sns": {
        "Message": "{\"originationNumber\": \"+12065550142\", \"messageBody\": \"Yes\"}\",
        "Timestamp": "2019-02-20T17:47:44.147Z"
      }
    }
  ]
}
```

Dans l'exemple de code précédent, remplacez les valeurs de l'attribut `originationNumber` par le numéro de téléphone que vous avez utilisé lorsque vous avez testé la fonction Lambda précédente. Remplacez la valeur de `messageBody` par le mot clé SMS bidirectionnel que vous avez spécifié dans [Activer les SMS bidirectionnels](#). Le cas échéant, vous pouvez remplacer la valeur `Timestamp` par la date et l'heure actuelles.

- d. Choisissez Créer.
3. Choisissez Test à nouveau.

4. Sous Execution result : succeeded (Résultat de l'exécution : succès), choisissez Details (Détails). Dans la section Log output (Sortie de journal), vérifiez le résultat de la fonction. Assurez-vous que la fonction s'est exécutée sans erreurs.
5. Ouvrez la console Amazon Pinpoint à l'adresse <https://console.aws.amazon.com/pinpoint/>.
6. Sur la page Tous les projets, choisissez le projet que vous avez créé dans [Créer un projet Amazon Pinpoint](#).
7. Dans le volet de navigation, sélectionnez Segments. Sur la page Segments, choisissez Create a segment (Créer un segment).
8. Dans le groupe Segment group 1 (1er groupe de segments), sous Add filters to refine your segment (Ajouter des filtres pour affiner votre segment), choisissez Filter by user (Filtrer par utilisateur).
9. Pour Choisir un attribut utilisateur, sélectionnez FirstName. Ensuite, pour Choose values (Choisir des valeurs), choisissez le premier nom que vous avez spécifié dans l'événement de test.

La section Segment estimate (Estimation du segment) doit montrer qu'il existe un point de terminaison éligible et un point de terminaison global.

Suivant : [Configuration d'Amazon API Gateway](#)

Étape 4 : Configuration d'Amazon API Gateway

Dans cette section, vous allez créer une API avec Amazon API Gateway. Le formulaire d'enregistrement que vous déployez dans cette solution appelle cette API. API Gateway transmet ensuite les informations capturées sur le formulaire d'enregistrement à la fonction Lambda que vous avez créée dans Create [Lambda functions](#).

Créer l'API

Tout d'abord, vous devez créer une nouvelle API dans API Gateway. Les procédures suivantes vous montrent comment créer une nouvelle API REST.

Pour créer une nouvelle API

1. Ouvrez la console API Gateway à l'adresse <https://console.aws.amazon.com/apigateway>.
2. Sélectionnez Create API (Créer une API). Effectuez les sélections suivantes :
 - Sous Choose the protocol (Choisir le protocole), choisissez REST.

- Sous Create new API (Créer une nouvelle API), choisissez New API (Nouvelle API).
- Sous Settings (Paramètres), pour Name (Nom), entrez un nom, par exemple **RegistrationForm**. Pour Description, vous pouvez saisir un texte qui décrit l'objectif de l'API. Pour Endpoint Type (Type de point de terminaison), choisissez Regional (Régional). Ensuite, choisissez Create API (Créer une API).

Un exemple de ces paramètres est illustré dans l'image suivante.

Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

REST WebSocket

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

New API Clone from existing API Import from Swagger or Open API 3 Example API

Settings

Choose a friendly name and description for your API.

API name*	<input type="text" value="RegistrationForm"/>
Description	<input type="text" value="Collects input from a registration form. which is passed on to a"/>
Endpoint Type	<input type="text" value="Regional"/> ⓘ

* Required

Create API

Créer une ressource

Maintenant que vous avez créé une API, vous pouvez commencer à y ajouter des ressources. Une fois que vous ajoutez une méthode POST à la ressource et demandez à API Gateway de transmettre les données que vous recevez de cette méthode à votre fonction Lambda.

1. Dans le menu Actions, choisissez Create Resource (Créer une ressource). Dans le volet New Child Resource (Nouvelle ressource enfant), pour Resource Name (Nom de la ressource), entrez **register**, comme illustré dans l'image suivante. Choisissez Create Resource.

New Child Resource

Use this page to create a new child resource for your resource. 

Configure as [proxy resource](#)

Resource Name*

Resource Path*

You can add path parameters using brackets. For example, the resource path **{username}** represents a path parameter called 'username'. Configuring **/{proxy+}** as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to /foo. To handle requests to /, add a new ANY method on the / resource.

Enable API Gateway CORS

* Required

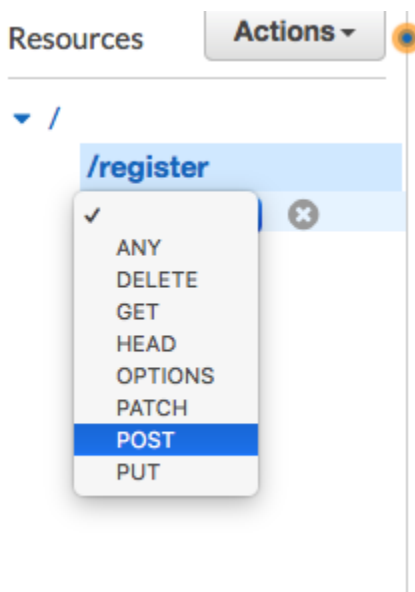
Cancel

Create Resource

2. Dans le menu Actions, sélectionnez Create method (Créer une méthode). Dans le menu qui s'affiche, choisissez POST, comme illustré dans l'image suivante. Ensuite, choisissez la coche



).



3. Dans le volet /register - POST - Setup, effectuez les sélections suivantes :

- Pour Integration type (Type d'intégration), choisissez Lambda Function (Fonction Lambda).

- Sélectionnez Utiliser l'intégration de proxy Lambda.
- Pour Région Lambda, choisissez la région dans laquelle vous avez créé la fonction Lambda.
- Pour Lambda Function, choisissez la RegisterEndpoint fonction que vous avez créée dans Create [Lambda functions](#).

Un exemple de ces paramètres est illustré dans l'image suivante.

/register - POST - Setup



Choose the integration point for your new method.

Integration type Lambda Function ⓘ

HTTP ⓘ

Mock ⓘ

AWS Service ⓘ

VPC Link ⓘ

Use Lambda Proxy integration ⓘ

Lambda Region

Lambda Function

ⓘ

Use Default Timeout ⓘ

Save

Choisissez Enregistrer. Dans la fenêtre qui s'affiche, choisissez OK pour donner à API Gateway l'autorisation d'exécuter votre fonction Lambda.

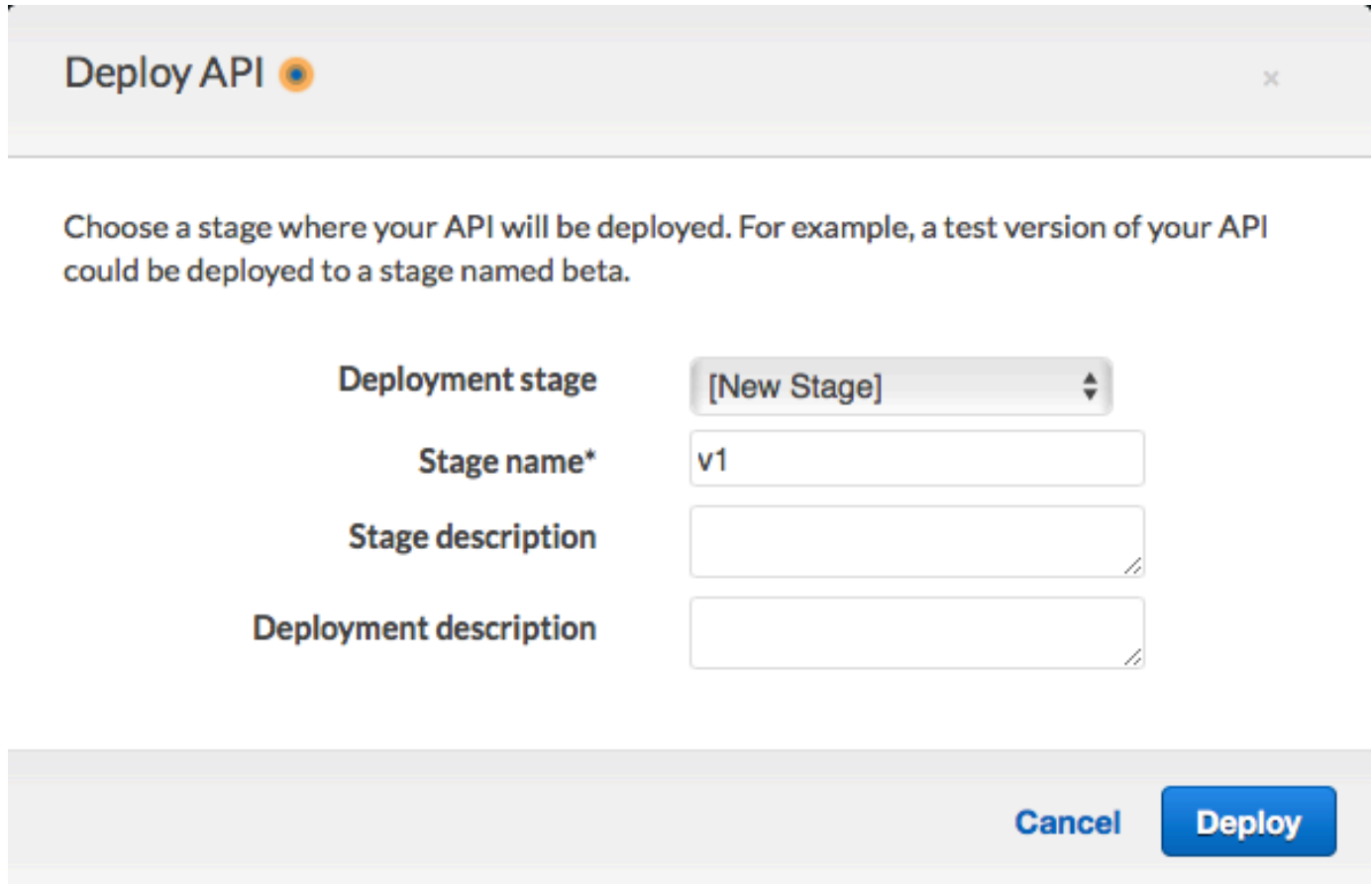
Déploiement de l'API

L'API est maintenant prête à être utilisée. À ce stade, vous devez la déployer pour créer un point de terminaison accessible publiquement.

1. Dans le menu Actions, choisissez Deploy API (Déployer l'API). Dans la fenêtre Deploy API (Déployer l'API), effectuez les sélections suivantes :

- Dans Deployment stage (Étape de déploiement), sélectionnez [New Stage] [Nouvelle étape].
- Sous Stage name (Nom de l'étape), entrez **v1**.
- Choisissez Deploy (Déployer).

Un exemple de ces sélections est illustré dans l'image suivante.



2. Dans le volet v1 Stage Editor, choisissez la ressource /register, puis choisissez la méthode POST. Copiez l'adresse qui est affichée à côté d'Invoke URL (URL d'appel)), comme illustré dans l'image suivante.

v1 - POST - /register

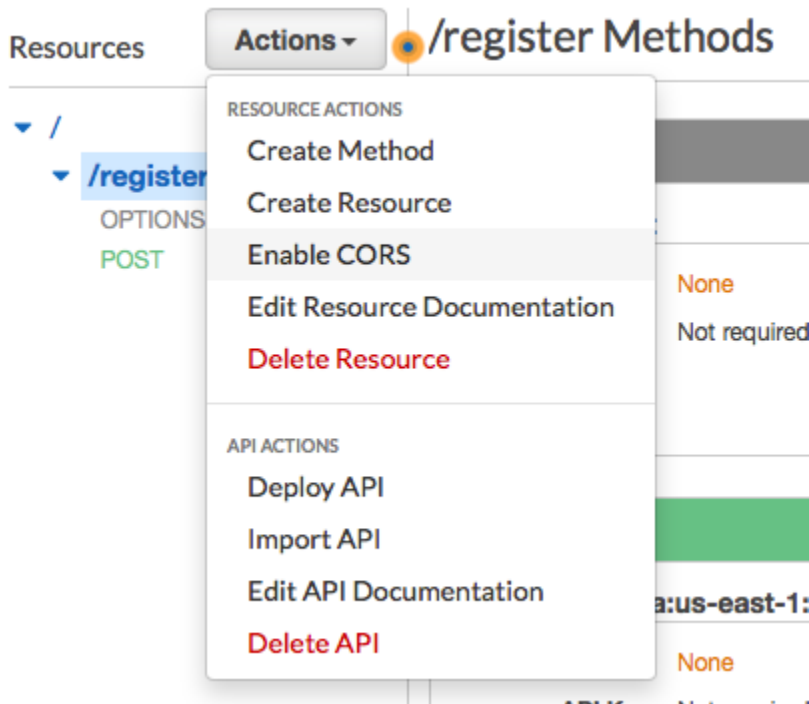
Invoke URL: <https://example.execute-api.us-east-1.amazonaws.com/v1/register>

Use this page to override the **v1 stage** settings for the POST to /register method.

- Settings Inherit from stage
 Override for this method

Save Changes

3. Dans le panneau de navigation, choisissez Ressources (Ressources). Dans la liste des ressources, choisissez la ressource /register. Enfin, dans le menu Actions, choisissez Enable CORS (Activer CORS), comme illustré dans l'image suivante.



4. Dans le volet Enable CORS (Activer CORS), choisissez Enable CORS and replace existing CORS headers (Activer CORS et remplacer les en-têtes CORS existants).

Suivant : [Créer et déployer le formulaire web](#)

Étape 5 : Créer et déployer le formulaire web

Tous les composants de cette solution qui utilisent AWS des services sont désormais en place. La dernière étape consiste à créer et à déployer le formulaire web qui capture des données du client.

Création du gestionnaire JavaScript de formulaires

Dans cette section, vous créez une JavaScript fonction qui analyse le contenu du formulaire Web que vous créez dans la section suivante. Après avoir analysé le contenu, cette fonction envoie les données à l'API que vous avez créée dans [Set up Amazon API Gateway](#).

Pour créer le gestionnaire de formulaire

1. Dans un éditeur de texte, créez un fichier.
2. Dans l'éditeur, collez le code suivant.

```
$(document).ready(function() {

    // Handle form submission.
    $("#submit").click(function(e) {

        var firstName = $("#firstName").val(),
            lastName = $("#lastName").val(),
            source = window.location.pathname,
            optTimestamp = undefined,
            utcSeconds = Date.now() / 1000,
            timestamp = new Date(0),
            phone = $("#areaCode").val()
                + $("#phone1").val()
                + $("#phone2").val();

        e.preventDefault();

        if (firstName == "") {
            $('#form-response').html('<div class="mt-3 alert alert-info"
            role="alert">Please enter your first name.</div>');
        } else if (lastName == "") {
            $('#form-response').html('<div class="mt-3 alert alert-info"
            role="alert">Please enter your last name.</div>');
        } else if (phone.match(/^[^0-9]/gi)) {
            $('#form-response').html('<div class="mt-3 alert alert-info"
            role="alert">Your phone number contains invalid characters. Please check the phone
            number that you supplied.</div>');
        } else if (phone.length < 10) {
            $('#form-response').html('<div class="mt-3 alert alert-info"
            role="alert">Please enter your phone number.</div>');
        } else if (phone.length > 10) {
            $('#form-response').html('<div class="mt-3 alert alert-info"
            role="alert">Your phone number contains too many digits. Please check the phone
            number that you supplied.</div>');
        } else {
            $('#submit').prop('disabled', true);
            $('#submit').html('<span class="spinner-border spinner-border-sm"
            role="status" aria-hidden="true"></span> Saving your preferences</button>');

            timestamp.setUTCSeconds(utcSeconds);

            var data = JSON.stringify({
                'destinationNumber': phone,
```

```
'firstName': firstName,
'lastName': lastName,
'source': source,
'optTimestamp': timestamp.toString()
});

$.ajax({
  type: 'POST',
  url: 'https://example.execute-api.us-east-1.amazonaws.com/v1/register',
  contentType: 'application/json',
  data: data,
  success: function(res) {
    $('#form-response').html('<div class="mt-3 alert alert-success"
role="alert"><p>Congratulations! You've successfully registered for SMS
Alerts from ExampleCorp.</p><p>We just sent you a message. Follow the instructions
in the message to confirm your subscription. We won't send any additional
messages until we receive your confirmation.</p><p>If you decide you don't
want to receive any additional messages from us, just reply to one of our messages
with the keyword STOP.</p></div>');
    $('#submit').prop('hidden', true);
    $('#unsubAll').prop('hidden', true);
    $('#submit').text('Preferences saved!');
  },
  error: function(jqxhr, status, exception) {
    $('#form-response').html('<div class="mt-3 alert alert-danger"
role="alert">An error occurred. Please try again later.</div>');
    $('#submit').text('Save preferences');
    $('#submit').prop('disabled', false);
  }
});
}
});
});
```

3. Dans l'exemple précédent, remplacez *https://example.execute-api.us-east-1.amazonaws.com/v1/register* par l'URL Invoke que vous avez obtenue dans [Deploy the API](#).
4. Enregistrez le fichier.

Création du fichier de formulaire

Dans cette section, vous allez créer un fichier HTML qui contient le formulaire que les clients utilisent pour s'inscrire à votre programme de SMS. Ce fichier utilise le gestionnaire de JavaScript formulaire que vous avez créé dans la section précédente pour transmettre les données du formulaire à votre fonction Lambda.

Important

Lorsqu'un utilisateur envoie ce formulaire, il déclenche une fonction Lambda qui appelle plusieurs opérations d'API Amazon Pinpoint. Les utilisateurs malveillants pourraient lancer sur votre formulaire une attaque susceptible de causer un grand nombre de demandes. Si vous prévoyez d'utiliser cette solution pour une utilisation en production, vous devez la sécuriser en utilisant un système tel que [Google reCAPTCHA](#).

Pour créer le formulaire

1. Dans un éditeur de texte, créez un fichier.
2. Dans l'éditeur, collez le code suivant.

```
<!doctype html>
<html lang="en">

<head>
  <!-- Meta tags required by Bootstrap -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/
css/bootstrap.min.css" integrity="sha384-gg0yR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/
popper.js/1.14.7/umd/popper.min.js" integrity="sha384-
U02eT0CpHqdSjQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
```

```

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/
bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/
nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>

<script type="text/javascript" src="SMSFormHandler.js"></script>
<title>SMS Registration Form</title>
</head>

<body>
<div class="container">
<div class="row justify-content-center mt-3">
<div class="col-md-6">
<h1>Register for SMS Alerts</h1>
<p>Enter your phone number below to sign up for PromotionName messages from
ExampleCorp.</p>
<p>We don't share your contact information with anyone else. For more
information, see our <a href="http://example.com/privacy">Privacy Policy</a>.</p>
<p>ExampleCorp alerts are only available to recipients in the United
States.</p>
</div>
</div>
<div class="row justify-content-center">
<div class="col-md-6">
<form>
<div class="form-group">
<label for="firstName" class="font-weight-bold">First name</label>
<input type="text" class="form-control" id="firstName"
placeholder="Your first name" required>
</div>
<div class="form-group">
<label for="lastName" class="font-weight-bold">Last name</label>
<input type="text" class="form-control" id="lastName" placeholder="Your
last name" required>
</div>
<label for="areaCode" class="font-weight-bold">Phone number</label>
<div class="input-group">
<span class="h3">(&nbsp;</span>
<input type="tel" class="form-control" id="areaCode" placeholder="Area
code" required>
<span class="h3">&nbsp;</span>)&nbsp;</div>
<input type="tel" class="form-control" id="phone1" placeholder="555"
required>

```

```
        <span class="h3">&nbsp;&nbsp;&nbsp;&-&nbsp;&nbsp;&nbsp;&</span>
        <input type="tel" class="form-control" id="phone2" placeholder="0199"
required>
    </div>
    <div id="form-response"></div>
    <button id="submit" type="submit" class="btn btn-primary btn-block
mt-3">Submit</button>
</form>
</div>
</div>
<div class="row mt-3">
    <div class="col-md-12 text-center">
        <small class="text-muted">Copyright © 2019, ExampleCorp or its
affiliates.</small>
    </div>
</div>
</div>
</div>
</body>

</html>
```

3. Dans l'exemple précédent, remplacez *SMS FormHandler .js* par le chemin complet du JavaScript fichier de gestionnaire de formulaire que vous avez créé dans la section précédente.
4. Enregistrez le fichier.

Téléchargez les fichiers du formulaire

Maintenant que vous avez créé le formulaire HTML et le gestionnaire de JavaScript formulaires, la dernière étape consiste à publier ces fichiers sur Internet. Cette section suppose que vous avez déjà un fournisseur d'hébergement web. Si vous n'avez pas encore de fournisseur d'hébergement, vous pouvez lancer un site Web en utilisant Amazon Route 53, Amazon Simple Storage Service (Amazon S3) et Amazon CloudFront. Pour de plus amples informations, veuillez consulter [Hébergement d'un site web statique](#).

Si vous utilisez un autre fournisseur d'hébergement web, consultez la documentation de votre fournisseur pour plus d'informations sur la publication des pages web.

Tester le formulaire

Une fois que vous avez publié le formulaire, vous devez envoyer certains événements de test pour vous assurer qu'il fonctionne comme prévu.

Pour tester le formulaire d'inscription

1. Dans un navigateur web, accédez à l'emplacement où vous avez téléchargé le formulaire d'inscription. Si vous avez utilisé l'exemple de code de [Create the JavaScript form handler](#), vous voyez un formulaire qui ressemble à l'exemple de l'image suivante.

Register for SMS Alerts

Enter your phone number below to sign up for PromotionName messages from ExampleCorp.

We don't share your contact information with anyone else. For more information, see our [Privacy Policy](#).

ExampleCorp alerts are only available to recipients in the United States.

First name

Last name

Phone number

() -

Copyright © 2019, ExampleCorp or its affiliates.

2. Saisissez vos coordonnées dans les champs Prénom, Nom et Numéro de téléphone.

Note

Lorsque vous soumettez le formulaire, Amazon Pinpoint tente d'envoyer un message au numéro de téléphone que vous avez spécifié. En raison de cette fonctionnalité, vous devez utiliser un numéro de téléphone réel pour tester la solution du début à la fin.

Si vous avez testé la fonction Lambda dans Create [Lambda functions](#), votre projet Amazon Pinpoint contient déjà au moins un point de terminaison. Lorsque vous testez ce formulaire, vous devez soit envoyer un numéro de téléphone différent sur le formulaire, soit supprimer le point de terminaison existant à l'aide de l'opération [DeleteEndpointAPI](#).

3. Vérifiez l'appareil qui est associé au numéro de téléphone que vous avez spécifié pour vous assurer qu'il a reçu le message.
4. Ouvrez la console Amazon Pinpoint à l'adresse <https://console.aws.amazon.com/pinpoint/>.
5. Sur la page Tous les projets, choisissez le projet que vous avez créé dans [Créer un projet Amazon Pinpoint](#).
6. Dans le volet de navigation, sélectionnez Segments. Sur la page Segments, choisissez Create a segment (Créer un segment).
7. Dans le groupe Segment group 1 (1er groupe de segments), sous Add filters to refine your segment (Ajouter des filtres pour affiner votre segment), choisissez Filter by user (Filtrer par utilisateur).
8. Pour Choisir un attribut utilisateur, sélectionnez FirstName. Ensuite, pour Choose values (Choisir des valeurs), choisissez le premier nom que vous avez spécifié lorsque vous avez envoyé le formulaire.

La section Segment estimate (Estimation du segment) doit indiquer qu'il n'y a pas de points de terminaison éligibles, et un point de terminaison (sous Total endpoints (Total des points de terminaison)), comme illustré dans l'exemple suivant. Ce résultat est prévu. Lorsque la fonction Lambda crée un nouveau point de terminaison, le point de terminaison est désinscrit par défaut.

The screenshot displays the Amazon Pinpoint console interface for configuring a segment and viewing its estimate. On the left, the 'Segment group 1' configuration is shown. It includes a description of segment groups and options to include endpoints from 'any' of the following segments ('All segments'). Under 'Endpoints that match any of the following filters:', a filter named 'Filter 1: User' is configured with the attribute 'FirstName' and the value 'Carlos'. Below this, there is an 'OR' section and a prompt to 'Add filters to refine your segment.' On the right, the 'Segment estimate' section shows 'Eligible endpoints' as '0 endpoints' with a red warning box stating 'No matches found'. Below this, 'Total endpoints' is listed as '1 endpoints'.

9. Sur l'appareil qui a reçu le message, répondez-y avec le mot clé SMS bidirectionnel que vous avez spécifié dans [Activer les SMS bidirectionnels](#). Amazon Pinpoint envoie immédiatement un message de réponse.
10. Dans la console Amazon Pinpoint, répétez les étapes 4 à 8. Cette fois, lorsque vous créez le segment, vous voyez un point de terminaison éligible, et un point de terminaison global. Ce résultat est prévu, car le point de terminaison est désormais inscrit.

Étapes suivantes

En suivant ce didacticiel, vous avez effectué les opérations suivantes :

- Création d'un projet Amazon Pinpoint, configuration du canal SMS et obtention d'un code long dédié.
- Création d'une politique IAM qui utilise le principal du moindre privilège pour accorder les droits d'accès et association de cette stratégie à un rôle.
- Création de deux fonctions Lambda qui utilisent les opérations `PhoneNumberValidate`, `UpdateEndpoint` et `SendMessages` dans l'API Amazon Pinpoint.
- Création d'une API REST avec API Gateway.
- Création et déploiement d'un formulaire web qui collecte les informations de contact des clients.
- Exécution de tests sur la solution pour vous assurer qu'elle fonctionne.

Cette section décrit quelques méthodes grâce auxquelles vous pouvez utiliser les informations clients collectées en utilisant cette solution. Elle inclut aussi des suggestions sur la façon dont vous pouvez personnaliser cette solution pour l'adapter à votre propre cas d'utilisation.

Création de segments clients

Toutes les informations clients que vous collectez grâce à ce formulaire sont stockées en tant que points de terminaison. Cette solution crée des points de terminaison qui contiennent plusieurs attributs que vous pouvez utiliser à des fins de segmentation.

Par exemple, cette solution capture un attribut de point de terminaison appelé `Source`. Cet attribut contient le chemin d'accès complet à l'emplacement où le formulaire était hébergé. Lorsque vous créez un segment, vous pouvez filtrer le segment par point de terminaison, puis affiner le filtre en choisissant un attribut `Source`.

La création de segments en fonction de l'attribut `Source` attribut peut être utile de plusieurs façons. Tout d'abord, elle vous permet de créer rapidement un segment de clients qui se sont inscrits pour recevoir des messages SMS de votre part. De plus, l'outil de segmentation dans Amazon Pinpoint exclut automatiquement les points de terminaison qui ne se sont pas inscrits pour recevoir des messages.

L'attribut `Source` est également utile si vous décidez d'héberger le formulaire d'inscription dans plusieurs emplacements différents. Par exemple, vos documents marketing peuvent faire référence à un formulaire qui est hébergé dans un emplacement, tandis que les clients qui rencontrent le formulaire tout en parcourant votre site web peuvent afficher une version hébergée ailleurs. Dans ce cas, les attributs `Source` des clients qui remplissent le formulaire après avoir consulté vos documents marketing sont différents de ceux qui remplissent le formulaire après l'avoir trouvé sur votre site web. Vous pouvez utiliser cette différence pour créer des segments distincts, puis envoyer des communications adaptées à chacun de ces groupes.

Envoi de messages de campagne personnalisés

Une fois que vous avez créé des segments, vous pouvez commencer à envoyer des campagnes à ces segments. Lorsque vous créez les messages de campagne, vous pouvez les personnaliser en spécifiant les attributs de point de terminaison que vous souhaitez inclure dans le message. Par exemple, le formulaire web utilisé dans cette solution nécessite que le client entre son nom et prénom. Ces valeurs sont stockées dans l'enregistrement utilisateur associé au point de terminaison.

Par exemple, si vous utilisez l'opération d'API `GetEndpoint` pour récupérer les informations sur un point de terminaison qui a été créé à l'aide de cette solution, vous voyez une section qui ressemble à l'exemple suivant :

```
...
"User": {
  "UserAttributes": {
    "FirstName": [
      "Carlos"
    ],
    "LastName": [
      "Salazar"
    ]
  }
}
...
```

Si vous souhaitez inclure les valeurs de ces attributs dans votre message de campagne, vous pouvez utiliser la notation sous forme de point pour faire référence à l'attribut. Ensuite, entourez la totalité de la référence d'accolades doubles. Par exemple, si vous souhaitez inclure le prénom de chaque destinataire dans un message de campagne, incluez la chaîne suivante dans le message : `{{User.UserAttributes.FirstName}}`. Quand Amazon Pinpoint envoie le message, la chaîne est remplacée par la valeur de l'attribut `FirstName`.

Utilisation d'un formulaire pour collecter des informations supplémentaires

Vous pouvez modifier cette solution pour collecter des informations supplémentaires sur le formulaire d'inscription. Par exemple, vous pouvez demander au client de fournir son adresse, puis utiliser cette donnée pour remplir les champs `Location.City`, `Location.Country`, `Location.Region` et `Location.PostalCode` de la ressource `Endpoint`. La collecte des informations d'adresse sur le formulaire d'inscription peut se traduire par le fait que le point de terminaison contient des informations plus précises. Pour effectuer cette modification, vous devez ajouter les champs appropriés au formulaire web. Vous pouvez également modifier le code JavaScript du formulaire pour transmettre les nouvelles valeurs. Enfin, vous devez modifier la fonction Lambda qui crée le point de terminaison pour traiter les nouvelles informations entrantes.

Vous pouvez également modifier le formulaire de façon à ce qu'il recueille les informations de contact d'autres canaux. Par exemple, vous pouvez utiliser le formulaire pour collecter l'adresse e-mail du client en plus de son numéro de téléphone. Pour effectuer cette modification, vous devez modifier le code HTML et JavaScript du formulaire web. Vous pouvez également modifier la fonction Lambda qui crée le point de terminaison de façon à créer deux points de terminaison distincts (un pour le point de terminaison de messagerie et un autre pour le point de terminaison SMS). Vous devez également modifier la fonction Lambda afin qu'elle génère une valeur unique pour l'attribut `User.UserId` et associe cette valeur aux deux points de terminaison.

Enregistrement d'attributs supplémentaires à des fins d'audit

Cette solution enregistre deux attributs précieux lors de la création et de la mise à jour des points de terminaison. Tout d'abord, lorsque la première fonction Lambda crée initialement le point de terminaison, il enregistre l'URL du formulaire lui-même dans l'attribut `Attributes.Source`. Si le client répond au message, la deuxième fonction Lambda crée un attribut `Attributes.OptInTimestamp`. Cet attribut contient la date et l'heure exactes auxquelles le client a accepté de recevoir des messages de votre part.

Ces deux champs peuvent être utiles si vous êtes invité par un opérateur mobile ou un organisme de réglementation à fournir la preuve du consentement d'un client. Vous pouvez récupérer ces informations à tout moment à l'aide de l'opération d'API [GetEndpoint](#).

Vous pouvez également modifier les fonctions Lambda pour enregistrer les données supplémentaires susceptibles d'être utiles à des fins d'audit, telles que l'adresse IP à partir de laquelle la demande d'inscription a été envoyée.

Intégration de votre application à Amazon Pinpoint

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

Une fois que l'intégration a eu lieu, lorsque les utilisateurs lancent votre application, ils se connectent au service Amazon Pinpoint pour ajouter ou mettre à jour les 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.

En outre, votre application fournit les données d'utilisation, ou les données d'événement. 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 “Connecter votre application frontale avec AWS Amplify”](#).

En dehors de votre client, vous pouvez utiliser les [kits SDK AWS pris en charge](#) ou les [API Amazon Pinpoint](#) pour importer les points de terminaison, exporter les données des événements, définir les segments de clientèle, créer et exécuter les campagnes, et bien plus encore.

Rubriques

- [Utilisation d'Amazon Pinpoint avec un SDK AWS](#)
- [Connexion de votre application frontale à Amazon Pinpoint avec AWS Amplify](#)
- [Enregistrement des points de terminaison dans votre application](#)
- [Rapport des événements dans votre application](#)
- [Gestion des notifications push](#)

Utilisation d'Amazon Pinpoint avec un SDK AWS

AWS des kits de développement logiciel (SDK) sont disponibles pour de nombreux langages de programmation populaires. 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 for C++	AWS SDK for C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK for Go	AWS SDK for Go exemples de code
AWS SDK for Java	AWS SDK for Java exemples de code
AWS SDK for JavaScript	AWS SDK for JavaScript exemples de code
Kit AWS SDK pour Kotlin	Kit AWS SDK pour Kotlin exemples de code
AWS SDK for .NET	AWS SDK for .NET exemples de code
AWS SDK for PHP	AWS SDK for PHP exemples de code
AWS Tools for PowerShell	Outils pour des exemples PowerShell de code
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemples de code
AWS SDK for Ruby	AWS SDK for Ruby exemples de code
Kit AWS SDK pour Rust	Kit AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
Kit AWS SDK pour Swift	Kit AWS SDK pour Swift exemples de code

Pour des exemples spécifiques à Amazon Pinpoint, consultez [Exemples de code pour Amazon Pinpoint utilisant des kits SDK AWS](#)

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.

Connexion de votre application frontale à Amazon Pinpoint avec AWS 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, consultez [Démarrer](#) 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 analytiques, de messagerie intégrée et de notifications push, consultez [AWS Amplify](#).

Étape suivante

Vous avez intégré AWS Amplify à votre application. Ensuite, mettez à jour votre code pour enregistrer les périphériques de vos utilisateurs en tant que points de terminaison. Consultez [Enregistrement des points de terminaison dans votre application](#).

Enregistrement des points de terminaison 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 des ID utilisateur aux points de terminaison, vous pouvez afficher des graphiques sur l'activité des utilisateurs dans la console, notamment les graphiques Daily active users et Monthly active users.

Avant de commencer

Si vous ne l'avez pas déjà fait, intégrez le kit SDK AWS Mobile pour Android ou iOS, ou intégrez la bibliothèque AWS Amplify JavaScript à votre application. Consultez [Connexion de votre application frontale à Amazon Pinpoint avec AWS Amplify](#).

Enregistrement des points de terminaison avec les kits SDK AWS Mobile pour Android ou iOS

Vous pouvez utiliser les kits SDK AWS Mobile pour Android ou iOS pour enregistrer et personnaliser les points de terminaison. 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.

Enregistrement des points de terminaison avec la bibliothèque AWS Amplify JavaScript

Vous pouvez utiliser la bibliothèque AWS Amplify JavaScript pour enregistrer et mettre à jour des points de terminaison dans vos applications. Pour plus d'informations et pour afficher des exemples de code, consultez [Mise à jour des points de terminaison](#) dans la documentation d'AWS Amplify JavaScript.

Étapes suivantes

Vous avez mis à jour votre application pour enregistrer les points de terminaison. Maintenant, lorsque les utilisateurs lancent votre application, les informations sur les appareils et les attributs personnalisés sont transmises à Amazon Pinpoint. Vous pouvez utiliser ces informations pour définir les segments de public ciblé. Dans la console, vous pouvez voir les métriques sur les points de terminaison et, le cas échéant, les utilisateurs auxquels sont attribués les ID d'utilisateur.

Ensuite, procédez comme indiqué dans la section [Rapport des événements dans votre application](#) pour mettre à jour votre application de manière à créer un rapport sur les données d'utilisation.

Rapport des événements dans votre application

Dans votre application mobile ou web, vous pouvez utiliser les kits SDK AWS Mobile ou [l'API d'événements Amazon Pinpoint](#) pour communiquer 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 [Diffusion d'événements Amazon Pinpoint sur Kinesis](#).

En utilisant les kits SDK AWS Mobile et les bibliothèques AWS Amplify JavaScript, vous pouvez appeler l'API Amazon Pinpoint pour générer des rapports sur 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.

Événements 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 avec AWS Amplify. Consultez [Connexion de votre application frontale à Amazon Pinpoint avec AWS Amplify](#).
- Mettez à jour votre application pour enregistrer les points de terminaison. Consultez [Enregistrement des points de terminaison dans votre application](#).

Rapports sur les événements avec les kits SDK AWS Mobile pour Android ou iOS

Vous pouvez activer une application mobile pour signaler des événements à Amazon Pinpoint en utilisant les kits SDK AWS Mobile pour iOS et Android.

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

- [Analyse](#) dans la documentation du kit SDK iOS
- [Analyse](#) dans la documentation du kit SDK Android

Rapport sur les événements avec la bibliothèque AWS Amplify JavaScript

Vous pouvez activer des applications JavaScript ou React Native pour rapporter les événements d'utilisation de l'application sur Amazon Pinpoint en utilisant la bibliothèque AWS Amplify JavaScript. Pour plus d'informations sur la mise à jour de votre application pour envoyer les événements à Amazon Pinpoint, consultez [Analyse](#) dans la documentation d'AWS Amplify JavaScript.

Rapport des événements à l'aide de l'API Amazon Pinpoint

Vous pouvez utiliser l'API Amazon Pinpoint ou un kit SDK AWS pour envoyer des événements à Amazon Pinpoint en nombre. Pour plus d'informations, consultez [Événements](#) dans la Référence de l'API Amazon Pinpoint.

Étape suivante

Vous avez mis à jour votre application pour créer des rapports sur les événements. Maintenant, lorsque les utilisateurs interagissent avec votre application, elle envoie les données d'utilisation à Amazon Pinpoint. Vous pouvez afficher ces données dans la console, et vous pouvez la diffuser en continu sur Amazon Kinesis.

Ensuite, modifiez votre application afin de gérer les notifications push que vous envoyez avec Amazon Pinpoint. Consultez [Gestion des notifications push](#).

Gestion des notifications push

Les rubriques suivantes expliquent comment modifier votre application Swift, Android, React Native ou Flutter afin qu'elle reçoive les notifications push que vous envoyez avec Amazon Pinpoint.

Rubriques

- [Configuration des notifications push pour Amazon Pinpoint](#)
- [Gestion des notifications push](#)

Configuration des notifications push pour Amazon Pinpoint

Pour configurer Amazon Pinpoint de façon à pouvoir envoyer des notifications push à vos applications, vous devez d'abord fournir les informations d'identification qui autorisent Amazon Pinpoint à envoyer des messages à votre application. Les informations d'identification que vous fournissez dépendent du système de notification push que vous utilisez :

- Pour les applications iOS, vous fournissez un certificat SSL, que vous obtenez à partir du portail Apple Developer. Le certificat autorise Amazon Pinpoint à envoyer des messages à votre application via le service Apple Push Notification.
- Pour les applications Android, fournissez une clé d'API web que vous obtenez depuis la console Firebase. Ces informations d'identification autorisent Amazon Pinpoint à envoyer des messages à votre application via Firebase Cloud Messaging.

Après avoir obtenu les informations d'identification d'un canal de notification push, vous devez créer un projet dans Amazon Pinpoint et lui fournir les informations d'identification pour le service de notification push.

Rubriques

- [Configuration des notifications Swift Push](#)
- [Configuration des notifications push Android](#)
- [Configuration des notifications push Flutter](#)
- [Configuration des notifications push React Native](#)
- [Créer un projet dans Amazon Pinpoint](#)

Configuration des notifications Swift Push

Les notifications push des applications iOS sont envoyées à l'aide du service Apple Push Notification (APN). Avant de pouvoir envoyer des notifications push aux appareils iOS, vous devez créer un ID d'application sur le portail Apple Developer, ainsi que les certificats requis. Vous trouverez davantage d'informations sur cette procédure dans la section [Configuration des notifications push](#) de la documentation AWS Amplify.

Utilisation des jetons APN

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

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

Configuration des notifications push Android

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

Configuration des notifications push Flutter

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

Configuration des notifications push React Native

Les notifications push pour les applications React Native sont envoyées à l'aide de Firebase Cloud Messaging (FCM) pour Android et des APN pour iOS. Pour plus d'informations sur la réalisation de ces étapes, consultez la section Notifications push de la documentation [AWS d'Amplify JavaScript](#).

Créer un projet dans Amazon Pinpoint

Dans Amazon Pinpoint, un projet correspond à un ensemble de paramètres, de données, de campagnes et de segments qui partagent un objectif commun. Dans l'API Amazon Pinpoint, les projets sont également appelés applications. Cette section utilise le mot « projet » exclusivement pour désigner ce concept.

Pour commencer à envoyer des notifications push dans Amazon Pinpoint, vous devez créer un projet. Vous devez ensuite activer les canaux de notification push à utiliser en fournissant les informations d'identification appropriées.

Vous pouvez créer de nouveaux projets et configurer des canaux de notification push à l'aide de la console Amazon Pinpoint. Pour plus d'informations, consultez [Configuration des canaux de notification push Amazon Pinpoint](#) dans le Guide de l'utilisateur Amazon Pinpoint.

Vous pouvez également créer et configurer des projets à l'aide de l'[API Amazon Pinpoint](#), d'un [kit SDK AWS](#) ou de l'[AWS Command Line Interface](#) (AWS CLI). Pour créer un projet, utilisez la ressource Apps. Pour configurer des canaux de notification push, utilisez les ressources suivantes :

- [Canal APN](#) pour envoyer des messages aux utilisateurs d'appareils iOS en utilisant le service Apple Push Notification.
- [Canal ADM](#) pour envoyer des messages aux utilisateurs d'appareils Amazon Kindle Fire.
- [Canal Baidu](#) pour envoyer des messages aux utilisateurs de Baidu.
- [Canal GCM](#) pour envoyer des messages aux appareils Android à l'aide de Firebase Cloud Messaging (FCM), qui remplace Google Cloud Messaging (GCM).

Gestion des notifications push

Une fois que vous avez obtenu les informations d'identification requises pour envoyer des notifications push, vous pouvez mettre à jour vos applications afin qu'elles puissent recevoir des notifications push. Pour plus d'informations, consultez [Notifications push : démarrer](#) dans la documentation AWS Amplify.

Définition de votre audience pour 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 dirigez le message vers les points de terminaison qui représentent les membres de votre audience 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 de vous engager avec votre audience, la première étape consiste à ajouter des points de terminaison à votre projet Amazon Pinpoint.

Pour ajouter des points de terminaison, vous pouvez :

- Intégrer Amazon Pinpoint à vos client Android, iOS ou JavaScript afin que les points de terminaison soient ajoutés automatiquement lorsque les utilisateurs visitent votre application.
- Utiliser l'API Amazon Pinpoint pour ajouter des points de terminaison individuellement ou par lots.
- Importer des définitions de points de terminaison stockées en dehors d'Amazon Pinpoint.

Une fois que vous avez ajouté les points de terminaison, vous pouvez :

- Afficher des analyses sur votre audience dans la console Amazon Pinpoint.
- En savoir plus sur votre public en recherchant ou en exportant des données de point de terminaison.
- Définir des segments de public en fonction des attributs de point de terminaison, comme des données démographiques ou les intérêts des utilisateurs.
- Vous engager auprès de vos publics cibles avec des campagnes de messages personnalisés.
- Envoyer des messages directement à des listes de points de terminaison.

Utiliser les rubriques de cette section pour ajouter, mettre à jour et supprimer des points de terminaison à l'aide de l'API Amazon Pinpoint. Si vous souhaitez ajouter des points de terminaison automatiquement à partir de votre client Android, iOS ou JavaScript, consultez [Enregistrement des points de terminaison dans votre application](#) à la place.

Rubriques

- [Ajouter des points de terminaison à Amazon Pinpoint](#)

- [Association d'utilisateurs à des points de terminaison Amazon Pinpoint](#)
- [Ajout d'un lot de points de terminaison à Amazon Pinpoint](#)
- [Importation de points de terminaison dans Amazon Pinpoint](#)
- [Supprimer des points de terminaison d'Amazon Pinpoint](#)
- [Gérer le nombre maximal de points de terminaison d'une audience](#)

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.

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 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 peuvent inclure des données relatives à vos utilisateurs, leurs préférences, leurs appareils, les versions du client qu'ils utilisent, ou leurs emplacements. 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 cliente mobile ou JavaScript enregistre les points de terminaison automatiquement si vous intégrez Amazon Pinpoint en utilisant les kits SDK AWS Mobile ou la bibliothèque JavaScript AWS 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 à partir d'un client mobile ou JavaScript, consultez [Enregistrement des points de terminaison 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 en mode push Apple. L'adresse du point de terminaison est le jeton d'appareil fourni par le service de notification en mode push Apple.

AWS CLI

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

Exemple 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` correspond au chemin d'accès à un fichier JSON local contenant l'entrée pour le paramètre `--endpoint-request`.

Exemple 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 schéma [EndpointRequest](#) dans la Référence de l'API Amazon Pinpoint.

AWS SDK for Java

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

Exemple 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 obligatoires et fournissez le schéma JSON [EndpointRequest](#) en tant que 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=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
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"](#).

Association d'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 utilisateur, Amazon Cognito peut ajouter automatiquement les ID et les attributs utilisateur 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'ID utilisateur. 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 des messages à un segment d'utilisateurs, ou vous pouvez

envoyer un message directement à une liste d'ID utilisateur. 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.

Exemple 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* 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.
- *endpoint_request_file.json* correspond au chemin d'accès à un fichier JSON local contenant l'entrée pour le paramètre `--endpoint-request`.

Exemple 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'objet `User` dans le schéma [EndpointRequest](#) de la Référence de l'API Amazon Pinpoint.

AWS SDK for Java

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

Example Code

Pour ajouter un utilisateur à un point de terminaison, initialisez un objet [EndpointRequest](#) et transmettez-le à la méthode [updateEndpoint](#) du client `AmazonPinpoint`. 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 ajoutant un objet [EndpointUser](#) à l'objet `EndpointRequest` :

```
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 obtenir l'exemple complet du kit SDK, consultez [AddExampleUser.java](#) sur [GitHub](#).

HTTP

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

Exemple 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* 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 obligatoires et fournissez le schéma JSON [EndpointRequest](#) en tant que 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=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170  
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 définir un segment en important une liste d'ID utilisateur, consultez [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 ID utilisateur, consultez [Messages utilisateur](#) dans la Référence de l'API Amazon Pinpoint.

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"](#).

Ajout d'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 [Importation de 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.

Exemple 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* correspond à l'ID du projet Amazon Pinpoint dans lequel vous ajoutez ou mettez à jour les points de terminaison.
- *endpoint_batch_request_file.json* correspond au chemin d'accès à un fichier JSON local contenant l'entrée pour le paramètre `--endpoint-batch-request`.

Exemple 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 schéma [EndpointBatchRequest](#) dans la Référence de l'API Amazon Pinpoint.

AWS SDK for Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK for 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 obtenir l'exemple complet du kit SDK, consultez [AddExampleEndpoints.java](#) sur [GitHub](#).

HTTP

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

Exemple 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 *application-id* correspond à 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 obligatoires et fournissez le schéma JSON [EndpointBatchRequest](#) en tant que 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=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
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.

Importation de 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 compte AWS. Si vous avez besoin de davantage de bande passante pour vos tâches d'importation, vous pouvez envoyer une demande d'augmentation de quota de service à AWS 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.

- Un rôle AWS Identity and Access Management (IAM) qui accorde des autorisations de lecture Amazon Pinpoint à 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 JSON [EndpointRequest](#) dans la Référence de l'API Amazon Pinpoint.

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.

Exemple 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/*.

Exemple 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` correspond à l'emplacement dans Amazon S3 contenant 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-arn` correspond à l'Amazon Resource Name (ARN) du rôle IAM qui autorise Amazon Pinpoint à accéder en lecture au compartiment.
- `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 être soit `true`, soit `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.

Combinaisons RegisterEndpoints et DefineSegments

RegisterEndpoints	DefineSegments	Description
true	true	Amazon Pinpoint importera les points de terminaison et créera un segment contenant les points de terminaison.

RegisterEndpoints	DefineSegments	Description
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.

Exemple 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 for Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK for 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.getMessage());
            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.getMessage());
        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.

Exemple 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=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
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.

Exemple 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 obligatoires et fournissez le schéma JSON [ImportJobRequest](#) en tant que 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=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
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 correspond à l'emplacement dans Amazon S3 contenant 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-arn correspond à l'Amazon Resource Name (ARN) du rôle IAM qui autorise Amazon Pinpoint à accéder en lecture au compartiment.
- 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.

Exemple 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=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
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.

Supprimer des points de terminaison d'Amazon Pinpoint

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.

Exemple 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 for Java

Vous pouvez utiliser l'API Amazon Pinpoint dans vos applications Java en utilisant le client fourni par AWS SDK for 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();

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

    public static void deletePinEndpoint(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 obtenir l'exemple complet du kit SDK, consultez [DeleteEndpoint.java](#) sur [GitHub](#).

HTTP

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

Exemple 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.

Gérer le nombre maximal de points de terminaison d'une audience

Chaque membre de votre audience peut avoir un maximum de 15 points de terminaison associés à son UserID ; consultez [Quotas de points de terminaison](#). Si vous essayez d'ajouter un 16ème point de terminaison, selon le ChannelType, vous obtiendrez soit un résultat BadRequestException, soit vous réussirez en supprimant le point de terminaison dont la valeur EffectiveDate est la plus ancienne.

Ajouter un 16ème point de terminaison

- Si le nouveau type de canal pour le point de terminaison est SMS, PUSH, VOICE, EMAIL, CUSTOM ou IN_APP, le résultat BadRequestException est renvoyé car le membre de l'audience a atteint son nombre maximum 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](#).
- 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 ChannelType ADM, GCM, APNS, APNS_VOICE, APNS_VOIP_SANDBOX

ou BAIDU. Si ce n'est pas le cas, le résultat `BadRequestException` est renvoyé et un point de terminaison doit être supprimé avant de réessayer. Consultez [Supprimer des points de terminaison d'Amazon Pinpoint](#).

- Sinon, le point de terminaison dont la valeur `EffectiveDate` est la plus ancienne est défini sur `INACTIVE`, où l'attribut `ChannelType` est `ADM`, `GCM`, `APNS`, `APNS_VOIP`, `APNS_VOIP_SANDBOX` ou `BAIDU`.
 - L'attribut `UserId` 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 ajoutant à nouveau l'`UserId` au point de terminaison.

Accès aux données d'audience dans Amazon Pinpoint

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.

Pendant que votre public s'accroît et change, vos données de point de terminaison aussi. Pour afficher les dernières informations qu'Amazon Pinpoint possède sur votre audience, vous pouvez rechercher les points de terminaison individuellement, ou exporter tous les points de terminaison d'un projet Amazon Pinpoint. En affichant vos données de points de terminaison, vous pouvez en savoir plus sur les caractéristiques du public que vous enregistrez dans vos points de terminaison, par exemple :

- Les appareils et les plateformes de vos utilisateurs.
- Les fuseaux horaires de vos utilisateurs.
- Les versions de votre application installées sur les appareils de vos utilisateurs.
- Les emplacements de vos utilisateurs, tels que leurs villes ou pays.
- Tous les attributs ou métriques personnalisés enregistrés.

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

Avant de pouvoir rechercher des points de terminaison, vous devez les ajouter à votre projet Amazon Pinpoint. Pour ajouter des points de terminaison, consultez [Définition de votre audience pour Amazon Pinpoint](#).

Utilisez les rubriques de cette section pour rechercher ou exporter des points de terminaison à l'aide de l'API Amazon Pinpoint.

Rubriques

- [Recherche de points de terminaison à l'aide d'Amazon Pinpoint](#)
- [Exporter des points de terminaison d'Amazon Pinpoint](#)

- [Liste des ID de point de terminaison avec Amazon Pinpoint](#)

Recherche de points de terminaison à l'aide d'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 “Exportation de points de terminaison”](#).

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.

Exemple 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 for Java

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

Exemple 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.

Exporter des points de terminaison d'Amazon Pinpoint

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.

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.
- Un rôle AWS Identity and Access Management (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.

Exemple 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.

Exemple 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.

Exemple 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 for Java

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

Exemple 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 obtenir l'exemple complet du kit SDK, consultez [ExportEndpoints.java](#) sur [GitHub](#).

HTTP

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

Exemple 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=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170  
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.

Exemple 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=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
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 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.

Liste des ID de point 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. Par conséquent, si vous souhaitez exécuter ces opérations sur tous les points de terminaison d'un projet Amazon Pinpoint, la première étape consiste à répertorier tous les ID de points de terminaison appartenant à ce projet. Ensuite, vous pouvez itérer sur ces ID pour, par exemple, ajouter un attribut globalement ou supprimer tous les points de terminaison de votre projet.

L'exemple suivant utilise le kit AWS SDK for Java et effectue les opérations suivantes :

1. Appelle l'exemple de méthode `exportEndpointsToS3` de l'exemple de code dans [Exporter des points de terminaison d'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 les ID de points de terminaison dans 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 obtenir l'exemple complet du kit SDK, consultez [ListEndpoints.java](#) sur [GitHub](#).

Création de segments

Un segment d'utilisateurs représente un sous-ensemble de vos utilisateurs basé sur des caractéristiques communes, comme leur dernière date d'utilisation de votre application ou la plateforme de périphérique utilisée. 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

- [Préparation des segments](#)
- [Importation des segments](#)
- [Personnalisation de segments avec AWS Lambda](#)

Préparation des segments

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.

Préparation de segments avec le kit AWS SDK for Java

L'exemple suivant montre comment créer un segment avec le kit AWS SDK for Java.

```
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](#) sur [GitHub](#).

Importation des segments

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.

Importation d'un segment

L'exemple suivant montre comment importer un segment à l'aide du kit AWS SDK for 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 obtenir l'exemple complet du kit SDK, consultez [ImportingSegments.java sur GitHub](#).

Personnalisation de segments avec AWS Lambda

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

Vous pouvez utiliser une fonction AWS Lambda pour personnaliser une campagne Amazon Pinpoint en fonction de votre audience cible. Avec AWS Lambda, vous pouvez modifier le segment de la campagne au moment où Amazon Pinpoint envoie le message de la campagne.

AWS Lambda est un service de calcul qui permet d'exécuter du code sans avoir à mettre en service ni à gérer des 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 contrôlez mieux les destinataires et le contenu des messages. 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 [Création de canaux personnalisés dans Amazon Pinpoint](#). 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 des 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.

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 d'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** : ID de la variante de la campagne utilisée pour les tests A/B.
- **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** : mappage qui associe les ID et les 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.

L'instruction `console.log()` consigne 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 format fourni par les données d'événement, à savoir un mappage qui associe les ID et les définitions des 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.

Attribution d'une stratégie 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'invoquer, 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 stratégie suivante accorde au principal de service Amazon Pinpoint d'utiliser l'action `lambda:InvokeFunction` pour une campagne spécifique (*ID de campagne*) :

```
{
  "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 bloc `Condition` doit également inclure une clé `AWS:SourceAccount` qui contrôle quel compte AWS peut invoquer l'action.

Pour écrire une stratégie plus générique, utilisez des caractères génériques (*) correspondant à plusieurs caractères. Par exemple, vous pouvez utiliser le bloc `Condition` suivant pour autoriser n'importe quelle campagne d'un projet Amazon Pinpoint donné (*ID d'application*) 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 [Attribution d'une fonction Lambda à une campagne](#).

Octroi de l'autorisation d'invocation à Amazon Pinpoint

Vous pouvez utiliser AWS Command Line Interface (AWS CLI) pour ajouter des autorisations à la stratégie 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 rechercher les ID de vos campagnes à l'aide de la commande [get-campaigns](#) dans 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.

Attribution d'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.

Création de campagnes

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 essayer d'autres stratégies de campagne, configurez votre campagne comme test A/B. Un test A/B inclut deux ou plusieurs traitements du message ou du calendrier. 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 la section [Campagnes](#).

Création de campagnes standard

Une campagne standard envoie une notification push personnalisée à un segment spécifié selon un calendrier que vous définissez.

Création de campagnes à l'aide du AWS SDK for Java

L'exemple suivant montre comment créer une campagne avec le kit AWS SDK for Java.

```
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, consultez [CreateCampaign.java](#) on. [GitHub](#)

Création de campagnes de tests A/B

Un test A/B se comporte comme une campagne standard, mais vous permet de définir différents traitements pour le message ou le calendrier de la campagne.

Création de campagnes de tests A/B avec AWS SDK for Java

L'exemple suivant montre comment créer une campagne de test A/B avec le kit AWS SDK for 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
```

Utilisation de l'API SMS et voix Amazon Pinpoint, version 2

Note

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 Amazon Pinpoint SMS and Voice, consultez le nouveau [Guide de l'utilisateur Amazon Pinpoint SMS](#).

La rubrique suivante a été déplacée vers le nouveau guide de [l'utilisateur Amazon Pinpoint SMS](#).


- [Gestion des numéros de téléphone](#)
- [Gestion des identifiants d'expéditeur](#)
- [Gestion des pools](#)
- [Gestion des listes de désinscription](#)
- [Gestion des ensembles de configuration](#)
- [Gestion des mots clés](#)
- [Gestion des destinations d'événements](#)
- [Envoi de messages](#)

Amazon Pinpoint inclut une API (appelée API SMS et voix, 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 l'API SMS et voix 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 qui ont une architecture multi-locataires, tels que les fournisseurs de logiciels indépendants (ISV). 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. Toutefois, si vous migrez vers la version 2, vous bénéficierez de fonctionnalités supplémentaires, telles que la possibilité de créer des pools 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 actuellement être effectuées qu'à l'aide de la console Amazon Pinpoint. Par exemple, si vous souhaitez [vérifier un numéro de téléphone à utiliser lorsque votre compte est dans l'environnement de test \(sandbox\) SMS](#), ou si vous souhaitez vous [inscrire pour utiliser 10DLC](#), vous devez utiliser la console Amazon Pinpoint.

Cette section fournit des informations sur cette API et inclut des exemples d'utilisation. Vous trouverez également de la documentation de référence dans la [Référence de l'API SMS et voix, version 2](#).

Envoyer et valider des mots de passe à usage unique (OTP) avec Amazon Pinpoint

Amazon Pinpoint inclut une fonctionnalité de gestion des mots de passe unique (OTP). Vous pouvez utiliser cette fonctionnalité pour générer de nouveaux mots de passe uniques et les envoyer à vos destinataires par SMS. Vos applications peuvent ensuite appeler l'API Amazon Pinpoint pour vérifier ces mots de passe.

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, consultez [À propos du sandbox Amazon Pinpoint SMS](#) et [Demander un numéro de téléphone](#) dans le guide de l'utilisateur Amazon Pinpoint SMS.

Dans certains pays et régions, vous devez obtenir un numéro de téléphone ou un ID 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 [Mots de passe uniques](#). Si vous avez besoin d'aide pour déterminer les autorisations, consultez [Dépannage de la gestion des identités et des accès Amazon Pinpoint](#).

Envoi d'un message OTP

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 [Exigences particulières pour l'envoi de messages SMS à des destinataires en Inde](#) 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-message](#) commande 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 `5` `CodeLength` par le nombre de chiffres qui figureront dans le code OTP envoyé au destinataire.
- Remplacez `20` `ValidityPeriod` pouces par la durée, en minutes, pendant laquelle le code OTP sera valide.
- Remplacez `5` `AllowedAttempts` par le nombre de fois où le destinataire peut tenter sans succès de vérifier l'OTP.
- Remplacez `+1855550142` par l'identité `OriginationIdentity` d'origine utilisée pour envoyer le code OTP.
- Remplacez `+12065550007` par le numéro de `DestinationIdentity` téléphone auquel envoyer le code OTP.
- `SampleReferenceId` `ReferenceId` Remplacez-le par un numéro de référence unique pour la demande.

Réponse d'`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": "nvrimgq9kq4en96qgp0tlqli3og1at6aexample",
        "StatusCode": 200,
        "StatusMessage": "MessageId: nvrimgq9kq4en96qgp0tlqli3og1at6aexample"
      }
    }
  }
}
```

Validation d'un message OTP

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-message](#) commande 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.
- Remplacez `+12065550007` par le numéro de téléphone `DestinationIdentity` 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.
- Remplacez `01234` `Otp` par un `Otp` envoyé au. `DestinationIdentity`

Réponse d'`VerifyOtpMessage`

Lorsque vous envoyez une demande à l'API `VerifyOTPMMessage`, 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

Cette section contient des exemples de code qui montrent comment utiliser le kit SDK for Python (Boto3) pour envoyer et vérifier des codes OTP.

Génération d'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()
```

Envoi de 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")
```

Validation des 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")
```

Envoyer et récupérer des messages intégrés à l'application dans Amazon Pinpoint

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.

Cette section fournit des informations sur la demande de messages intégrés à l'application pour un point de terminaison dans votre application et sur l'interprétation du résultat de cette demande.

Récupération de messages dans l'application pour un point de terminaison

Vos applications peuvent appeler l'API [GetInAppMessages](#) pour récupérer tous les messages dans l'application 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 [Comprendre les réponses de l'API GetInAppMessages](#).

Vous pouvez utiliser les kits SDK AWS pour appeler l'opération GetInAppMessages. 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()
```

Comprendre les réponses de l'API `GetInAppMessages`

Lorsque vous appelez l'opération d'API [GetInAppMessages](#), elle renvoie une liste de messages auxquels le point de terminaison spécifié est autorisé à accéder. 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 davantage d'informations sur les composants de cette réponse.

Objet **InAppMessageCampaigns**

L'objet `InAppMessageCampaigns` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
<code>CampaignId</code>	Chaîne contenant le nom et l'ID de campagne unique de la campagne Amazon Pinpoint à partir de laquelle le message a été envoyé. Nom précède l'ID de campagne. Les deux	Créé automatiquement par Amazon Pinpoint lorsque vous créez la campagne.

Attribut	Description	Emplacement d'installation
	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 terminais on par campagne.	

Objet **InAppMessage**

L'objet InAppMessage contient les attributs suivants :

Attribut	Description	Emplacement d'installation
Content	Tableau contenant un objet InAppMessageContent , 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 l'appareil du destinataire. Les valeurs possibles sont :	

Attribut	Description	Emplacement d'installation
	<ul style="list-style-type: none"> • 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
<code>Alignment</code>	Chaîne qui indique l'alignement du texte de l'en-tête. Les valeurs possibles sont LEFT, CENTER et RIGHT.	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.

Attribut	Description	Emplacement d'installation
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	Un objet BodyConfig , qui contient des informations relatives au contenu principal du corps du message.	
HeaderConfig	Un objet HeaderConfig , qui contient des informations relatives à l'en-tête ou au titre du message.	
ImageUrl	L'URL de l'image qui apparaît dans le message.	
PrimaryBtn	Un objet InAppMessageButton qui contient des informations sur le bouton principal du message.	
SecondaryBtn	Un objet InAppMessageButton qui contient 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
<code>EndDate</code>	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 <code>Schedule</code> (si la campagne a été créée à l'aide de l'API ou d'un kit SDK).
<code>EventFilter</code>	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
<code>DefaultConfig</code>	Un objet DefaultButtonConfig qui contient 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.
<code>Android</code>	Un objet OverrideButtonConfig qui spécifie 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 <code>DefaultConfig</code> .	
IOS	Un objet OverrideButtonConfig qui spécifie le comportement du bouton sur les appareils iOS. Cette option remplace la configuration par défaut des boutons décrite dans l'objet <code>DefaultConfig</code> .	
Web	Un objet OverrideButtonConfig qui spécifie le comportement du bouton dans les applications Web. Cette option remplace la configuration par défaut des boutons décrite dans l'objet <code>DefaultConfig</code> .	

Objet `DefaultButtonConfig`

Un objet `DefaultButtonConfig` contient les attributs suivants :

Attribut	Description	Emplacement d'installation
<code>BackgroundColor</code>	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.
<code>BorderRadius</code>	Rayon de la bordure du bouton en pixels, exprimé	

Attribut	Description	Emplacement d'installation
	sous forme de nombre entier. 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 : <ul style="list-style-type: none">• 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 dont la 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).	

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
<code>ButtonAction</code>	L'action qui se produit lorsqu'un destinataire choisit un bouton dans le message intégré à l'application. Les valeurs possibles sont : <ul style="list-style-type: none">• <code>LINK</code> : un lien vers une destination web.• <code>DEEP_LINK</code> : un lien vers une page spécifique d'une application.• <code>CLOSE</code> : ignore le message.	Sur la base du contenu du modèle de message dans l'application spécifié pour la campagne.
<code>Link</code>	L'URL de destination d'un bouton. Non présent pour les boutons dont la valeur <code>ButtonAction</code> est <code>CLOSE</code> .	
<code>Text</code>	Le texte qui apparaît sur le bouton.	
<code>TextColor</code>	La couleur du texte du bouton, exprimée sous la forme d'une chaîne contenant un code couleur hexadécimal (par exemple : « <code>#000000</code> » pour le noir).	

Validation des numéros de téléphone dans 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 de numéro de téléphone

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 [Ajout d'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.

Utilisation du service de validation de numéro de téléphone

L'exemple suivant montre comment valider un numéro de téléphone à l'aide du AWS CLI. Pour plus d'informations, consultez [phone-number-validate](#) le manuel de référence des AWS CLI commandes. Pour des exemples de réponses de validation, voir [Réponses du service de validation de 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 de pays ou de région ISO à deux chiffres.

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**.

Réponses du service de validation de 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.

Envoi de messages transactionnels depuis vos applications

Utilisez l'API Amazon Pinpoint et les kits SDK AWS pour envoyer des messages transactionnels directement à partir de vos applications. 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.

Rubriques de cette section :

- [Envoi d'e-mails transactionnels](#)
- [Envoi de SMS](#)
- [Envoi de messages vocaux](#)
- [Envoi de notifications push](#)

Envoi d'e-mails transactionnels

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

- [En utilisant l' opération SendMessages de l'API Amazon Pinpoint : vous pouvez utiliser l'opération SendMessages de l'API](#) Amazon Pinpoint pour envoyer des messages sur tous les canaux pris en charge par Amazon Pinpoint, y compris les notifications 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'opération SendMessages vous permet également de remplacer le contenu de vos e-mails, et d'envoyer des e-mails à des ID de point de terminaison Amazon Pinpoint 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.

Rubriques de cette section :

- [Choix d'une méthode d'envoi d'e-mail](#)
- [Choix entre Amazon Pinpoint et Amazon Simple Email Service \(SES\)](#)
- [Envoyer un e-mail à l'aide de l'API Amazon Pinpoint](#)
- [Envoi d'un e-mail avec des en-têtes de désabonnement](#)

Choix d'une méthode d'envoi d'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 `SendMessage` de l'API Amazon Pinpoint.

Choix entre Amazon Pinpoint et Amazon Simple Email Service (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.

C#

Utilisez cet exemple pour envoyer des e-mails à l'aide du kit [AWS SDK for .NET](#). Cette procédure suppose que vous avez déjà installé et configuré le kit AWS SDK for .NET. Pour plus d'informations, consultez [Mise en route avec AWS SDK for .NET](#) dans le Guide du développeur AWS SDK for .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 de plus amples informations, veuillez consulter la section [Configurer les informations d'identification AWS](#) du Guide du développeur pour le kit AWS SDK for .NET .

Cet exemple de code a été testé à l'aide de la AWS SDK for .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...");
SendMessageResponse 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 for 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 for Java .

Cet exemple de code a été testé avec la version 2.3.1 et la AWS SDK for 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 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 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 = `
<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.sendMessage(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 for 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).

Envoi d'un e-mail avec des en-têtes de désabonnement

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. Le lien doit prendre en charge les requêtes HTTP POST pour traiter 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, procédez comme suit :

- Remplacez *AppId* par l'identifiant de votre application.
- Remplacez *richard_roe@example.com* par l'adresse e-mail du destinataire.
- Remplacez *https://example.com/unsub* 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"
}
]
}
}'
```

Envoi de SMS

Vous pouvez utiliser l'API Amazon Pinpoint pour envoyer des SMS à des numéros de téléphone ou des ID de point de terminaison spécifiques. 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.

C#

Utilisez cet exemple pour envoyer un SMS à l'aide du kit [AWS SDK for .NET](#). Cette procédure suppose que vous avez déjà installé et configuré le kit AWS SDK for .NET. Pour de plus amples informations, veuillez consulter [Mise en route](#) dans le Manuel du développeur AWS SDK for .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 for .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
        }
    }
};
SendMessageResponse response = await client.SendMessageAsync(sendRequest);
return response;
}
}

```

Java

Utilisez cet exemple pour envoyer un SMS à l'aide du kit [AWS SDK for 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 for 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 for 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.SendMessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessageResponse;
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.sendMessage(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 for 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": {
                "SMSGTemplate": {"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).

Envoi de messages vocaux

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 complet que vous pouvez utiliser pour envoyer des messages vocaux via l'API de SMS et voix Amazon Pinpoint à l'aide d'un kit SDK AWS.

Java

Utilisez cet exemple pour envoyer un message vocal à l'aide du kit [AWS SDK for 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 for 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, consultez [Configurer les informations d'identification AWS et la région pour le développement](#) dans le Guide du développeur AWS SDK for Java.

```
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.
 *
 * For more information, see the following documentation topic:
 *
 * 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 = "<speack>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></speack>";

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 obtenir l'exemple complet du kit SDK, consultez [SendVoiceMessage.java](#) sur [GitHub](#).

JavaScript (Node.js)

Utilisez cet exemple pour envoyer un message vocal à l'aide du kit SDK AWS pour JavaScript dans Node.js. Cette procédure suppose que vous avez déjà installé et configuré le kit SDK pour JavaScript dans Node.js.

Cet exemple suppose que vous utilisez un fichier d'informations d'identification partagé pour spécifier la clé d'accès et la clé d'accès secrète d'un utilisateur existant. Pour plus d'informations, consultez [Définition des informations d'identification](#) dans le Guide du développeur du kit SDK AWS pour JavaScript dans 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 for 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()
```

Envoi de notifications push

L'API Amazon Pinpoint peut envoyer des notifications push transactionnelles à des identifiants d'appareil spécifiques. Cette section contient des exemples de code complet que vous pouvez utiliser pour envoyer des SMS via l'API Amazon Pinpoint à l'aide d'un kit SDK AWS.

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

Note

Lorsque vous envoyez des notifications push via le service Firebase Cloud Messaging (FCM), utilisez le nom du service GCM dans votre appel à l'API Amazon Pinpoint. Le service

Google Cloud Messaging (GCM) a été interrompu par Google le 10 avril 2018. Toutefois, l'API Amazon Pinpoint utilise le nom du service GCM pour les messages qu'elle envoie via le service FCM afin de maintenir la compatibilité avec le code d'API écrit avant l'arrêt du service GCM.

JavaScript (Node.js)

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

Cet exemple suppose que vous utilisez un fichier d'informations d'identification partagé pour spécifier la clé d'accès et la clé d'accès secrète d'un utilisateur existant. Pour plus d'informations, consultez [Définition des informations d'identification](#) dans le Guide du développeur du kit SDK AWS pour JavaScript dans Node.js.

```
'use strict';

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

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

// The title that appears at the top of the push notification.
var title = 'Test message sent from Amazon Pinpoint.';

// The content of the push notification.
var message = 'This is a sample message sent from Amazon Pinpoint by using the '
    + 'AWS SDK for JavaScript in Node.js';

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

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

```
'service': 'GCM'
};

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

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

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

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

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

function CreateMessageRequest() {
    var token = recipient['token'];
    var service = recipient['service'];
    if (service == 'GCM') {
        var messageRequest = {
            'Addresses': {
                [token]: {
                    'ChannelType' : 'GCM'
                }
            },
            'MessageConfiguration': {
                'GCMMessage': {
                    'Action': action,
                    'Body': message,
                    'Priority': priority,
                    'SilentPush': silent,
                }
            }
        };
    }
}
```

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

```
var messageRequest = {
  'Addresses': {
    [token]: {
      'ChannelType' : 'ADM'
    }
  },
  'MessageConfiguration': {
    'ADMMessage': {
      'Action': action,
      'Body': message,
      'SilentPush': silent,
      'Title': title,
      'Url': url
    }
  }
};

return messageRequest
}

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

function SendMessage() {
  var token = recipient['token'];
  var service = recipient['service'];
  var messageRequest = CreateMessageRequest();

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

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

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

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

SendMessage()
```

Python

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

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

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

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

# The title that appears at the top of the push notification.
title = "Test message sent from Amazon Pinpoint."

# The content of the push notification.
message = ("This is a sample message sent from Amazon Pinpoint by using the "
          "AWS SDK for Python (Boto3).")
```



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

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

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

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

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

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

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

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

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

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

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

return message_request

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

# Send the message through the appropriate channel.
```

```
def send_message():

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

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

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

send_message()
```

Création de canaux personnalisés dans Amazon Pinpoint

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 les API en utilisant un webhook ou en appelant une fonction AWS Lambda.

Les segments vers lesquels vous envoyez des campagnes de canaux personnalisés 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).

Création d'une campagne qui envoie des messages à travers un canal personnalisé

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 a 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 :
- L'Amazon Resource Name (ARN) d'une fonction Lambda que vous souhaitez exécuter lors de l'exécution de la campagne.
 - URL du webhook auquel vous souhaitez envoyer des données de point de terminaison 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 [Personnalisation de segments avec AWS Lambda](#).

Comprendre les données d'événement qu'Amazon Pinpoint envoie vers des canaux personnalisés

Avant de créer une fonction Lambda qui envoie des messages à travers un canal personnalisé, vous devez vous familiariser 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 test A/B, 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.

Configuration de webhooks

Si vous utilisez un webhook pour envoyer des messages de canal personnalisé, l'URL du webhook doit commencer par « `https://` ». L'URL du webhook peut uniquement contenir des caractères alphanumériques, plus les symboles suivants : tiret (-), point (.), trait de soulignement (_), tilde (~), point d'interrogation (?), barre oblique ou solidus (/), dièse ou signe de hachage (#) et point-virgule (;). L'URL doit être conforme à la norme [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 ID de compte AWS.

Configuration des fonctions Lambda

Cette section présente les étapes à suivre pour créer une fonction Lambda qui envoie des messages à travers un canal personnalisé. Tout d'abord, vous créez la fonction. Ensuite, vous ajoutez une stratégie d'exécution à cette dernière. 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 consigne le nombre de points de terminaison de chaque type de point de terminaison dans 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 afficher ces métriques dans la section Métriques de la console CloudWatch. Les métriques sont disponibles dans l'espace de noms PinpointCustomChannelExecution.

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 le partage oui/non 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.

Yes/no split [Info](#) ✕

Select a condition type
Event ▼

Choose a journey message activity and event
lambdaFunction (Custom) ▼
Call to function or webhook response ▼

Response [Info](#)

Attribute	Value
<input type="text"/>	<input type="text"/>

[+ Add condition](#)

Condition evaluation
The amount of time that Amazon Pinpoint waits before it evaluates the conditions.

Evaluate after ▼

▼

Description - *optional*

Accorder une autorisation Amazon Pinpoint pour invoquer la fonction Lambda

Vous pouvez utiliser AWS Command Line Interface (AWS CLI) pour ajouter des autorisations à la stratégie 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 région AWS où 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.

Diffusion d'événements Amazon Pinpoint sur Kinesis

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

- [Configuration de la diffusion en continu des événements](#)
- [Événements d'application](#)
- [Événements de campagne](#)
- [Événements de parcours](#)
- [Événements d'e-mail](#)
- [Événements SMS](#)

Configuration de la diffusion en continu des événements

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.

Prérequis

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 [Intégration de votre application à Amazon Pinpoint](#).
- 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 AWS Identity and Access Management (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-stream](#) commande. 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 for 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 crée un objet [WriteEventStream](#) qui stocke les ARN du flux Kinesis et du 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'ID d'application et utilise l'exemple de méthode précédent, `createEventStream`, pour attribuer un flux à chaque application :

```
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.

Désactivation de la diffusion en continu des événements

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 for Java

Utilisez la [deleteEventStream](#) méthode du client Amazon Pinpoint :

```
pinClient.deleteEventStream(new DeleteEventStreamRequest().withApplicationId(appId));
```

Événements d'application

Une fois que vous avez intégré votre application (application) à Amazon Pinpoint, Amazon Pinpoint peut diffuser des données d'événements concernant l'activité des utilisateurs, les événements personnalisés et les envois de messages pour l'application.

Exemple


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 qui sont inclus dans le flux d'événements de l'application.

Attribut	Description
event_type	Type d'événement. Les valeurs possibles sont : <ul style="list-style-type: none">• <code>_session.start</code> : le point de terminaison a commencé une nouvelle session.• <code>_session.stop</code> : le point de terminaison a mis fin à une session.• <code>_userauth.sign_in</code> : point de terminaison connecté à votre application.• <code>_userauth.sign_up</code> : le nouveau point de terminaison a terminé le processus d'inscription dans votre application.• <code>_userauth.auth_fail</code> : le point de terminaison a tenté de se connecter à votre application, mais n'a pas pu terminer le processus.• <code>_monetization.purchase</code> : le point de terminaison a effectué un achat dans votre application.• <code>_session.pause</code> : 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.• <code>_session.resume</code> : 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.

Attribut	Description
<code>arrival_timestamp</code>	Heure à laquelle l'événement a été reçu par Amazon Pinpoint, affichée sous la forme Heure Unix en millisecondes.
<code>event_version</code>	Version du schéma JSON de l'événement. <div data-bbox="829 478 1507 842"><p> Tip</p><p>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.</p></div>
<code>application</code>	Informations sur le projet Amazon Pinpoint associé à l'événement. Pour plus d'informations, consultez le tableau Application .
<code>client</code>	Informations sur le point de terminaison qui a signalé l'événement. Pour plus d'informations, consultez le tableau Client .
<code>device</code>	Informations sur l'appareil qui a signalé l'événement. Pour plus d'informations, consultez le tableau Appareil .
<code>session</code>	Informations sur la session qui a généré l'événement. Pour plus d'informations, consultez le tableau Session .
<code>attributes</code>	Attributs associés à l'événement. Pour les événements rapportés par vos applications, cet objet inclut les attributs personnalisés que vous définissez.

Attribut	Description
<code>metrics</code>	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
<code>app_id</code>	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
<code>cognito_identity_pool_id</code>	ID de la réserve d'identités Amazon Cognito auquel le point de terminaison est associé.
<code>package_name</code>	Nom du package d'application, par exemple <code>com.example.my_app</code> .
<code>sdk</code>	Informations sur le kit SDK qui a été utilisé pour signaler l'événement. Pour plus d'informations, consultez le tableau kit SDK .
<code>title</code>	Nom de l'application.
<code>version_name</code>	Nom de la version de l'application (par exemple, <code>V2.5</code>).
<code>version_code</code>	Numéro de version de l'application, tel que <code>3</code> .

Kit SDK

Informations sur le kit SDK qui a été utilisé pour signaler l'événement.

Attribut	Description
<code>name</code>	Nom du kit SDK qui a été utilisé pour signaler l'événement.
<code>version</code>	Version du kit SDK.

Client

Inclut des informations sur le point de terminaison qui a généré l'événement.

Attribut	Description
<code>client_id</code>	ID du point de terminaison.
<code>cognito_id</code>	Jeton d'ID Amazon Cognito associé au point de terminaison.

Device

Inclut des informations sur l'appareil du point de terminaison qui a généré l'événement.

Attribut	Description
<code>locale</code>	Informations sur les paramètres de langue et de région de l'appareil du point de terminaison. Pour plus d'informations, consultez le tableau Locale .
<code>make</code>	Fabricant de l'appareil du point de terminaison.
<code>model</code>	Identifiant du modèle de l'appareil du point de terminaison.
<code>platform</code>	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>code</code>	Identifiant local associé à l'appareil.
<code>country</code>	Pays ou région associé aux paramètres régionaux de l'appareil.
<code>language</code>	Langue associée aux paramètres régionaux de l'appareil.

Plateforme

Informations sur le système d'exploitation de l'appareil du point de terminaison.

Attribut	Description
<code>name</code>	Nom du système d'exploitation sur l'appareil.
<code>version</code>	Version du système d'exploitation de l'appareil.

Session

Inclut des informations sur la session qui a généré l'événement.

Attribut	Description
<code>session_id</code>	ID unique qui identifie la session.
<code>start_timestamp</code>	Date et heure de début de la session, affichées sous la forme Heure Unix en millisecondes.
<code>stop_timestamp</code>	Date et heure auxquelles la session s'est terminée, affichées sous la forme Heure Unix en millisecondes.

Événements de campagne

Si vous utilisez Amazon Pinpoint pour envoyer des campagnes via un canal, Amazon Pinpoint peut diffuser des données d'événements sur ces campagnes. Cela inclut des données d'événement pour tous les messages électroniques ou SMS que vous envoyez à partir d'une campagne. Pour plus d'informations sur les données diffusées par Amazon Pinpoint pour ces types de messages, consultez [the section called “Événements d'e-mail”](#) et [the section called “Événements SMS”](#).

Exemple d'événement

L'objet JSON d'un événement contient les données illustrées 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	<p>Type d'événement. Les valeurs possibles sont :</p> <ul style="list-style-type: none">• <code>_campaign.send</code> : Amazon Pinpoint a exécuté la campagne.• <code>_campaign.opened_notification</code> : pour les campagnes de notification push, ce type d'événement indique que le destinataire a utilisé la notification pour l'ouvrir.• <code>_campaign.received_foreground</code> : 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.• <code>_campaign.received_background</code> : 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.

 **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 avec AWS

Attribut	Description
	<p>Amplify. Consultez Connexion de votre application frontale à Amazon Pinpoint avec AWS Amplify.</p>
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 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.

Attribut	Description
<code>session</code>	Informations sur la session qui a généré l'événement. Pour les campagnes, cet objet est vide.
<code>attributes</code>	<p>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.</p> <p>Pour plus d'informations, consultez le tableau Attributs.</p>
<code>client_context</code>	Contient un objet custom, qui contient une propriété <code>endpoint</code> . La propriété <code>endpoint</code> contient le contenu de l'enregistrement de point de terminaison pour le point de terminaison auquel la campagne a été envoyée.
<code>awsAccountId</code>	ID du compte AWS 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.

Attributs

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 test A/B, 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	<p>Mode de livraison de la campagne. Ne confondez pas cet attribut avec le champ <code>ChannelType</code> spécifié sous la propriété <code>endpoint</code> de <code>client_context</code>. Le champ <code>ChannelType</code> est généralement basé sur le point de terminaison auquel le message est envoyé.</p> <p>Pour les canaux qui ne prennent en charge qu'un seul type de point de terminaison, les champs <code>delivery_type</code> et <code>ChannelType</code> ont la même valeur. Par exemple, pour le canal e-mail, les champs <code>delivery_type</code> et <code>ChannelType</code> ont la même valeur que <code>EMAIL</code>.</p>

Attribut	Description
	<p>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 <code>delivery_type</code> identifie un événement de distribution personnalisé, CUSTOM, et <code>ChannelType</code> 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 Création de canaux personnalisés.</p> <p>Les valeurs possibles sont :</p> <ul style="list-style-type: none">• EMAIL• SMS• ADM• APNS• APNS_SANDBOX• APNS_VOIP• APNS_VOIP_SANDBOX• VOICE• GCM• BAIDU• PUSH• CUSTOM
<code>campaign_id</code>	ID unique de la campagne à partir de laquelle le message a été envoyé.

Attribut	Description
campaign_send_status	<p>Indique le statut de la campagne pour le point de terminaison cible. Les valeurs possibles incluent :</p> <ul style="list-style-type: none">• 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 terminaison 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
	<p>envoyé au point de terminaison à partir de cette campagne.</p> <ul style="list-style-type: none">• 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 distribution personnalisée a échoué.• RECOMMENDATION_FAILURE : le moteur de recommandation 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
<code>client_id</code>	ID du point de terminaison auquel la campagne a été envoyée.

Événements de parcours

Si vous publiez un parcours, Amazon Pinpoint peut diffuser des données d'événement sur le parcours. Cela inclut des données d'événement pour tous les e-mails, SMS, notifications push ou messages personnalisés que vous envoyez à partir du parcours.

Consultez les informations suivantes pour obtenir des informations sur les données diffusées par Amazon Pinpoint :

- Pour les e-mails, consultez [the section called “Événements d'e-mail”](#).
- Pour les SMS, consultez [Événements SMS](#).

Exemple d'événement

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
<code>event_type</code>	Type d'événement. Pour les événements de parcours, la valeur de cet attribut est toujours <code>_journey.send</code> , ce qui indique que le parcours a été exécuté par Amazon Pinpoint.
<code>event_timestamp</code>	Heure à laquelle l'événement a été signalé, affichée sous la forme Heure Unix en millisecondes.
<code>arrival_timestamp</code>	Heure à laquelle l'événement a été reçu par Amazon Pinpoint, affichée sous la forme Heure Unix en millisecondes.
<code>event_version</code>	Version du schéma JSON de l'événement.

Attribut	Description
	<p> Tip</p> <p>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.</p>
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
<code>awsAccountId</code>	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
<code>app_id</code>	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
<code>sdk</code>	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
<code>client_id</code>	ID du point de terminaison.

Attributs

Inclut des informations sur le parcours qui a généré l'événement.

Attribut	Description
<code>journey_run_id</code>	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	<p>Indique l'état de remise du message qui est associé à l'événement. Les valeurs possibles incluent :</p> <ul style="list-style-type: none">• 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 distribution personnalisée a échoué.• FAILURE_PERMANENT : une défaillance permanente s'est produite lors de l'envoi vers le point de terminaison. <div data-bbox="862 936 1507 1728" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Tip</p><p>Vous pouvez filtrer les événements dont le statut FAILURE_PERMANENT est <code>journey_send_status_code</code> défini sur 403 pour déterminer s'il existe une politique d'accès et une violation de rôle. Pour les campagnes sortantes avec voix, ces exceptions sont typiques des cas où le rôle d'exécution de campagnes Connect liant les campagnes Amazon Pinpoint Journeys à Amazon Connect est supprimé par inadvertance pour les exécutions de trajets en vol.</p></div> <ul style="list-style-type: none">• THROTTLED : l'envoi a été limité.

Attribut	Description
	<ul style="list-style-type: none"> • 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'estimation 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'événement.
journey_activity_id	ID unique de l'activité de parcours qui a généré l'événement.
journey_activity_type	<p>Type d'activité lié au parcours de l'événement. Cela peut être EMAIL, SMS, PUSH, CONTACT_CENTER ou CUSTOM.</p> <div data-bbox="829 1514 1507 1732" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>VOICE n'est pas un type d'activité de parcours pris en charge.</p> </div>
journey_send_status_message	Description de l'état de l'événement d'envoi.

Attribut	Description
<code>journey_send_status_code</code>	Le code d'état HTTP de la demande.

Événements d'e-mail

Lorsque vous envoyez des e-mails, Amazon Pinpoint peut diffuser des données qui fournissent des informations supplémentaires sur les types d'événements pour ces messages :

- Envois
- Messages délivrés
- Retours à l'expéditeur
- Réclamations
- Messages ouverts
- Clics
- Rejets
- Désabonnement
- Échec du rendu

Les types d'événements de la liste précédente sont expliqués en détail dans [Attributs 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.

Exemples d'événements

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": "0200000073rn bmd1-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": "0200000073rn bmd1-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": "0200000073rn bmd1-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-mbv dg3uo-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 qui sont inclus dans les données de flux d'événements générés par Amazon Pinpoint lorsque vous envoyez un e-mail.

Attribut	Description
event_type	<p>Type d'événement. Les valeurs possibles sont :</p> <ul style="list-style-type: none">• <code>_email.send</code> : Amazon Pinpoint a accepté le message et a tenté de le remettre au destinataire.• <code>_email.delivered</code> : le message a été remis au destinataire.• <code>_email.rejected</code> : Amazon Pinpoint a déterminé que le message contenait un programme malveillant et n'a pas tenté de l'envoyer.• <code>_email.hardbounce</code> : un problème permanent a empêché Amazon Pinpoint de délivrer le message. Amazon Pinpoint ne tentera pas de remettre le message à nouveau.• <code>_email.softbounce</code> : un problème temporaire 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 <code>SOFTBOUNCE</code>.• <code>_email.complaint</code> : 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 indésirable » de son client de messagerie).

Attribut	Description
	<ul style="list-style-type: none">• <code>_email.open</code> : le destinataire a reçu le message et l'a ouvert.• <code>_email.click</code> : le destinataire a reçu le message et a cliqué sur un lien qu'il contient.• <code>_email.unsubscribe</code> : le destinataire a reçu le message et a cliqué sur le lien de désabonnement qu'il contient.• <code>_email.rendering_failure</code> : l'e-mail n'a pas été envoyé en raison d'un échec de rendu. 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.
<code>event_timestamp</code>	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.
<code>arrival_timestamp</code>	Heure à laquelle l'événement a été reçu par Amazon Pinpoint, affichée sous la forme Heure Unix en millisecondes.
<code>event_version</code>	Version du schéma JSON de l'événement. <div data-bbox="829 1415 1507 1780"><p> Tip</p><p>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.</p></div>

Attribut	Description
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.
device	<p>Informations sur l'appareil qui a signalé l'événement. Pour plus d'informations, consultez le tableau Appareil.</p> <p>Pour les événements d'e-mail, cet objet est vide.</p>
session	Pour les événements d'e-mail, cet objet est vide.
attributes	<p>Attributs associés à l'événement. Pour plus d'informations, consultez le tableau Attributs.</p> <p>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 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 informations liées au message lui-même.</p>

Attribut	Description
<code>client_context</code>	Pour les événements de messagerie, cet objet contient un objet custom, qui contient un attribut <code>legacy_identifieur</code> . La valeur de l'attribut <code>legacy_identifieur</code> est l'ID du projet à partir de laquelle le message a été envoyé.
<code>facets</code>	Informations supplémentaires sur le message, telles que les en-têtes d'e-mail. Consultez le tableau Facettes pour plus d'informations.
<code>awsAccountId</code>	ID du compte AWS qui a été utilisé pour envoyer le message.

Application

Inclut les informations sur le projet Amazon Pinpoint auquel l'événement est associé.

Attribut	Description
<code>app_id</code>	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
<code>sdk</code>	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 <code>_email.click</code> , 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 <code>received</code> , <code>opened</code> ou <code>clicked</code> .
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 <code>0</code> .
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 <code>_email.click</code> , 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

Attribut	Description
	événements, cette valeur représente le type d'événement, tel que <code>received</code> , <code>delivered</code> ou <code>opened</code> .
<code>journey_id</code>	ID unique du parcours qui a envoyé le message.
<code>journey_activity_id</code>	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
<code>client_id</code>	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
<code>email_channel</code>	Contient un objet <code>mail_event</code> , qui contient deux objets : <code>mail</code> 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.
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	<p>Objet qui contient plusieurs paires nom-valeur correspondant aux en-têtes du message. Cet objet contient généralement des informations sur les en-têtes suivants :</p> <ul style="list-style-type: none">• From : adresse e-mail de l'expéditeur.• To : adresse e-mail du destinataire.• Subject : ligne Objet de l'e-mail. <div data-bbox="862 1356 1507 1623" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Tip</p><p>L'en-tête du sujet n'est pas inclus pour les événements Campaign_email.send.</p></div> <ul style="list-style-type: none">• MIME-Version : indique que le message est au format MIME. Si cet en-tête est présent, la valeur est toujours 1.0.

Attribut	Description
<code>common_headers</code>	<ul style="list-style-type: none"><code>Content-Type</code> : type de média MIME du contenu du message. <p>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.</p>

Événements SMS

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. 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 distribution des messages sortants. Après 72 heures, si Amazon Pinpoint n'a pas reçu d'événement final de la part d'un opérateur, le service renvoie automatiquement un `record_status UNKNOWN` car nous ne savons pas ce qu'il est advenu de ce message.

Exemple

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"
},
"metrics": {
  "price_in_millicents_usd": 645.0
},
"awsAccountId": "123456789012"
}
```

Attributs d'événement SMS

Cette section définit les attributs qui sont inclus dans les données de flux d'événements générés par Amazon Pinpoint lorsque vous envoyez un SMS.

Événement

Attribut	Description
event_type	Type d'événement. Les valeurs possibles sont : <ul style="list-style-type: none"><code>_SMS.BUFFERED</code> : le message est toujours en cours de remise au destinataire.<code>_SMS.SUCCESS</code> : le message a été accepté par l'opérateur/remis au destinataire.<code>_SMS.FAILURE</code> : 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 <code>attributes.record_status</code> .

Attribut	Description
	<ul style="list-style-type: none">• <code>_SMS.OPTOUT</code> : 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. <div data-bbox="829 814 1507 1178" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> 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.</p></div>
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
<code>session</code>	Pour les événements SMS, cet objet est vide.
<code>attributes</code>	<p>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.</p> <p>Pour plus d'informations, consultez le tableau Attributs.</p>
<code>metrics</code>	Métriques supplémentaires associées à l'événement. Pour plus d'informations, consultez le tableau Métriques .
<code>awsAccountId</code>	ID du compte AWS 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
<code>app_id</code>	ID unique du projet Amazon Pinpoint qui a signalé l'événement.
<code>sdk</code>	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
<code>sender_request_id</code>	ID unique associé à la demande d'envoi du SMS.
<code>campaign_activity_id</code>	ID unique de l'activité au sein de la campagne.
<code>origination_phone_number</code>	Numéro de téléphone à partir duquel le message a été envoyé.
<code>destination_phone_number</code>	Numéro de téléphone auquel vous avez tenté d'envoyer le message.
<code>record_status</code>	Informations supplémentaires sur l'état du message. Les valeurs possibles incluent : <ul style="list-style-type: none">• 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 indisponible. Par exemple, l'appareil peut être éteint ou déconnecté du réseau. Vous pouvez

Attribut	Description
	<p>réessayer d'envoyer le message ultérieurement.</p> <ul style="list-style-type: none">• 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
	<ul style="list-style-type: none">• 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 Demander une augmentation de votre quota mensuel de dépenses en SMS du Guide de l'utilisateur Amazon Pinpoint.• 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 connaître tous les pays pris en charge, consultez Pays et régions pris en charge (canal SMS).• ACCOUNT_IN_SANDBOX : votre compte est dans l'environnement de test (sandbox) et il peut uniquement envoyer des messages vers

Attribut	Description
	<p>des numéros de destination vérifiés. Vous pouvez vérifier le numéro de destination dans la console Amazon Pinpoint ou lancer le processus de transfert du compte hors de l'environnement de test (sandbox). Consultez la section Passer de l'environnement de test (sandbox) SMS Amazon Pinpoint à la production.</p> <ul style="list-style-type: none">• RATE_EXCEEDED : vous avez tenté d'envoyer un message trop rapidement et vous avez été limité. Vous devez ralentir votre débit d'appels. Pour plus de détails sur nos limites, consultez Limites de parties de message par seconde (MPS).• 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.• 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.

Attribut	Description
	<ul style="list-style-type: none">• 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
<code>number_of_message_parts</code>	<p>Nombre de parties de message créées par Amazon Pinpoint pour envoyer le message.</p> <p>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.</p>
<code>message_id</code>	ID unique généré par Amazon Pinpoint lorsqu'il accepte le message.
<code>message_type</code>	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'opération SendMessage de l'API Amazon Pinpoint.
<code>campaign_id</code>	ID unique de la campagne Amazon Pinpoint qui a envoyé le message.

Client

Présente des informations sur le client d'application installé sur l'appareil qui a signalé l'événement.

Attribut	Description
<code>client_id</code>	Pour les événements générés par les applications, cette valeur est l'ID unique du client d'application installé sur l'appareil. Cet ID est

Attribut	Description
	automatiquement généré par les kits AWS Mobile SDK for iOS et AWS Mobile SDK for 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.
<code>cognito_id</code>	ID unique attribué au client d'application dans la réserve d'identités Amazon Cognito utilisée par votre application.


Device

Présente des informations sur l'appareil qui a signalé l'événement.

Attribut	Description
<code>locale</code>	Paramètres régionaux du périphérique.
<code>make</code>	Fabricant du périphérique, tel qu'Apple ou Samsung.
<code>model</code>	Modèle du périphérique, tel qu'iPhone.
<code>platform</code>	Plateforme du périphérique, telle qu'ios ou android.

Métriques

Comprend des informations sur les métriques associées à l'événement.

Attribut	Description
<code>price_in_millicents_usd</code>	<p>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 ($645 / 1000 = 0,645 \text{ ¢} = 0,00645 \text{ \\$}$).</p> <div data-bbox="829 562 1507 829"><p> Note</p><p>Cette propriété n'apparaît pas pour les messages ayant un <code>event_type</code> <code>_SMS.BUFFERED</code>.</p></div>

Interrogation des données d'analyse Amazon Pinpoint

En plus d'utiliser les pages analytiques sur la console Amazon Pinpoint, vous pouvez utiliser l'API Amazon Pinpoint afin d'interroger les données analytiques pour un sous-ensemble de métriques standard qui fournissent des détails sur les tendances liées à l'implication des utilisateurs, au rayonnement de la campagne, et bien plus encore. Ces métriques, également appelées indicateurs de performance clés (KPI), sont des valeurs mesurables qui peuvent vous aider à surveiller et évaluer les performances de vos projets, campagnes et parcours.

Si vous utilisez les API pour interroger les données analytiques, vous pouvez analyser les données à l'aide de l'outil de création de rapports de votre choix, sans avoir à vous connecter à la console Amazon Pinpoint ou à analyser les données d'événement 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.

Métriques prises en charge

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. 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. Pour accéder aux données d'une métrique d'application, utilisez la ressource [Métriques d'application](#) de l'API Amazon Pinpoint.
- **Métriques de campagne** : ces métriques fournissent des informations sur les performances de chaque campagne. Par exemple, vous pouvez utiliser une métrique de campagne pour voir le

nombre de points de terminaison vers lesquels un message de campagne a été envoyé. Pour accéder aux données d'une métrique de campagne, utilisez la ressource [Métriques de campagne](#) de l'API Amazon Pinpoint.

- 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. Pour accéder aux données d'une métrique d'engagement du parcours, utilisez la ressource [Métriques d'engagement du parcours](#) de l'API Amazon Pinpoint.
- 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 poursuivant les activités du parcours. Pour accéder aux données d'une métrique d'exécution du parcours, utilisez la ressource [Métriques d'exécution du parcours](#) de l'API Amazon Pinpoint.
- 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 de l'activité du parcours pour déterminer le nombre de participants terminant une activité. Pour accéder aux données d'une métrique d'exécution de l'activité du parcours, utilisez la ressource [Métriques d'exécution de l'activité du parcours](#) de l'API Amazon Pinpoint.

Pour obtenir la liste complète des métriques standard que vous pouvez interroger par programme, consultez [Métriques standard](#).

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 [Diffusion d'événements Amazon Pinpoint sur Kinesis](#).

Principes de base de la requête

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 [Métriques standard](#).

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 [Utilisation des résultats des requêtes](#).

Politiques IAM pour l'interrogation de données analytiques Amazon Pinpoint

À l'aide de l'API Amazon Pinpoint, vous pouvez interroger les données analytiques pour un sous-ensemble de métriques standard, également appelées indicateurs de performance clés (KPI) 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 de AWS Identity and Access Management (IAM) qui définissent les autorisations des rôles ou utilisateurs IAM 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 analytiques, vous pouvez utiliser la AWS Management Console, l'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 stratégie suivante permet l'accès par programmation à toutes les données d'analyse de tous vos projets, campagnes et parcours dans toutes les régions AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "QueryAllAnalytics",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:GetApplicationDateRangeKpi",
        "mobiletargeting:GetCampaignDateRangeKpi",
        "mobiletargeting:GetJourneyDateRangeKpi",
        "mobiletargeting:GetJourneyExecutionMetrics",
        "mobiletargeting:GetJourneyExecutionActivityMetrics"
      ],
      "Resource": [
        "arn:aws:mobiletargeting:*:accountId:apps/*/kpis/*",
        "arn:aws:mobiletargeting:*:accountId:apps/*/campaigns/*/kpis/*",
        "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/kpis/*",
        "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/execution-
metrics",
        "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/activities/*/
execution-metrics"
      ]
    }
  ]
}
```

Où *accountId* est votre ID de compte AWS.

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 prendre en charge cette pratique et mettre en œuvre un contrôle plus granulaire, vous pouvez restreindre l'accès programmatique aux données d'analyse pour un projet particulier dans une région AWS spécifique, par exemple :

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

```
    "Effect": "Allow",
    "Action": [
        "mobiletargeting:GetApplicationDateRangeKpi",
        "mobiletargeting:GetCampaignDateRangeKpi",
        "mobiletargeting:GetJourneyDateRangeKpi",
        "mobiletargeting:GetJourneyExecutionMetrics",
        "mobiletargeting:GetJourneyExecutionActivityMetrics"
    ],
    "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/kpis/*",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/*/"
kpis/*",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/"
kpis/*",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/"
execution-metrics",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/"
activities/*/execution-metrics"
    ]
}
]
```

Où :

- *region* est le nom de la région AWS qui héberge le projet.
- *accountId* est votre ID de compte AWS.
- *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 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "QueryCampaignAnalytics",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetCampaignDateRangeKpi",
      "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/"
campaigns/campaignId/kpis/*"
    }
  ]
}
```



```

    }
  ]
}

```

Où :

- *region* est le nom de la région AWS qui héberge le projet.
- *accountId* est l'ID de votre Compte AWS.
- *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 :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "QueryJourneyAnalytics",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:GetJourneyDateRangeKpi",
        "mobiletargeting:GetJourneyExecutionMetrics",
        "mobiletargeting:GetJourneyExecutionActivityMetrics"
      ],
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/
journeys/journeyId/kpis/*",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/
journeys/journeyId/execution-metrics",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/
journeys/journeyId/activities/*/execution-metrics"
      ]
    }
  ]
}

```

Où :

- *region* est le nom de la région AWS qui héberge le projet.
- *accountId* est votre ID de compte AWS.
- *projectId* est l'identifiant du projet associé au parcours.
- *projectId* est l'identifiant du parcours 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](#).

Métriques analytiques Amazon Pinpoint standard

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

- [Métriques d'application pour les campagnes](#)
- [Métriques d'application pour les e-mails transactionnels](#)
- [Métriques d'application pour les SMS transactionnels](#)
- [Métriques de campagne](#)
- [Métriques d'engagement du parcours](#)
- [Métriques d'exécution du parcours](#)
- [Métriques d'exécution de l'activité du parcours](#)
- [Métriques relatives à l'exécution des parcours et des campagnes](#)

Métriques d'application 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.

Métrique	Kpi-name	Description
		<p>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.</p>
Delivery rate, grouped by date (Taux de remise, regroupement par date)	successful-delivery-rate-grouped-by-date	<p>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.</p> <p>Cette métrique est calculée de la façon suivante : le nombre de messages envoyés par toutes les campagnes pour un projet et remis aux destinataires, 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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métrique	Kpi-name	Description
Email open rate (Taux d'ouverture d'e-mail)	<code>email-open-rate</code>	<p>Pour toutes les campagnes associées à un projet, pourcentage d'e-mails ayant été ouverts par les destinataires.</p> <p>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 destinataires, divisé par le nombre d'e-mails qui ont été envoyés par l'ensemble de ces campagnes et remis aux destinataires.</p>

Métrique	Kpi-name	Description
Email open rate, grouped by campaign (Taux d'ouverture d'e-mail, regroupement par campagne)	<code>email-open-rate-grouped-by-campaign</code>	<p>Pour chaque campagne associée à un projet, pourcentage d'e-mails ayant été ouverts par les destinataires.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID de campagne (<code>CampaignId</code>), une chaîne qui identifie de façon unique une campagne.</p>
Endpoint deliveries (Remises au point de terminaison)	<code>unique-deliveries</code>	<p>Pour toutes les campagnes associées à un projet, nombre de points de terminaison uniques auxquels les messages ont été envoyés.</p>

Métrique	Kpi-name	Description
Endpoint deliveries, grouped by campaign (Remises au point de terminaison, regroupement par campagne)	unique-deliveries-grouped-by-campaign	<p>Pour chaque campagne associée à un projet, nombre de points de terminais on uniques auxquels les messages ont été envoyés.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID de campagne (CampaignId), une chaîne qui identifie de façon unique une campagne.</p>
Endpoint deliveries, grouped by date (Remises au point de terminaison, regroupement par date)	unique-deliveries-grouped-by-date	<p>Pour toutes les campagnes associées à un projet, nombre de points de terminais on uniques auxquels les messages ont été envoyés, pour chaque jour de la plage de dates spécifiée.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métrique	Kpi-name	Description
Messages delivered, grouped by campaign (Messages remis, regroupement par campagne)	successful-deliveries-grouped-by-campaign	<p>Pour chaque campagne associée à un projet, nombre de messages ayant été remis aux destinataires.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID de campagne (<code>CampaignId</code>), une chaîne qui identifie de façon unique une campagne.</p>

Métrique	Kpi-name	Description
Push open rate (Taux d'ouverture des notifications push)	push-open-rate	<p>Pour toutes les campagnes associées à un projet, pourcentage de notifications push ayant été ouvertes par les destinataires.</p> <p>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.</p>

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	<p>Pour chaque campagne associée à un projet, pourcentage de notifications push ayant été ouvertes par les destinataires.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID de campagne (CampaignId), une chaîne qui identifie de façon unique une campagne.</p>

Métriques d'application 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	<code>txn-emails-clicked</code>	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	<code>txn-emails-clicked-grouped-by-date</code>	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>
Complaint rate (Taux de réclamations)	<code>txn-emails-complaint-rate</code>	<p>Pourcentage de messages signalés par les destinataires comme des e-mails non souhaités ou indésirables.</p> <p>Cette métrique est calculée en fonction du nombre de messages signalés par les</p>

Métrique	Kpi-name	Description
		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	<p>Pourcentage de messages signalés par les destinataires comme des e-mails non souhaités ou indésirables, pour chaque jour de la plage de dates spécifiée.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>
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	<code>txn-emails-with-complaints-grouped-by-date</code>	<p>Nombre de messages signalés par les destinataires comme des e-mails non souhaités ou indésirables, pour chaque jour de la plage de dates spécifiée.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>
Messages délivrés	<code>txn-emails-delivered</code>	<p>Nombre de messages remis à leurs destinataires prévus.</p> <p>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.</p>

Métrique	Kpi-name	Description
Livraisons, regroupées par date	txn-emails-delivered-grouped-by-date	<p>Nombre de messages remis aux destinataires, pour chaque jour de la plage de dates spécifiée.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métrique	Kpi-name	Description
Delivery rate (Taux de remise)	txn-emails-delivery-rate	<p>Pourcentage de messages remis aux destinataires.</p> <p>Cette métrique est calculée en fonction du nombre de messages envoyés et remis aux destinataires, divisé par le nombre de messages envoyés.</p>
Delivery rate, grouped by date (Taux de remise, regroupement par date)	txn-emails-delivery-rate-grouped-by-date	<p>Pourcentage de messages remis aux destinataires, pour chaque jour de la plage de dates spécifiée.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

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	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>
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	<p>Nombre de messages ouverts par les destinataires, pour chaque jour de la plage de dates spécifiée.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>
Envois	txn-emails-sent	Nombre de messages envoyés.
Envois, regroupés par date	txn-emails-sent-grouped-by-date	<p>Nombre de messages envoyés, pour chaque jour de la plage de dates spécifiée.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>
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	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métrique	Kpi-name	Description
Événements de clics utilisateur uniques	txn-emails-unique-clicks	<p>Nombre de destinataires uniques (points de terminaison) ayant cliqué sur des liens dans des messages.</p> <p>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.</p>

Métrique	Kpi-name	Description
Événements de clics utilisateur uniques, regroupés par date	txn-emails-unique-clicks-grouped-by-date	<p>Nombre de destinataires uniques (points de terminaison) ayant cliqué sur des liens dans des messages, pour chaque jour de la plage de dates spécifiée.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métrique	Kpi-name	Description
Événements d'ouverture utilisateur uniques	txn-emails-unique-opens	<p>Nombre de destinataires uniques (points de terminaison) ayant ouvert des messages.</p> <p>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.</p>

Métrique	Kpi-name	Description
Événements d'ouverture utilisateur uniques, regroupés par date	txn-emails-unique-opens-grouped-by-date	<p>Nombre de destinataires uniques (points de terminaison) ayant ouvert des messages, pour chaque jour de la plage de dates spécifiée.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métriques d'application 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	<p>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 ($645 / 1000 = 0,645 \text{ ¢} = 0,00645 \text{ \\$}$).</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.</p>
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
		<p>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 ($645 / 1000 = 0,645 \text{ ¢} = 0,00645 \text{ \\$}$).</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.</p>
Messages délivrés	txn-sms-delivered	Nombre de messages remis à leurs destinataires prévus.
Livraisons, regroupées par pays	txn-sms-delivered-grouped-by-country	<p>Nombre de messages remis aux destinataires, pour chaque pays ou région où les messages ont été envoyés.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.</p>

Métrique	Kpi-name	Description
Livraisons, regroupées par date	<code>txn-sms-delivered-grouped-by-date</code>	<p>Nombre de messages remis aux destinataires, pour chaque jour de la plage de dates spécifiée.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>
Erreurs de livraison	<code>txn-sms-error-distribution</code>	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par code d'erreur, pour chaque type d'erreur qui s'est produite.</p>
Delivery rate (Taux de remise)	<code>txn-sms-delivery-rate</code>	<p>Pourcentage de messages remis aux destinataires.</p> <p>Cette métrique est calculée en fonction du nombre de messages envoyés et remis aux destinataires, divisé par le nombre de messages envoyés.</p>

Métrique	Kpi-name	Description
Delivery rate, grouped by date (Taux de remise, regroupement par date)	txn-sms-delivery-rate-grouped-by-date	<p>Pourcentage de messages remis aux destinataires, pour chaque jour de la plage de dates spécifiée.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>
Parties de message remises	txn-sms-delivered-by-parts	<p>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.</p>

Métrique	Kpi-name	Description
Parties de message remises, regroupées par pays	<code>txn-sms-delivered-by-parts-grouped-by-country</code>	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.</p>
Parties de message envoyées	<code>txn-sms-sent-by-parts</code>	<p>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.</p>

Métrique	Kpi-name	Description
Parties de message envoyées, regroupées par pays	txn-sms-sent-by-parts-grouped-by-country	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.</p>
Messages sent (Messages envoyés)	txn-sms-sent	Nombre de messages envoyés.
Messages envoyés, regroupés par pays	txn-sms-sent-grouped-by-country	<p>Nombre de messages envoyés, pour chaque pays ou région où les messages ont été envoyés.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.</p>
Messages envoyés, regroupés par date	txn-sms-sent-grouped-by-date	<p>Nombre de messages envoyés, pour chaque jour de la plage de dates spécifiée.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métrique	Kpi-name	Description
Prix total, regroupé par pays	txn-sms-total-price-grouped-by-country	<p>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 ($645 / 1000 = 0,645 \text{ ¢} = 0,00645 \text{ \\$}$).</p> <p>Les résultats de la requête pour cette métrique sont regroupés par pays ou région, au format ISO 3166-1 alpha-2.</p>

Métriques de campagne

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
		<p>les messages dans lesquels l'adresse e-mail du destinataire a rencontré un problème permanent ayant empêché la remise du message.</p> <p>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.</p>

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	<p>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.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (<code>CampaignActivityId</code>), une chaîne qui identifie de façon unique une campagne.</p>

Métrique	Kpi-name	Description
Delivery rate (Taux de remise)	successful-delivery-rate	<p>Pour l'ensemble des exécutions de campagnes, pourcentage de messages ayant été remis aux destinataires.</p> <p>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</p> .

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	<p>Pour chaque exécution de campagne, pourcentage de messages ayant été remis aux destinataires.</p> <p>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.</p> <p>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.</p>

Métrique	Kpi-name	Description
Delivery rate, grouped by date (Taux de remise, regroupement par date)	successful-delivery-rate-grouped-by-date	<p>Pour l'ensemble des exécutions de campagnes, pourcentage de messages ayant été remis aux destinataires, pour chaque jour de la plage de dates spécifiée.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métrique	Kpi-name	Description
Email open rate (Taux d'ouverture d'e-mail)	<code>email-open-rate</code>	<p>Pour l'ensemble des exécutions de campagnes, pourcentage d'e-mails ayant été ouverts par les destinataires.</p> <p>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.</p>

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	<p>Pour chaque exécution de campagne, pourcentage d'e-mails ayant été ouverts par les destinataires.</p> <p>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.</p> <p>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.</p>
Emails opened, grouped by campaign run (E-mails ouverts, regroupement par exécution de campagne)	direct-email-opens-grouped-by-campaign-activity	<p>Pour chaque exécution de campagne, nombre d'e-mails ayant été ouverts par les destinataires.</p> <p>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.</p>

Métrique	Kpi-name	Description
Endpoint deliveries (Remises au point de terminaison)	<code>unique-deliveries</code>	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)	<code>unique-deliveries-grouped-by-campaign-activity</code>	<p>Pour chaque exécution de campagne, nombre de points de terminaison uniques auxquels les messages ont été envoyés.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (<code>CampaignActivityId</code>), une chaîne qui identifie de façon unique une campagne.</p>
Endpoint deliveries, grouped by date (Remises au point de terminaison, regroupement par date)	<code>unique-deliveries-grouped-by-date</code>	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par jour calendaire, au format étendu ISO 8601.</p>

Métrique	Kpi-name	Description
Links clicked, grouped by campaign run (Liens activés, regroupement par exécution de campagne)	<code>clicks-grouped-by-campaign-activity</code>	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité de campagne (<code>CampaignActivityId</code>), une chaîne qui identifie de façon unique une campagne.</p>

Métrique	Kpi-name	Description
Messages delivered, grouped by campaign run (Messages remis, regroupement par exécution de campagne)	successful-deliveries-grouped-by-campaign-activity	<p>Pour chaque exécution de campagne, nombre de messages ayant été remis aux destinataires.</p> <p>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.</p> <p>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.</p>
Messages sent, grouped by campagne run (Messages envoyés, regroupement par exécution de campagne)	attempted-deliveries-grouped-by-campaign-activity	<p>Pour chaque exécution de campagne, nombre de messages ayant été envoyés.</p> <p>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.</p>

Métrique	Kpi-name	Description
Push open rate (Taux d'ouverture des notifications push)	push-open-rate	<p>Pour l'ensemble des exécutions de campagnes, pourcentage de notifications push ayant été ouvertes par les destinataires.</p> <p>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.</p>

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	<p>Pour chaque exécution de campagne, pourcentage de notifications push ayant été ouvertes par les destinataires.</p> <p>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.</p> <p>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.</p>
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	<p>Pour chaque exécution de campagne, nombre de notifications push ayant été ouvertes par les destinataires.</p> <p>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.</p>

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.

Métriques d'engagement du parcours

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
		<p>chaque clic est inclus dans le total.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (<code>JourneyActivityId</code>), une chaîne qui identifie de façon unique une activité.</p>
Réclamations	<code>journey-emails-complained</code>	Nombre de messages signalés par les participants comme des e-mails indésirables.
Réclamations, regroupées par activité	<code>emails-complained-grouped-by-journey-activity</code>	<p>Pour chaque activité du parcours, le nombre de messages signalés par les participants comme des e-mails indésirables.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (<code>JourneyActivityId</code>), une chaîne qui identifie de façon unique une activité.</p>

Métrique	Kpi-name	Description
Messages délivrés	<code>journey-emails-delivered</code>	<p>Nombre de messages remis aux participants.</p> <p>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.</p>
Messages remis, regroupés par activité	<code>emails-delivered-grouped-by-journey-activity</code>	<p>Pour chaque activité du parcours, le nombre de messages ayant été remis aux participants.</p> <p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (<code>JourneyActivityId</code>), une chaîne qui identifie de façon unique une activité.</p>

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	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (<code>JourneyActivityId</code>), une chaîne qui identifie de façon unique une activité.</p>
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	<p>Pour chaque activité du parcours, le nombre de messages ayant été ouverts par les participants.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (<code>JourneyActivityId</code>), une chaîne qui identifie de façon unique une activité.</p>
Rejets	journey-emails-rejected	<p>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.</p>

Métrique	Kpi-name	Description
Messages rejetés, regroupés par activité	emails-rejected-grouped-by-journey-activity	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (<code>JourneyActivityId</code>), une chaîne qui identifie de façon unique une activité.</p>
Envois	journey-emails-sent	Nombre de messages envoyés.
Messages envoyés, regroupés par activité	emails-sent-grouped-by-journey-activity	<p>Pour chaque activité du parcours, le nombre de messages envoyés.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (<code>JourneyActivityId</code>), une chaîne qui identifie de façon unique une activité.</p>

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	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (JourneyActivityId), une chaîne qui identifie de façon unique une activité.</p>
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	<p>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.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par ID d'activité (JourneyActivityId), une chaîne qui identifie de façon unique une activité.</p>

Métriques d'exécution du parcours

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	<p>Nombre de participants poursuivant activement les activités du parcours.</p> <p>Cette métrique est calculée de la façon suivante : nombre de participants qui ont commencé le parcours, moins nombre</p>

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	<p>Nombre de participants qui ne peuvent pas commencer le parcours parce qu'ils sont déjà des participants actifs au parcours.</p> <p>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).</p>

Métriques d'exécution de l'activité du parcours

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 split (Répartition Oui/Non) (CONDITIONAL_SPLIT)	<p>Les métriques sont les suivantes :</p> <ul style="list-style-type: none"> 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
	<ul style="list-style-type: none">• <code>Branch_TRUE</code> : nombre de participants qui répondaient aux conditions de l'activité et qui se sont dirigés vers l'activité sur le chemin « Oui ». <p>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.</p>
Holdout (Exclusion) (HOLDOUT)	<p>Les métriques sont les suivantes :</p> <ul style="list-style-type: none">• <code>HOLDOUT</code> : nombre de participants ayant été exclus du parcours dans le cadre du pourcentage d'exclusion pour l'activité.• <code>PASSED</code> : nombre de participants qui sont passés à l'activité suivante du parcours.

Type d'activité	Métriques
Email (E-mail) (MESSAGE)	<p>Les métriques sont les suivantes :</p> <ul style="list-style-type: none">• DAILY_CAP_EXCEEDED : nombre de messages n'ayant pas été envoyés parce qu'ils auraient dépassé le nombre maximum de messages qu'un seul participant 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)	<p>Pour chaque chemin de l'activité, le nombre de participants qui se sont dirigés vers l'activité sur le chemin.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par chemin, <code>Branch_#</code>, où <code>#</code> est l'identifiant numérique d'un chemin : par exemple, <code>Branch_1</code> pour le premier chemin de l'activité.</p> <p>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.</p>
Random split (Répartition aléatoire) (RANDOM_SPLIT)	<p>Pour chaque chemin de l'activité, le nombre de participants qui se sont dirigés vers l'activité sur le chemin.</p> <p>Les résultats de la requête pour cette métrique sont regroupés par chemin, <code>Branch_#</code>, où <code>#</code> est l'identifiant numérique d'un chemin : par exemple, <code>Branch_1</code> pour le premier chemin de l'activité.</p> <p>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.</p>

Type d'activité	Métriques
Wait (Attente) (WAIT)	<p>Les métriques sont les suivantes :</p> <ul style="list-style-type: none">• <code>WAIT_FINISHED</code> : nombre de participants ayant fini d'attendre le temps spécifié.• <code>WAIT_SKIPPED</code> : 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.• <code>WAIT_STARTED</code> : 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)	<p>Les métriques sont les suivantes :</p> <ul style="list-style-type: none">• CALL_QUEUED : nombre de participants qui ont été appelés et mis en file d'attente pour accéder à Amazon Connect. 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 : nombre de participants qui n'ont reçu aucun code de disposition Amazon Connect 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.

Métriques relatives à l'exécution des parcours et des campagnes

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 terminaison 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.
ENDPOINT_MISSING_ADDRESS	les deux	Il manquait une adresse au point de terminaison et il n'a

Nom de la métrique	S'applique aux parcours, aux campagnes ou aux deux	Description
		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	Ceci est émis dans les campagnes A/B, pour les critères dont les cohortes ne correspondent pas au traitement actuel. Par exemple, dans une répartition A/B 50/50, 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.

Interrogation des données analytiques Amazon Pinpoint pour les campagnes

Outre l'utilisation des pages d'analyse de la console Amazon Pinpoint, vous pouvez utiliser les API analytiques Amazon Pinpoint pour interroger les données analytiques d'un sous-ensemble de métriques standard qui fournissent les informations sur les tendances de diffusion et d'engagement des 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 kit SDK AWS Mobile, Amazon Pinpoint étend cette prise en charge pour inclure des métriques 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 [Intégration de votre application à Amazon Pinpoint](#).

Si vous utilisez les API analytiques Amazon Pinpoint pour interroger les 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.

Prérequis

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 [Métriques standard](#).

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](#).

Interrogation de données 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 [Métriques standard](#).

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 analytiques pour une campagne à l'aide de l'API REST Amazon Pinpoint, de AWS CLI et de AWS SDK for Java. Vous pouvez utiliser n'importe quel kit SDK AWS pris en charge pour interroger les données d'analyse d'une campagne. Les exemples AWS CLI sont mis en forme 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 de la région AWS 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 valeur *kpi-name* 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/1234567890123456789012345example/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.
- `1234567890123456789012345example` 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 l'AWS CLI, utilisez la commande `get-campaign-date-range-kpi` 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 valeur `kpi-name` 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 1234567890123456789012345example ^
  --campaign-id 80b8efd84042ff8d9c96ce2f8example ^
  --kpi-name unique-deliveries ^
  --start-time 2019-07-19T00:00:00Z ^
  --end-time 2019-07-26T23:59:59Z
```

Où :

- `1234567890123456789012345example` 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 l'AWS SDK for Java, utilisez la `GetCampaignDateRangeKpiRequest` méthode de l'API [métriques de campagne](#). 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 valeur `kpi-name` 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("1234567890123456789012345example")
    .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 :

- 1234567890123456789012345example 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": "1234567890123456789012345example",
    "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 [Utilisation des résultats des requêtes](#).

Interrogation de données 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 [Métriques standard](#).

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 analytiques pour une campagne à l'aide de l'API REST Amazon Pinpoint, de AWS CLI et de AWS SDK for Java. Vous pouvez utiliser n'importe quel kit SDK AWS pris en charge pour interroger les données d'analyse d'une campagne. Les exemples AWS CLI sont mis en forme 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 de la région AWS qui héberge le projet associé aux campagnes.
- *application-id* est l'identifiant unique du projet associé aux campagnes.
- *kpi-name* est la valeur `kpi-name` 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/1234567890123456789012345example/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.
- `1234567890123456789012345example` 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 l'AWS CLI, utilisez la commande `get-application-date-range-kpi` 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 valeur `kpi-name` 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 1234567890123456789012345example ^
  --kpi-name unique-deliveries-grouped-by-campaign ^
  --start-time 2019-07-19T00:00:00Z ^
  --end-time 2019-07-26T23:59:59Z
```

Où :

- `1234567890123456789012345example` 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 l'AWS SDK for Java, utilisez la `GetApplicationDateRangeKpiRequest` méthode de l'API [métriques d'application](#). 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 à la campagne.
- *kpiName* est la valeur kpi-name 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("1234567890123456789012345example")  
    .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 :

- `1234567890123456789012345example` 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":"1234567890123456789012345example",
    "EndTime":"2019-07-26T23:59:59Z",
    "KpiName":"unique-deliveries-grouped-by-campaign",
    "KpiResult":{
      "Rows":[
        {
          "GroupedBy":[
            {
              "Key":"CampaignId",
              "Type":"String",
              "Value":"80b8efd84042ff8d9c96ce2f8example"
            }
          ],
          "Values":[
            {
              "Key":"UniqueDeliveries",
              "Type":"Double",
              "Value":"123.0"
            }
          ]
        },
        {
          "GroupedBy":[
            {
              "Key":"CampaignId",
              "Type":"String",
              "Value":"810c7aab86d42fb2b56c8c966example"
            }
          ],
          "Values":[
            {
              "Key":"UniqueDeliveries",
              "Type":"Double",
              "Value":"456.0"
            }
          ]
        }
      ]
    }
  },
}
```

```
{
  "GroupedBy": [
    {
      "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 `GroupedBy` 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 [Utilisation des résultats des requêtes](#).

Interrogation des données analytiques Amazon Pinpoint pour les messages transactionnels

Outre l'utilisation des pages analytiques de la console Amazon Pinpoint, vous pouvez utiliser les API analytiques Amazon Pinpoint pour interroger les données analytiques d'un sous-ensemble de métriques standard qui fournissent des informations sur les tendances de remise et d'engagement des messages transactionnels envoyés pour 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 les API analytiques Amazon Pinpoint pour interroger les 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.

Prérequis

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 [Métriques standard](#).

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](#).

Interrogation de données pour les messages électroniques 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 [Métriques standard](#).

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 les données analytiques pour les e-mails transactionnels à l'aide de l'API REST Amazon Pinpoint, de l'AWS CLI ou de AWS SDK for Java. Vous pouvez utiliser n'importe quel kit SDK AWS pris en charge pour interroger les données d'analyse pour les messages transactionnels. Les exemples AWS CLI sont mis en forme 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 de la région AWS qui héberge le projet.
- *application-id* est l'identifiant unique du projet.
- *kpi-name* est la valeur *kpi-name* 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/1234567890123456789012345example/  
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.
- *1234567890123456789012345example* 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 messages électroniques transactionnels à l'aide de l'AWS CLI, utilisez la commande `get-application-date-range-kpi` 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 valeur `kpi-name` 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 1234567890123456789012345example ^  
  --kpi-name txn-emails-sent ^  
  --start-time 2019-09-06T00:00:00Z ^  
  --end-time 2019-09-13T23:59:59Z
```

Où :

- 1234567890123456789012345example 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 messages électroniques transactionnels à l'aide de l'AWS SDK for Java, utilisez la `GetApplicationDateRangeKpiRequest` méthode de l'API [métriques d'application](#). 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 valeur `kpi-name` 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("1234567890123456789012345example")
    .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 :

- 1234567890123456789012345example 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":"1234567890123456789012345example",
    "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 [Utilisation des résultats des requêtes](#).

Interrogation de données pour les messages 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 [Métriques standard](#).

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 analytiques pour les messages SMS transactionnels à l'aide de l'API REST Amazon Pinpoint, de l'AWS CLI ou de AWS SDK for Java. Vous pouvez utiliser n'importe quel kit SDK AWS pris en charge pour interroger les données d'analyse pour les messages transactionnels. Les exemples AWS CLI sont mis en forme 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 de la région AWS qui héberge le projet.
- *application-id* est l'identifiant unique du projet.
- *kpi-name* est la valeur kpi-name 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/1234567890123456789012345example/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.
- `1234567890123456789012345example` 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 l'AWS CLI, utilisez la commande `get-application-date-range-kpi` 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 valeur `kpi-name` 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 1234567890123456789012345example ^  
  --kpi-name txn-sms-sent-grouped-by-date ^  
  --start-time 2019-09-06T00:00:00Z ^  
  --end-time 2019-09-08T23:59:59Z
```

Où :

- 1234567890123456789012345example 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 du AWS SDK for Java, utilisez la méthode `GetApplicationDateRangeKpiRequest` de l'API des [métriques d'application](#) 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 valeur `kpi-name` 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("1234567890123456789012345example")
    .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 :

- 1234567890123456789012345example 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":"1234567890123456789012345example",
    "EndTime":"2019-09-08T23:59:59Z",
    "KpiName":"txn-sms-sent-grouped-by-date",
    "KpiResult":{
      "Rows":[
        {
          "GroupedBy": [
            {
              "Key":"Date",
              "Type":"String",
              "Value":"2019-09-06"
            }
          ],
          "Values": [
            {
              "Key":"TxnSmsSent",
```

```
        "Type": "Double",
        "Value": "29.0"
      }
    ],
  },
  {
    "GroupedBy": [
      {
        "Key": "Date",
        "Type": "String",
        "Value": "2019-09-07"
      }
    ],
    "Values": [
      {
        "Key": "TxnSmsSent",
        "Type": "Double",
        "Value": "35.0"
      }
    ]
  },
  {
    "GroupedBy": [
      {
        "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 `GroupedBy` 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 [Utilisation des résultats des requêtes](#).

Utilisation des résultats des requêtes analytiques Amazon Pinpoint

Lorsque vous utilisez les API analytiques Amazon Pinpoint pour interroger des données analytiques, 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 :

- 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 que les API analytiques Amazon Pinpoint ne sont pas conçues pour créer ou stocker des objets persistants que vous pouvez lire ou utiliser par la suite dans un projet Amazon Pinpoint ou votre compte Amazon Pinpoint. Au lieu de cela, les API sont conçues pour vous aider à récupérer les données d'analyse et à transférer ces données vers d'autres services et applications pour

des analyses, un stockage ou des rapports plus poussés. 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 [Métriques analytiques Amazon Pinpoint standard](#).

Structure JSON

Pour vous aider à analyser et à utiliser les résultats de la requête, les API analytiques Amazon Pinpoint utilisent la même structure de réponse JSON pour toutes les données analytiques Amazon Pinpoint que vous pouvez interroger par programmation pour les métriques d'application, de campagne et d'engagement du 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 `GroupedBy` associé. L'objet `GroupedBy` 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 [Métriques analytiques Amazon Pinpoint standard](#).

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": "1234567890123456789012345example",
    "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": "2345678901234567890123456example",
    "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":"1234567890123456789012345example",
    "EndTime":"2019-08-31T23:59:59Z",
    "KpiName":"unique-deliveries-grouped-by-campaign",
    "KpiResult":{
      "Rows":[
        {
          "GroupedBy":[
            {
              "Key":"CampaignId",
              "Type":"String",
              "Value":"80b8efd84042ff8d9c96ce2f8example"
            }
          ],
          "Values":[
            {
              "Key":"UniqueDeliveries",
              "Type":"Double",
              "Value":"123.0"
            }
          ]
        },
        {
          "GroupedBy":[
            {
              "Key":"CampaignId",
              "Type":"String",
              "Value":"810c7aab86d42fb2b56c8c966example"
            }
          ],
          "Values":[
```

```
        {
            "Key": "UniqueDeliveries",
            "Type": "Double",
            "Value": "456.0"
        }
    ],
},
{
    "GroupedBy": [
        {
            "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 :

- `GroupedBy.Key` : nom de la propriété ou du champ qui stocke la valeur de regroupement spécifiée dans le champ `GroupedBy.Value` (`CampaignId`).
- `GroupedBy.Type` : type de données de la valeur spécifiée dans le champ `GroupedBy.Value` (`String`).
- `GroupedBy.Value` : valeur réelle du champ utilisé pour regrouper les données, comme spécifié dans le champ `GroupedBy.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
<code>Rows</code>	Tableau d'objets contenant le jeu de résultats d'une requête.
<code>Rows.GroupedBy</code>	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
<code>Rows.GroupedBy.Key</code>	Pour une métrique à valeurs multiples, nom de la propriété ou du champ qui stocke la valeur spécifiée dans le champ <code>GroupedBy.Value</code> .
<code>Rows.GroupedBy.Type</code>	Pour une métrique à valeurs multiples, type de données de la valeur spécifiée dans le champ <code>GroupedBy.Value</code> .
<code>Rows.GroupedBy.Value</code>	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 <code>Values</code> associé.
<code>Rows.Values</code>	Tableau de champs contenant les résultats de la requête.
<code>Rows.Values.Key</code>	Le nom convivial de la mesure interrogée. La valeur de la métrique est spécifiée dans le champ <code>Values.Value</code> .
<code>Rows.Values.Type</code>	Type de données de la valeur spécifiée dans le champ <code>Values.Value</code> .
<code>Rows.Values.Value</code>	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 [Métriques analytiques Amazon Pinpoint standard](#).

Journalisation des 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 AWS Identity and Access Management 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 créer 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 les [sections Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [Réception de fichiers CloudTrail journaux de plusieurs comptes](#).

Vous pouvez l'utiliser CloudTrail pour consigner les actions pour les API Amazon Pinpoint suivantes :

- [API Amazon Pinpoint](#)
- [API SMS et voix Amazon Pinpoint](#)

Actions de l'API Amazon Pinpoint qui peuvent être enregistrées par 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](#)
- [PutEventStream](#)

- [RemoveAttributes](#)
- [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](#)

Les actions suivantes de l'API Amazon Pinpoint ne sont pas connectées : CloudTrail

- PutEvents

- `SendMessage`
- `SendUsersMessages`

Actions de l'API de messagerie Amazon Pinpoint qui peuvent être enregistrées par 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 SMS et voix Amazon Pinpoint qui peuvent être enregistrées par 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

Exemples : entrées de fichier journal 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": "AIDAIHTRCDA62EXAMPLE",
    "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": "AIDAIHTRCDA62EXAMPLE",
        "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":"AIDAIHTRCDA62EXAMPLE",
      "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"  
  }  
]  
}
```

Balisage des ressources Amazon Pinpoint

Une balise est une étiquette que vous pouvez éventuellement définir et associer à des ressources AWS, y compris 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.

Gestion des 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 valeurs de balise sont sensibles à la casse.
- Pour chaque ressource associée, chaque clé de balise doit être unique et ne peut avoir qu'une seule valeur.
- Le préfixe `aws:` est réservé pour être utilisé par AWS ; vous ne pouvez pas vous en servir dans les clés ou les valeurs de balise 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 sont disponibles uniquement pour votre compte AWS, pas pour les autres comptes qui partagent la ressource. De plus, les balises ne sont disponibles que pour les ressources qui sont situées dans la région AWS spécifiée pour votre compte AWS.

Pour ajouter, afficher, mettre à jour et supprimer les clés et les valeurs de balise de ressources Amazon Pinpoint, vous pouvez utiliser l'AWS Command Line Interface (AWS CLI), l'API Amazon Pinpoint, l'API Tagging AWS Resource Groups ou le kit SDK AWS. Pour gérer les clés et les valeurs de balise pour toutes les ressources AWS qui sont situées dans une région AWS donnée pour votre compte AWS, y compris les ressources Amazon Pinpoint, utilisez [l'API de balisage AWS Resource Groups](#).

Utilisation de balises dans les politiques IAM

Après avoir commencé à implémenter des balises, vous pouvez appliquer des autorisations de niveau ressource basées sur des balises aux politiques AWS Identity and Access Management (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. Une telle utilisation des balises vous aide à mettre en place un contrôle précis sur les groupes et les utilisateurs de votre compte AWS qui ont le droit de créer et de baliser des ressources, et sur les groupes et les utilisateurs sont autorisés à créer, à mettre à jour et à supprimer des balises plus généralement.

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 :

```
{
  "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.

Ajout de balises à des ressources

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 kit SDK AWS pris en charge 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 de groupe de ressources de l'AWS CLI ou du [Balisage d'API AWS Resource Groups](#).

Ajout de 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"
  }
}
```

Ajout de balises à l'aide de AWS CLI

Pour créer une nouvelle ressource et y ajouter une balise à l'aide de l'AWS CLI, utilisez la commande `create` correspondant à 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 :

- Remplacez *myProject* par le nom que vous souhaitez donner 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 :

- Remplacez *resource-arn* par l'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.

Affichage des balises pour des ressources

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 ressource Amazon Pinpoint. Vous pouvez également utiliser n'importe quel kit SDK AWS pris en charge pour afficher les balises associées à une ressource.

Afficher les balises 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 toutes les balises associées 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 récupère 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 de AWS CLI

Pour utiliser AWS CLI pour afficher la liste des balises associées à une ressource spécifique, exécutez la commande `list-tags-for-resource` et spécifiez l'Amazon Resource Name (ARN) de la ressource pour le paramètre `resource-arn`, 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 une liste de toutes les ressources Amazon Pinpoint qui ont des balises, ainsi que de toutes les balises associées à chacune de ces ressources, utilisez la commande [get-resources](#) de l'API AWS Resource Groups Tagging. 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"
```

La sortie de la commande est une liste d'ARN pour toutes les ressources Amazon Pinpoint qui ont des balises. La liste inclut toutes les clés et les valeurs de balise qui sont associées à chaque ressource.

Mise à jour des balises pour des ressources

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 plusieurs ressources en même temps, utilisez les opérations de balisage de groupes de ressources de l'AWS CLI ou du [Balisage d'API AWS Resource Groups](#). 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.

Suppression de balises de ressources

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 kit SDK AWS pris en charge 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 de groupes de ressources de l'AWS CLI ou du [Balisage d'API AWS Resource Groups](#). Pour supprimer uniquement une valeur de balise spécifique, pas une clé de balise, d'une ressource, [mettez à jour les balises de la ressource](#).

Suppression de balises 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 de la région AWS qui héberge la ressource.
- *resource-arn* correspond à l'ARN de la ressource de laquelle vous souhaitez supprimer une balise.
- *key* est la clé de balise 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.

Suppression de balises à l'aide de AWS CLI

Pour supprimer une balise d'une ressource à l'aide de l'AWS CLI, exécutez la commande `untag-resource`. 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 :

- Remplacez *resource-arn* par l'ARN de la ressource de laquelle vous souhaitez supprimer les balises.

- Remplacez *key1* et *key2* par les clés des balises que vous souhaitez supprimer de la ressource.

Informations connexes

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](#).

Personnalisation des recommandations 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 régions AWS suivantes : USA Est (Virginie du Nord), USA Ouest (Oregon), Asie Pacifique (Mumbai), Asie Pacifique (Sydney) et Europe (Irlande).

Rubriques

- [Utilisation des recommandations dans les messages](#)
- [Création de la fonction Lambda](#)
- [Attribution d'une stratégie de fonction Lambda](#)
- [Autorisation accordée à Amazon Pinpoint d'invoquer la fonction](#)
- [Configuration du modèle de recommandation](#)

Utilisation des recommandations dans les messages

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. Il s'agit notamment

d'appeler ou non une fonction AWS Lambda pour effectuer un traitement supplémentaire des données récupérées.

Amazon Personalize est un service AWS 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 qui permet d'exécuter du code sans avoir à mettre en service ni à gérer des serveurs. Vous empaquetez votre code et le téléchargez dans AWS Lambda en tant que fonction Lambda. AWS Lambda exécute alors la fonction lorsque la fonction est appelée. 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. Utilise l'opération [GetRecommendations](#) de l'API d'exécution Amazon Personalize 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 de la fonction Lambda

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 comprend un objet `Endpoints`, qui est une carte qui associe les ID de point de terminaison aux définitions de point de terminaison pour les destinataires de message.

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":"alejandrор@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 région AWS que le projet Amazon Pinpoint qui invoque 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 d'événement au gestionnaire en tant que paramètre `event`. 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":"alejandr@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`.

Attribution d'une stratégie de fonction Lambda

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'invoquer, 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 stratégie suivant permet au principal du service Amazon Pinpoint d'utiliser l'action `lambda:InvokeFunction` pour une campagne Amazon Pinpoint particulière (*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 écrire une stratégie qui permet au principal du service Amazon Pinpoint d'utiliser l'action `lambda:InvokeFunction` 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.

Autorisation accordée à Amazon Pinpoint d'invoquer la fonction

Après avoir attribué une stratégie de fonction Lambda à la fonction, vous pouvez ajouter des autorisations qui autorisent Amazon Pinpoint à invoquer la fonction pour un projet, une campagne ou un parcours spécifique. Vous pouvez le faire à l'aide de AWS Command Line Interface (AWS CLI) et de la commande [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.

Configuration du 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.

Suppression de données d'Amazon Pinpoint

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.

Suppression de points de terminaison

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.

Pour supprimer toutes les données d'un point de terminaison spécifique, vous pouvez utiliser l'API Amazon Pinpoint pour supprimer le point de terminaison. La procédure suivante montre comment supprimer un point de terminaison en utilisant AWS CLI pour interagir avec l'API Amazon Pinpoint. Cette procédure suppose que vous avez déjà installé et configuré l'AWS CLI. Pour plus d'informations, consultez [Installation d'AWS CLI](#) dans le Guide de l'utilisateur AWS Command Line Interface .

Pour supprimer un point de terminaison à l'aide du AWS CLI

- Sur la ligne de commande, entrez la commande suivante :

```
aws pinpoint delete-endpoint --application-id 810c7aab86d42fb2b56c8c966example --  
endpoint-id ad015a3bf4f1b2b0b82example
```

Dans la commande précédente, remplacez *810c7aab86d42fb2b56c8c966example* par l'ID du projet auquel le point de terminaison est associé. Remplacez également *ad015a3bf4f1b2b0b82example* par l'ID unique du point de terminaison lui-même.

Pour rechercher l'ID de point de terminaison d'un point de terminaison spécifique, déterminez à quel segment appartient le point de terminaison, puis exportez le segment à partir d'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 savoir comment procéder, consultez [Exporter des segments](#) dans le Guide de l'utilisateur Amazon Pinpoint. Vous pouvez également exporter un segment vers un compartiment Amazon Simple Storage Service (Amazon S3) à l'aide de l'API Amazon Pinpoint. Pour en savoir plus, consultez [Exportation de points de terminaison](#) dans ce guide.

Suppression de données de segment et de point de terminaison stockées dans Amazon S3

Vous pouvez importer des segments à partir d'un fichier qui est stocké dans un compartiment Amazon S3 à l'aide de la console Amazon Pinpoint ou de l'API. Vous pouvez également exporter des données d'application, de segment ou de point de terminaison à partir d'Amazon Pinpoint vers un compartiment Amazon S3. Les fichiers importés et exportés peuvent contenir des données personnelles, notamment des adresses électroniques, des numéros de téléphone mobile et des informations sur l'emplacement physique du point de terminaison. Vous pouvez supprimer ces fichiers d'Amazon S3.

Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez [How Do I Empty an S3 Bucket?](#) (Comment puis-je vider un compartiment S3 ?) ou [How Do I Delete an S3 Bucket?](#) (Comment supprimer un compartiment S3 ?).

Suppression de toutes les données du projet

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 ID utilisateur et les autres données que vous avez importées, ainsi que tous 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.

Supprimer toutes les AWS données en fermant votre AWS compte

Il est également possible de supprimer toutes les données que vous avez stockées dans Amazon Pinpoint en fermant votre compte AWS . Toutefois, cette action supprime également toutes les autres données, personnelles ou non, que vous avez stockées dans tous les autres services. AWS Après la période post-fermeture, ferme AWS définitivement votre AWS compte et vous ne pouvez plus le rouvrir. Tout contenu que vous n'avez pas supprimé est définitivement supprimé, et tous les AWS services que vous n'avez pas interrompus sont arrêtés. Pour plus d'informations, voir [Fermer un AWS compte](#) dans le Guide AWS Account Management de référence.


 Warning

La procédure suivante supprime complètement toutes les données stockées dans votre AWS compte, quels que soient les AWS services et les AWS régions.

Vous pouvez fermer votre AWS compte en utilisant le AWS Management Console.

Pour fermer votre AWS compte

1. Ouvrez le AWS Management Console à l'[adresse https://console.aws.amazon.com](https://console.aws.amazon.com).
2. Accédez à la page Account Settings (Paramètres de compte) à l'[adresse https://console.aws.amazon.com/billing/home?#/account](https://console.aws.amazon.com/billing/home?#/account).

 Warning

Les étapes suivantes suppriment définitivement toutes les données que vous avez stockées dans tous les AWS services dans toutes les AWS régions.

3. Dans la section Clôture du compte, lisez la clause de non-responsabilité qui décrit les conséquences de la fermeture de votre AWS compte. Si vous acceptez ces conditions, cochez la case, puis choisissez Close Account (Clôturer le compte).
4. Dans la boîte de dialogue de confirmation, sélectionnez Close Account (Clôturer le compte).

Exemples de code pour Amazon Pinpoint utilisant des kits SDK AWS

Les exemples de code suivants montrent comment utiliser Amazon Pinpoint avec un kit de développement logiciel AWS (SDK).

Pour obtenir la liste complète des guides de développement AWS SDK et des exemples de code, consultez [Utilisation 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 kits SDK AWS](#)
 - [Actions pour Amazon Pinpoint à l'aide de kits de développement logiciel AWS](#)
 - [Utilisation CreateApp avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateCampaign avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateExportJob avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateImportJob avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateSegment avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteApp avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteEndpoint avec un AWS SDK ou une CLI](#)
 - [Utilisation GetEndpoint avec un AWS SDK ou une CLI](#)
 - [Utilisation GetSegments avec un AWS SDK ou une CLI](#)
 - [Utilisation GetSmsChannel avec un AWS SDK ou une CLI](#)
 - [Utilisation GetUserEndpoints avec un AWS SDK ou une CLI](#)
 - [Utilisation SendMessages avec un AWS SDK ou une CLI](#)
 - [Utilisation UpdateEndpoint avec un AWS SDK ou une CLI](#)
- [Exemples de code pour l'API SMS et voix Amazon Pinpoint à l'aide de SDK AWS](#)
 - [Actions pour l'API SMS et voix Amazon Pinpoint à l'aide des SDK AWS](#)
 - [Utilisation SendVoiceMessage avec un AWS SDK ou une CLI](#)

Exemples de code pour Amazon Pinpoint à l'aide de kits SDK AWS

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 et dans des exemples interservices.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation 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

- [Actions pour Amazon Pinpoint à l'aide de kits de développement logiciel AWS](#)
 - [Utilisation CreateApp avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateCampaign avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateExportJob avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateImportJob avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateSegment avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteApp avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteEndpoint avec un AWS SDK ou une CLI](#)
 - [Utilisation GetEndpoint avec un AWS SDK ou une CLI](#)
 - [Utilisation GetSegments avec un AWS SDK ou une CLI](#)
 - [Utilisation GetSmsChannel avec un AWS SDK ou une CLI](#)
 - [Utilisation GetUserEndpoints avec un AWS SDK ou une CLI](#)
 - [Utilisation SendMessages avec un AWS SDK ou une CLI](#)
 - [Utilisation UpdateEndpoint avec un AWS SDK ou une CLI](#)

Actions pour Amazon Pinpoint à l'aide de kits de développement logiciel AWS

Les exemples de code suivants montrent comment effectuer des actions Amazon Pinpoint individuelles à l'aide AWS des SDK. Ces extraits appellent l'API Amazon Pinpoint et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. 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 la [référence de l'API Amazon Pinpoint](#).

Exemples

- [Utilisation CreateApp avec un AWS SDK ou une CLI](#)
- [Utilisation CreateCampaign avec un AWS SDK ou une CLI](#)
- [Utilisation CreateExportJob avec un AWS SDK ou une CLI](#)
- [Utilisation CreateImportJob avec un AWS SDK ou une CLI](#)
- [Utilisation CreateSegment avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteApp avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteEndpoint avec un AWS SDK ou une CLI](#)
- [Utilisation GetEndpoint avec un AWS SDK ou une CLI](#)
- [Utilisation GetSegments avec un AWS SDK ou une CLI](#)
- [Utilisation GetSmsChannel avec un AWS SDK ou une CLI](#)
- [Utilisation GetUserEndpoints avec un AWS SDK ou une CLI](#)
- [Utilisation SendMessages avec un AWS SDK ou une CLI](#)
- [Utilisation UpdateEndpoint avec un AWS SDK ou une CLI](#)

Utilisation **CreateApp** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser CreateApp.

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 [CreateApp](#) à la section Référence des AWS CLI commandes.

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.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 [CreateApp](#) à la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

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](#).

```
suspend fun createApplication(applicationName: String?): String? {  
  
    val createApplicationRequest0b = CreateApplicationRequest {  
        name = applicationName  
    }  
  
    PinpointClient { region = "us-west-2" }.use { pinpoint ->  
        val result = pinpoint.createApp(  
            CreateAppRequest {  
                createApplicationRequest = createApplicationRequest0b  
            }  
        )  
        return result.applicationResponse?.id  
    }  
}
```

- Pour plus de détails sur l'API, reportez-vous [CreateApp](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreateCampaign** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `CreateCampaign`.

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](#).

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 [CreateCampaign](#) à la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

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](#).

```
suspend fun createPinCampaign(appId: String, segmentIdVal: String) {

    val schedule0b = Schedule {
```

```
        startTime = "IMMEDIATE"
    }

    val defaultMessage0b = Message {
        action = Action.OpenApp
        body = "My message body"
        title = "My message title"
    }

    val messageConfiguration0b = MessageConfiguration {
        defaultMessage = defaultMessage0b
    }

    val writeCampaign = WriteCampaignRequest {
        description = "My description"
        schedule = schedule0b
        name = "MyCampaign"
        segmentId = segmentIdVal
        messageConfiguration = messageConfiguration0b
    }

    PinpointClient { 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 [CreateCampaign](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreateExportJob** avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser `CreateExportJob`.

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](#).

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 [CreateExportJob](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreateImportJob** avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser `CreateImportJob`.

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](#).

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 [CreateImportJob](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreateSegment** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `CreateSegment`.

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.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 [CreateSegment](#) à la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

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](#).

```
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 writeSegmentRequest0b = WriteSegmentRequest {
        name = "MySegment101"
        dimensions = dimensions0b
    }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val createSegmentResult: CreateSegmentResponse = pinpoint.createSegment(
            CreateSegmentRequest {
                applicationId = applicationIdVal
                writeSegmentRequest = writeSegmentRequest0b
            }
        )
        println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
        return createSegmentResult.segmentResponse?.id
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [CreateSegment](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeleteApp** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `DeleteApp`.

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 [DeleteApp](#) à la section Référence des AWS CLI commandes.

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](#).

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 [DeleteApp](#) à la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

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](#).

```
suspend fun deletePinApp(appId: String?) {  
  
    PinpointClient { 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 [DeleteApp](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeleteEndpoint** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `DeleteEndpoint`.

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](#).

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();

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

public static void deletePinEndpoint(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 [DeleteEndpoint](#) à la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

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](#).

```
suspend fun deletePinEndpoint(appIdVal: String?, endpointIdVal: String?) {  
  
    val deleteEndpointRequest = DeleteEndpointRequest {  
        applicationId = appIdVal  
        endpointId = endpointIdVal  
    }  
  
    PinpointClient { 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 [DeleteEndpoint](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetEndpoint** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent 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 récupère 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 [GetEndpoint](#) à la section Référence des AWS CLI commandes.

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 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 [GetEndpoint](#) à la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

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](#).

```
suspend fun lookupPinpointEndpoint(appId: String?, endpoint: String?) {

    PinpointClient { 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 [GetEndpoint](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetSegments** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `GetSegments`.

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](#).

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("Segment " + 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 [GetSegments](#) à la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

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](#).

```
suspend fun listSegs(appId: String?) {  
  
    PinpointClient { 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 [GetSegments](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation `GetSmsChannel` avec un AWS SDK ou une CLI

Les exemples de code suivants montrent 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 [GetSmsChannel](#) à la section Référence des AWS CLI commandes.

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.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 [GetSmsChannel](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetUserEndpoints** avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser `GetUserEndpoints`.

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.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 [GetUserEndpoints](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **SendMessage** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `SendMessage`.

.NET

AWS SDK for .NET

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](#).

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
    }
}
};
SendMessageResponse response = await
client.SendMessagesAsync(sendRequest);
return response;
}
}
```

- Pour plus de détails sur l'API, reportez-vous [SendMessages](#) à la section Référence des AWS SDK for .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 [SendMessages](#) à la section Référence des AWS CLI commandes.

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](#).

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.sendMessage(request);
        MessageResponse msg1 = response.getMessageResponse();
        Map map1 = msg1.getResult();

        // Write out the result of sendMessage.
        map1.forEach((k, v) -> System.out.println((k + ":" +
v)));

    } catch (PinpointException e) {
        System.err.println(e.getAwsErrorDetails().getErrorMessage());
        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.
 *
 * For more information, see the following documentation topic:
 *
 * 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.sendMessage(request);
        MessageResponse msg1 = response.getMessageResponse();
        Map map1 = msg1.getResult();

        // Write out the result of sendMessage.
        map1.forEach((k, v) -> System.out.println((k + ":" +
v)));

    } catch (PinpointException e) {
        System.err.println(e.getAwsErrorDetails().getErrorMessage());
        System.exit(1);
    }
}
}
```

- Pour plus de détails sur l'API, reportez-vous [SendMessages](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

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](#).

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.
var subject = "Amazon Pinpoint Test (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 = `
<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.
```



```
var 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);
    } else {
      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., XXXXXXXX66e4e9986478cXXXXXXXXXX
```

```
/* 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 the parameters to pass to the API.
var 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 [SendMessages](#) à la section Référence des AWS SDK for JavaScript API.

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](#).

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 = `
<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.sendMessage(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 [SendMessages](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

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](#).

```
/**
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 = destinationObj
        this.content = emailContent
    }

    PinpointEmailClient { region = "us-east-1" }.use { pinpointemail ->
        pinpointemail.sendEmail(sendEmailRequest)
        println("Message Sent")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [SendMessages](#) à la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour 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](#).

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 [SendMessages](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **UpdateEndpoint** avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser `UpdateEndpoint`.

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.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 [UpdateEndpoint](#) à 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, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples de code pour l'API SMS et voix Amazon Pinpoint à l'aide de SDK AWS

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 et dans des exemples interservices.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation 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

- [Actions pour l'API SMS et voix Amazon Pinpoint à l'aide des SDK AWS](#)
 - [Utilisation SendVoiceMessage avec un AWS SDK ou une CLI](#)

Actions pour l'API SMS et voix Amazon Pinpoint à l'aide des SDK AWS

Les exemples de code suivants montrent comment effectuer des actions individuelles sur les API SMS et Voice Amazon Pinpoint à l'aide AWS des SDK. Ces extraits appellent l'API de messages SMS et vocaux d'Amazon Pinpoint et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. 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 SendVoiceMessage avec un AWS SDK ou une CLI](#)

Utilisation **SendVoiceMessage** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `SendVoiceMessage`.

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.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.
 *
 * For more information, see the following documentation topic:
 *
 * 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, voir [SendVoiceMessage](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

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](#).

```
"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, voir [SendVoiceMessage](#) la section Référence des AWS SDK for JavaScript API.

Python

SDK pour 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](#).

```
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,
    ssm1_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"
    ssmml_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,
        ssmml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")

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

- Pour plus de détails sur l'API, consultez [SendVoiceMessage](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation d'Amazon Pinpoint avec un SDK AWS](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de 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 cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de [AWS conformité Programmes](#) de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à 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](#) 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 [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog [Modèle de responsabilité partagée AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (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.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- 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 140-2 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 (Federal Information Processing Standard) disponibles, consultez [Federal Information Processing Standard \(FIPS\) 140-2](#) (Normes de traitement de l'information fédérale).

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 Name (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 AWS 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 AWS Direct Connect routeur. Pour plus d'informations, consultez [Présentation d' AWS Direct Connect](#) dans le Guide de l'utilisateur AWS 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 PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La

plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un identifiant de clé d'accès et d'une clé d'accès secrète associés à un principal AWS Identity and Access Management (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 ressources AWS

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 PrivateLink](#) une technologie qui vous permet d'accéder en privé aux API Amazon Pinpoint sans passerelle Internet, appareil NAT, connexion VPN ou AWS 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

AWS Identity and Access Management (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 des 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 AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans Amazon Pinpoint.

Utilisateur du service : si vous utilisez le service Amazon Pinpoint service pour accomplir votre tâche, votre administrateur vous fournira les informations d'identification et les autorisations nécessaires. Vous pourrez avoir besoin d'autorisations supplémentaires si vous utilisez davantage de fonctions Amazon Pinpoint. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonction dans Amazon Pinpoint, consultez [Dépannage de la gestion des identités et des accès Amazon Pinpoint](#).

Administrateur du service : si vous êtes le responsable des ressources Amazon Pinpoint de votre entreprise, vous bénéficiez probablement d'un accès total à ce service. C'est à vous de déterminer les fonctions et les ressources Amazon Pinpoint auxquelles vos utilisateurs des services pourront accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour

comprendre les concepts de base d'IAM. Pour découvrir la façon dont votre entreprise peut utiliser IAM avec Amazon Pinpoint, consultez [Fonctionnement d'Amazon Pinpoint avec IAM](#).

Administrateur IAM : si vous êtes un administrateur IAM, vous pouvez souhaiter obtenir des informations sur l'écriture de politiques pour gérer l'accès à Amazon Pinpoint. Pour afficher des exemples de politiques basées sur l'identité Amazon Pinpoint que vous pouvez utiliser dans IAM, consultez [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 (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section [Comment vous connecter à votre compte Compte AWS dans](#) le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section [Signature des demandes AWS d'API](#) dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez [Authentification multifactorielle](#) dans le Guide de l'utilisateur AWS IAM Identity Center

et [Utilisation de l'authentification multifactorielle \(MFA\) dans l'interface AWS](#) dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur racine, consultez [Tâches nécessitant 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 au sein de votre Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification](#) dans le Guide de l'utilisateur IAM.

Un [groupe IAM](#) est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez [Quand créer un utilisateur IAM \(au lieu d'un rôle\)](#) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un [rôle IAM](#) est une identité au sein de votre Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console [changeant de rôle](#). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez [Utilisation de rôles IAM](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez [Différence entre les rôles IAM et les politiques basées sur les ressources](#) dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
 - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service.

FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez [Sessions de transmission d'accès](#).

- **Rôle de service** : il s'agit d'un [rôle IAM](#) attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.
- **Rôle lié à un service** — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- **Applications exécutées sur Amazon EC2** : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez [Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2](#) dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez [Quand créer un rôle IAM \(au lieu d'un utilisateur\)](#) dans le Guide de l'utilisateur IAM.

Gestion des 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 est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des

documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

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.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

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

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique 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.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de

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

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

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 d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- **Limite d'autorisations** : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui

spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.

- **Politiques de contrôle des services (SCP)** — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez [Fonctionnement des SCP](#) dans le Guide de l'utilisateur AWS Organizations .
- **Politiques de séance** : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez [politiques de séance](#) 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 détermine 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 AWS Identity and Access Management (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. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie 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. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, 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 Resource 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-12345678901234567890123456789012`, utilisez l'ARN suivant :

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/  
phone-12345678901234567890123456789012"
```

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` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. 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.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération AND logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une OR opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez [Éléments d'une politique IAM : variables et identifications](#) dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. 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

Les politiques d'autorisation basées sur les ressources sont des documents de politique JSON précisant les actions qu'un principal spécifié peut effectuer sur la 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 “Exemples de politiques basées sur l’identité”](#).

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 [Balisage des 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 [AssumeRole](#) ou [GetFederationToken](#).

Amazon Pinpoint est compatible avec l'utilisation des informations d'identification temporaires.

Rôles liés à un service

Les [rôles liés aux](#) AWS services permettent aux services d'accéder aux ressources d'autres services pour effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre compte IAM et sont la propriété du 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 de 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 AWS Identity and Access Management 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](#)
- [Mots de passe uniques](#)
- [Validation du numéro de téléphone](#)
- [Projets](#)
- [Modèles de recommandation](#)
- [Segments](#)
- [Balises](#)
- [Users](#)

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`

Mots de passe uniques

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*

Balises

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*:*

Users

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 — `/sms-voice/configuration-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-voice/configuration-sets/ /event-destinations *ConfigurationSetName*
- Méthode – GET
- ARN des ressources – Non disponible Utilisez *.

sms-voice:CreateConfigurationSetEventDestination

Créez une destination d'événement pour les événements vocaux.

- URI — /sms-voice/configuration-sets/ /event-destinations *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-voice/configuration-sets/ /destinations/
ConfigurationSetNameEventDestinationName
- Méthode – PUT
- ARN des ressources – Non disponible Utilisez *.

sms-voice>DeleteConfigurationSetEventDestination

Supprimez une destination d'événement pour les événements vocaux.

- URI — /sms-voice/configuration-sets/ /destinations/
ConfigurationSetNameEventDestinationName
- Méthode – DELETE
- ARN des ressources – Non disponible Utilisez *.

sms-voice:SendVoiceMessage

Créez et envoyez des messages vocaux.

- URI – /sms-voice/voice/message

- 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

Les politiques basées sur l'identité déterminent si une personne peut créer, consulter 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.
- Accorder 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 de IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utiliser 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 AWS CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer 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 politique 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 le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Configuration de l'accès aux API protégé par 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 II 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UseConsole",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    },
    {
      "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 *la région* 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewProject",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetApps",
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": [
```

```

        "arn:aws:mobiletargeting:region:accountId:apps/projectId",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/*",
        "arn:aws:mobiletargeting:region:accountId: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 AWS votre identifiant de 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 :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LimitedWriteProject",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetApps",
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*",
        "mobiletargeting:Create*",
        "mobiletargeting:Update*",
        "mobiletargeting:Put*"
      ],
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/810c7aab86d42fb2b56c8c966example",
        "arn:aws:mobiletargeting:region:accountId:apps/810c7aab86d42fb2b56c8c966example/*",
        "arn:aws:mobiletargeting:region:accountId: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"
        }
      }
    }
  ]
}

```

```

    }
  }
]
}

```

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.

```

{
  "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": "accountId"
        },
        "ArnLike": {

```

```

        "aws:SourceArn": "arn:aws:mobiletargeting:region:accountId:*"
    }
}
]
}

```

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 `Owner=richard-roe` ou `owner=richard-roe`. Dans le cas contraire, l'utilisateur se voit refuser l'accès. La clé de condition d'étiquette `Owner` correspond à la fois à `Owner` 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:ListGroupsWithUser",
        "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](#).

Accès en lecture seule

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewAllResources",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    }
  ]
}
```

Dans l'exemple précédent, remplacez *region* par le nom d'une AWS région, et *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 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FullAccess",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:*"
      ],
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    }
  ]
}
```

Dans l'exemple précédent, remplacez *accountId* par votre ID de compte AWS .

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](#).

Accès en lecture seule

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMSVoiceReadOnly",
      "Effect": "Allow",
      "Action": [
```

```

        "sms-voice:Get*",
        "sms-voice:List*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:sms-voice:region:accountId:"
        }
    }
}
]
}

```

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 :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMSVoiceFullAccess",
      "Effect": "Allow",
      "Action": [
        "sms-voice:*",
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:sms-voice:region:accountId:"
        }
      }
    }
  ]
}

```

Exemple : restriction de l'accès d'un projet Amazon Pinpoint à des adresses IP spécifiques

L'exemple de stratégie suivant accorde des autorisations à tout utilisateur pour effectuer une action Amazon Pinpoint sur un projet spécifié (*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.

```
{
  "Version": "2012-10-17",
  "Id": "AMZPinpointPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "mobiletargeting:*",
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId",
        "arn:aws:mobiletargeting:region:accountId: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 stratégie suivant accorde des autorisations pour effectuer une action Amazon Pinpoint sur un projet spécifié (*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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ModifyResourceIfOwner",
      "Effect": "Allow",
      "Action": "mobiletargeting:*",
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId",
        "arn:aws:mobiletargeting:region:accountId: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.

```
{
  "Version": "2008-10-17",
```

```

"Statement":[
  {
    "Sid":"PinpointEmail",
    "Effect":"Allow",
    "Principal":{
      "Service":"pinpoint.amazonaws.com"
    },
    "Action":"ses:*",
    "Resource":"arn:aws:ses:region:accountId:identity/emailId",
    "Condition":{
      "StringEquals":{
        "aws:SourceAccount":"accountId"
      },
      "StringLike":{
        "aws:SourceArn":"arn:aws:mobiletargeting:region:accountId:apps/*"
      }
    }
  }
]
}

```

Si vous utilisez Amazon Pinpoint dans la région AWS GovCloud (ouest des États-Unis), utilisez plutôt l'exemple de politique suivant :

```

{
  "Version":"2008-10-17",
  "Statement":[
    {
      "Sid":"PinpointEmail",
      "Effect":"Allow",
      "Principal":{
        "Service":"pinpoint.amazonaws.com"
      },
      "Action":"ses:*",
      "Resource":"arn:aws-us-gov:ses:us-gov-west-1:accountId:identity/emailId",
      "Condition":{
        "StringEquals":{
          "aws:SourceAccount":"accountId"
        },
        "StringLike":{
          "aws:SourceArn":"arn:aws-us-gov:mobiletargeting:us-gov-
west-1:accountId:apps/*"
        }
      }
    }
  ]
}

```

```
}  
    }  
  ]  
}
```

Rôles IAM pour les tâches Amazon Pinpoint courantes

Un [rôle IAM](#) est une identité AWS Identity and Access Management (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. Avant d'importer, vous devez déléguer les autorisations requises à Amazon Pinpoint. Pour ce faire, vous créez un rôle AWS Identity and Access Management (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 for Java, consultez ce [Importation des segments](#) guide.

Création d'un 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.

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



```

        "Effect": "Allow",
        "Principal": {
            "Service": "pinpoint.amazonaws.com"
        },
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "accountId"
            },
            "ArnLike": {
                "arn:aws:mobiletargeting:region:accountId:apps/application-id"
            }
        }
    }
]
}

```

Dans l'exemple précédent, procédez comme suit :

- Remplacez *la région* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
 - Remplacez *AccountID* par l'identifiant unique de votre compte. AWS
 - 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 :

```

{
  "Role": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",

```

```

        "Effect": "Allow",
        "Principal": {
            "Service": "pinpoint.amazonaws.com"
        },
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "accountId"
            },
            "ArnLike": {
                "aws:SourceArn":
                "arn:aws:mobiletargeting:region:accountId:apps/application-id"
            }
        }
    }
}
],
"RoleId": "AIDACKCEVSQ6C2EXAMPLE",
"CreateDate": "2016-12-20T00:44:37.406Z",
"RoleName": "PinpointSegmentImport",
"Path": "/",
"Arn": "arn:aws:iam::accountId:role/PinpointSegmentImport"
}
}

```

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 la liste au format JSON des points de terminaison et de leurs attributs (le canal, l'adresse, la possibilité d'accepter ou de refuser le statut, la date de création et l'ID de point de terminaison, par exemple).

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 :

```
{
  "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:::example-bucket" ],
      "Condition": {
        "StringEquals": {
          "s3:delimiter": [ "/" ],
          "s3:prefix": [
            "",
            "Exports/"
          ]
        }
      }
    },
    {
      "Sid": "AllowListingOfUserFolder",
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [ "arn:aws:s3:::example-bucket" ],
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "Exports/*"
          ]
        }
      }
    },
    {
      "Sid": "AllowAllS3ActionsInUserFolder",
      "Action": [ "s3:*" ],
      "Effect": "Allow",
      "Resource": [ "arn:aws:s3:::example-bucket/Exports/*" ]
    }
  ]
}
```

Dans le code précédent, remplacez toutes les instances d'*example-bucket* par le nom du compartiment Amazon S3 qui contient le dossier vers lequel vous souhaitez exporter les informations du segment. En outre, remplacez l'instance d'*Exports* 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

la rubrique [Ajout et suppression d'autorisations basées sur l'identité IAM](#) du 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 :

```
{
  "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:region:accountId: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/s3* par l'ARN de la politique que vous avez créée dans la section précédente. ExportPolicy

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 AWS Identity and Access Management (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 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RetrieveRecommendationsOneCampaign",
      "Effect": "Allow",
      "Action": [
        "personalize:DescribeSolution",
        "personalize:DescribeCampaign",
        "personalize:GetRecommendations"
      ],
      "Resource": [
        "arn:aws:personalize:region:accountId:solution/solutionId",
        "arn:aws:personalize:region:accountId:campaign/campaignId"
      ]
    }
  ]
}
```



```
}
```

Dans l'exemple précédent, remplacez le texte *en italique* par vos informations :

- *region* : nom de la région AWS qui héberge la solution et la campagne Amazon Personalize.
 - *accountId* : votre ID Compte AWS .
 - *solutionId* : ID de ressource unique pour la solution Amazon Personalize déployée par la campagne.
 - *campaignId* : ID de ressource unique pour la campagne Amazon Personalize à partir de laquelle 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 la rubrique [Ajout et suppression d'autorisations basées sur l'identité IAM](#) du 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 :

```
{
  "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:region:accountId: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`.

- 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
```

- 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 *`RetrieveRecommendationsPolicy`* 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 AWS Identity and Access Management (IAM) avec les autorisations requises. Pour plus d'informations, consultez [Diffusion en continu des événements Amazon Pinpoint vers Amazon 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 [Diffusion d'événements Amazon Pinpoint sur Kinesis](#) dans ce guide.

Création d'un 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 *la région* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
 - Remplacez *AccountID* par l'identifiant unique de votre compte. AWS
 - Remplacez *ApplicationID* par l'ID unique du projet.

```
{
  "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:region:accountId:apps/applicationId"
        }
      }
    ]
  }

```

Lorsque vous avez terminé, enregistrez le fichier sous `PinpointEventStreamTrustPolicy.json`.

- 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

```

- 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 *la région* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
- Remplacez *AccountID* par l'identifiant unique de votre compte. AWS
- Remplacez *StreamName* par le nom de votre flux Kinesis.

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Action": [
      "kinesis:PutRecords",
      "kinesis:DescribeStream"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:kinesis:region:accountId: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 *la région* par la AWS région dans laquelle vous utilisez Amazon Pinpoint.
- Remplacez *AccountID* par l'identifiant unique de votre compte. AWS
- *delivery-stream-name* Remplacez-le par le nom de votre stream Firehose.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "firehose:PutRecordBatch",
      "firehose:DescribeDeliveryStream"
    ],
    "Resource": [
      "arn:aws:firehose:region:accountId: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 AWS Identity and Access Management (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 la section [Tester les politiques IAM avec le simulateur de politiques IAM](#) dans le guide de l'utilisateur [IAM](#).

Création d'un 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 la *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 l' AWS GovCloud ouest des États-Unis est `aws-us-gov`.
 - Remplacez *la région* par le nom de celle Région AWS qui héberge le projet Amazon Pinpoint.
 - Remplacez *AccountID* par l'identifiant unique de votre. Compte AWS

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

```

        "Sid": "PinpointUsesSESEmailSends",
        "Effect": "Allow",
        "Action": [
            "ses:SendEmail",
            "ses:SendRawEmail"
        ],
        "Resource": [
            "arn:partition:ses:region:accountId:identity/*",
            "arn:partition:ses:region:accountId: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.
 - Remplacez *AccountID* par l'identifiant unique de votre compte AWS

```

{
  "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 à 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 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, un utilisateur doit 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 ne dispose pas des autorisations nécessaires pour transférer 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 à 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 en savoir plus, 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 découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez [Différence entre les rôles IAM et les politiques basées sur les ressources](#) 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 à AWS CloudTrail, un service qui fournit un enregistrement des actions effectuées dans Amazon Pinpoint par un utilisateur, un rôle ou un autre service. 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 [Journalisation des 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' AWS Identity and Access Management 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 AWS Support .

Validation de la conformité pour Amazon Pinpoint

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon Pinpoint dans le cadre de plusieurs programmes de conformité AWS . Il s'agit notamment des contrôles AWS du système et de l'organisation (SOC), FedRAMP, HIPAA, ISO/IEC 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, ISO/IEC 9001:2015 pour les systèmes de gestion de la qualité, etc.

Pour obtenir la liste des AWS services concernés par des programmes de conformité spécifiques, voir [AWS Services concernés par programme de conformité AWS](#) . 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](#) 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 conformes à la loi HIPAA. AWS
- 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](#) — 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. 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.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

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 des 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 for Android ou le AWS Mobile SDK for iOS. Pour les applications mobiles ou Web JavaScript basées, 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	Quota de débordement/taux par défaut (demandes par seconde)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10
CreateImportJob	300

Opération	Quota de débordement/taux 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	Quota de débordement/taux 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 `SendMessage`, une demande simultanée pour l'opération `UpdateEndpoint` peut aboutir.

Quotas de campagne

Les quotas suivants s'appliquent à la ressource [Campagnes](#) 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
	<p> Note</p> <p>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_N EXT_RUN .</p>	
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	<p>Chaque projet peut comporter jusqu'à 25 campagnes envoyées lorsque des événements se produisent.</p> <p>Les campagnes qui utilisent des déclencheurs basés sur des événements doivent employer des segments dynamiques. Elles ne peuvent pas utiliser de segments importés.</p> <p>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 for Android version 2.7.2 ou ultérieure, ou la version 2.6.30 ou ultérieure. AWS Mobile SDK for iOS</p> <p>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.</p>	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	Non

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.


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	<p>Si votre compte se trouve encore dans l'environnement de test (sandbox), l'ensemble des adresses ou domaines de messagerie doivent être vérifiés.</p> <p>Si votre compte se trouve en dehors de l'environnement de test, vous pouvez envoyer un e-mail à n'importe quelle adresse valide.</p>	Oui
Nombre de destinataires par message	50 destinataires par message	Non
Nombre d'identités que vous pouvez vérifier	10 000 identités par AWS région <div data-bbox="592 1066 1031 1717" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"><p> Note</p><p>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.</p></div>	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 de l'environnement de test (sandbox) et que votre quota/taux d'envoi a été augmenté, ces modifications s'appliquent toutes à votre compte Pinpoint dans us-east-1.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre d'e-mails pouvant être envoyés par période de 24 heures (quota d'envoi)	<p>Si votre compte se trouve dans l'environnement de test : 200 e-mails par période de 24 heures.</p> <p>Si votre compte est en dehors de l'environnement de test, le quota varie en fonction de votre cas d'utilisation.</p> <div data-bbox="591 1020 1029 1570" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>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 ..</p> </div>	Oui
Nombre d'e-mails pouvant être envoyés chaque seconde (taux d'envoi)	<p>Si votre compte est dans l'environnement de test : 1 e-mail par seconde.</p> <p>Si votre compte est en dehors de l'environnement de test, le</p>	Oui

Ressource	Quota par défaut	Éligible à une augmentation
	taux varie en fonction de votre cas d'utilisation. <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>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 :</p> </div>	

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 <code>Attributes</code> , <code>Metrics</code> et <code>UserAttributes</code> , collectivement	250 pour tous les paramètres d'attribut par application	Non

Ressource	Quota par défaut	Éligible à une augmentation
Attributs assignés au paramètre <code>Attributes</code>	250 pour tous les paramètres d'attribut par application	Non
Attributs assignés au paramètre <code>Metrics</code>	250 pour tous les paramètres d'attribut par application	Non
Attributs assignés au paramètre <code>UserAttributes</code>	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 <code>EndpointBatchItem</code> dans une charge utile <code>EndpointBatchRequest</code>	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 <code>Attributes</code>	50 par attribut	Non
Valeurs assignées aux attributs de paramètre <code>UserAttributes</code>	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 des SDK AWS mobiles et de la ressource [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 tentatives d'invocations 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 maximal de caractères dans les parties de modèle spécifiques à l'APN d'un modèle de notification push	4 000 caractères	Non

Ressource	Quota par défaut	Éligible à une augmentation
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 (APN)	4 Ko par message	Non
Taille de la charge utile des messages de l'environnement de test (sandbox) des APN	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 <code>GetInAppMessages</code> 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

Les quotas suivants s'appliquent au canal des SMS.

Pour plus d'informations sur le coût des SMS, consultez [Comprendre la facturation et les rapports d'utilisation des SMS pour Amazon Pinpoint](#) dans le Guide de l'utilisateur Amazon Pinpoint.

Ressource	Quota par défaut	Éligible à une augmentation
Seuil de dépenses	1 USD par compte	Oui , mais les limites de dépenses varient selon les

Ressource	Quota par défaut	Éligible à une augmentation
		<p>régions. Vous devez spécifier la ou les régions dans lesquelles vous souhaitez une augmentation.</p>
<p>Nombre de SMS pouvant être envoyés par seconde (fréquence d'envoi#)</p>	<p>Varie en fonction du pays de destination et du numéro de téléphone d'origine. Pour plus d'informations, consultez Limites de parties par seconde des messages dans le guide de l'utilisateur Amazon Pinpoint.</p>	<p>Oui, mais il se peut que vous deviez obtenir un numéro de téléphone compatible avec un débit plus élevé. Si vous ne savez pas quel type de numéro utiliser, AWS Support contactez votre responsable de AWS compte pour plus d'informations</p> <p>Si vous utilisez un ID d'expéditeur alphanumérique pour envoyer des messages, vous pourrez peut-être augmenter votre débit. Pour savoir si une augmentation du débit est disponible pour votre ID d'expéditeur, ouvrez une demande d'ID d'expéditeur dans la console du Centre de support. Dans votre demande, précisez votre ID d'expéditeur existant, le pays dans lequel vous utilisez cet ID d'expéditeur et le débit que vous souhaitez demander.</p>

Ressource	Quota par défaut	Éligible à une augmentation
Nombre de messages SMS pouvant être envoyés à un seul destinataire chaque seconde	1 message par seconde.	Non
Nombre de rubriques Amazon SNS pour les SMS bidirectionnels	100,000 par compte	Oui
Nombre de mots clés pour les messages SMS bidirectionnels	30 mots clés par numéro	Oui
Nombre de numéros de messages SMS et vocaux	25 par compte et région	Oui
Nombre de numéros de téléphone dédiés	25 par compte	Oui
<p>Nombre de listes de désabonnement</p> <p>Remarque : la liste de désabonnement par défaut requise est prise en compte dans ce quota.</p>	25 par compte	Oui
Nombre de jeux de configurations	25 par compte	Oui
Nombre de destinations d'événement	5 par jeu de configuration	Non
Nombre de numéros de téléphone de destination vérifiés dans l'environnement de test (sandbox) SMS	10 par compte	Oui

Ressource	Quota par défaut	Éligible à une augmentation
Nombre de groupes de numéros de téléphone	25 par compte	Oui
Nombre d'identités d'origine pouvant être associées à un pool de numéros de téléphone	100 par pool de numéros de téléphone	Oui

Quotas de 10DLC

Les quotas suivants s'appliquent aux SMS envoyés à l'aide de numéros de téléphone 10DLC. Les numéros 10DLC ne peuvent être utilisés que pour envoyer des messages à des destinataires aux États-Unis.

Ressource	Quota par défaut	Éligible à une augmentation
Maximum de 10 entreprises de DLC par Compte AWS	25	Oui
Nombre maximal de campagnes 10DLC par société 10DLC	10	Oui
Nombre maximal de numéros 10DLC par campagne 10DLC	49	Non


Quotas de messages vocaux

Les quotas suivants s'appliquent au canal vocal.

Note

Lorsque votre compte est supprimé de l'environnement de test, vous êtes automatiquement éligible aux quotas maximaux indiqués dans le tableau suivant.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre de messages vocaux qu'il est possible d'envoyer sur une période de 24 heures	Si votre compte se trouve dans l'environnement de test (sandbox) : 20 messages	Non
Nombre de messages vocaux qu'il est possible d'envoyer à un seul destinataire au cours d'une période de 24 heures	5 messages	Non
Nombre de messages vocaux qu'il est possible d'envoyer par minute	Si votre compte se trouve dans l'environnement de test (sandbox) : 5 appels par minute Si votre compte se trouve en dehors de l'environnement de test (sandbox) : 20 appels par minute	Non
Nombre de messages vocaux qu'il est possible d'envoyer à partir d'un seul numéro de téléphone d'origine par seconde	1 message par seconde.	Non
Longueur du message vocal	Si votre compte se trouve dans l'environnement de test (sandbox) : 30 secondes Si votre compte se trouve en dehors de l'environnement de test (sandbox) : 5 minutes	Non
Capacité à envoyer des messages vocaux à des	Si votre compte se trouve dans l'environnement de test (sandbox), vous ne pouvez	Non

Ressource	Quota par défaut	Éligible à une augmentation
numéros de téléphone internationaux	<p>envoyer des messages qu'aux destinataires des pays suivants :</p> <ul style="list-style-type: none">• Australie• Canada• Allemagne• Hong Kong• Israël• Japon• Mexique• Singapour• Suède• États-Unis• Royaume-Uni <p>Si votre compte se trouve en dehors de l'environnement de test (sandbox), vous pouvez envoyer des messages à des destinataires de n'importe quel pays.</p> <div data-bbox="594 1381 1029 1843"><p> Note</p><p>Les appels internationaux sont soumis à des frais supplémentaires, qui varient en fonction du pays ou de la région de destination.</p></div>	

Ressource	Quota par défaut	Éligible à une augmentation
Nombre de caractères dans un message vocal	3 000 caractères facturables, dans les mots prononcés 6 000 caractères au total (y compris les caractères facturables et les balises SSML)	Non
Nombre de jeux de configurations	10 000 ensembles de configuration vocale par AWS région	Non

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 sur <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 Envoyer.

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
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 Amazon Pinpoint SMS. Pour plus d'informations, consultez le guide de l'utilisateur Amazon Pinpoint SMS .	8 février 2024
Quotas Amazon Pinpoint	Quotas ajoutés pour les règles relatives au nombre maximum de jours fermés,	19 décembre 2023

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](#).

[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 Amazon Pinpoint SMS and Voice, consultez le nouveau [Guide de l'utilisateur Amazon Pinpoint SMS](#).

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 et 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 (OTP) 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 obtenir plus d'informations, consultez [Envoi et validation de mots de passe uniques \(OTP\)](#).

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. AWS 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 <code>delivery_type</code> 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.

Modification	Description	Date
Disponibilité par région	Amazon Pinpoint est désormais disponible dans les	21 décembre 2018

Modification	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

Modification	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 rechercher des points de terminaison. Des exemples sont fournis pour le AWS CLI AWS SDK for Java, et l'API Amazon Pinpoint. Pour plus d'informations, consultez Définition de votre audience pour Amazon Pinpoint et Accès aux données d'audience 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.	1 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 AWS SDK ou de bibliothèques.	23 mars 2018

Modification	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 supplémentaires sur les quotas d'e-mails.	19 janvier 2018
Bêta publique pour les extensions Amazon Pinpoint	Utilisez AWS Lambda les fonctions pour personnaliser les segments ou créer des canaux de messagerie personnalisés .	28 novembre 2017
Quotas de charges applicatives 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

Modification	Description	Date
ID utilisateur et événements d'authentification avec groupes d'utilisateurs Amazon Cognito.	Si vous utilisez des groupes d'utilisateurs Amazon Cognito pour gérer la connexion des utilisateurs dans vos applications mobiles, Amazon Cognito attribue des ID utilisateur aux points de terminaison, et envoie les événements d'authentification à Amazon Pinpoint.	26 septembre 2017
ID utilisateur	Attribuez des ID utilisateur aux points de terminaison pour surveiller l'utilisation des applications par les différents utilisateurs. Des exemples sont fournis pour les kits SDK AWS Mobile et les kits SDK pour Java .	31 août 2017
Événements d'authentification	Signalez les événements d'authentification pour savoir à quelle fréquence les utilisateurs s'authentifient auprès de votre application. Des exemples sont fournis dans Rapport des événements 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

Modification	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
Autorisations	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

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.