aws

Guía para desarrolladores

Amazon Pinpoint

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

Amazon Pinpoint: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

..... x
¿Qué es Amazon Pinpoint? 1
Utilizar Amazon Pinpoint para enviar mensajes a segmentos de público y analizar datos 1
Definición de los segmentos de público 1
Programar campañas de mensajería 2
Envío de mensajes de transacciones 2
Utilizar informes de análisis y métricas 2
Utilizar puntos de conexión para definir a su público 3
Agregar puntos de conexión 4
Ejemplos 5
Información relacionada 10
Asociar usuarios con puntos de conexión 11
Ejemplos 12
Información relacionada 16
Agregar un lote de puntos de conexión 17
Ejemplos 17
Información relacionada 25
Importar puntos de conexión 25
Antes de empezar 25
Ejemplos 26
Información relacionada 38
Exportar puntos de conexión de Amazon Pinpoint a buckets de Amazon S3 39
Antes de empezar 39
Ejemplos 40
Información relacionada 51
Buscar puntos de conexión en un proyecto de Amazon Pinpoint 51
Ejemplos 52
Información relacionada 57
Descripción de ID de puntos de conexión 57
Administrar el número máximo de puntos de conexión 60
Eliminar puntos de conexión 61
Ejemplos 61
Crear o importar segmentos 65
Generar segmentos 65

Cree segmentos con AWS SDK para Java ... 65
Importar segmentos ... 69
Importar un segmento con AWS SDK para Java ... 69
Personalizar segmentos ... 72
Datos de evento ... 73
Crear una función de Lambda ... 74
Asignar una política de función de Lambda ... 77
Asignar una función de Lambda a una campaña ... 79
Crear campañas de Amazon Pinpoint mediante programación ... 81
Crear una campaña con el SDK para Java ... 81
Cree una campaña de A/B prueba ... 84
Administración de etiquetas ... 87
Utilizar etiquetas en políticas de IAM ... 88
Agrega etiquetas a los recursos de 89
Agregar etiquetas mediante la API ... 89
Añada etiquetas mediante la AWS CLI ... 90
Mostrar etiquetas de recursos ... 92
Mostrar etiquetas mediante la API ... 92
Muestre las etiquetas mediante el AWS CLI ... 93
Actualizar o sobrescribir etiquetas ... 94
Quitar etiquetas de los recursos ... 94
Quitar etiquetas mediante la API ... 94
Elimine las etiquetas mediante el AWS CLI ... 95
Integrar con la aplicación ... 97
Cómo utilizar los AWS SDK ... 98
Conectar la aplicación frontend mediante Amplify ... 99
Siguiente paso ... 99
Registrar puntos de conexión en la aplicación ... 99
Antes de empezar ... 100
AWS mobile SDKs ... 100
AWS Amplify ... 100
Siguientes pasos ... 100
Notificar eventos en la aplicación ... 101
Antes de empezar ... 102
AWS móvil SDKs ... 102
Web y React Native ... 100

API de eventos de Amazon Pinpoint ... 103
Siguientes pasos ... 103
Enviar mensajes transaccionales desde la aplicación ... 104
Enviar correos electrónicos transaccionales ... 104
Elegir un método de envío de correos electrónicos ... 105
Elegir entre Amazon Pinpoint y Amazon SES ... 105
Envío de correo electrónico mediante la API ... 105
Agregar encabezados de cancelación de suscripción a correos electrónicos ... 120
Envío de mensajes SMS ... 122
Enviar mensajes de voz ... 135
Usar la API de SMS y voz ... 144
Generar contraseñas de un solo uso ... 146
Respuesta de SendOtpMessage ... 149
Validar mensajes de OTP ... 150
Respuesta de VerifyOtpMessage ... 150
Ejemplos de código OTP en Amazon Pinpoint ... 151
Generar un ID de referencia ... 151
Enviar códigos de OTP ... 152
Validar códigos de OTP ... 153
Personalizar mensajes integrados en la aplicación ... 155
Recuperar mensajes integrados en la aplicación para un punto de conexión ... 155
Ejemplo de JSON de respuesta de la API GetInAppMessages ... 158
Objeto InAppMessageCampaigns ... 159
Objeto InAppMessage ... 161
Objeto HeaderConfig ... 162
Objeto BodyConfig ... 163
Objeto InAppMessageContent ... 164
Objeto Schedule ... 165
Objeto InAppMessageButton ... 165
Objeto DefaultButtonConfig ... 167
Objeto OverrideButtonConfig ... 168
Validación de números de teléfono ... 170
Casos de uso de validación de números de teléfono de Amazon Pinpoint ... 170
Valide un número de teléfono mediante el AWS CLI ... 171
Respuesta de validación de números de teléfono ... 172
Crear un canal personalizado ... 176

Usar un webhook ... 176
Utilizar una función de Lambda ... 176
Puede asignar una función de Lambda o un webhook a una campaña individual. ... 178
Crear y configurar una función de Lambda para una campaña de Amazon Pinpoint ... 179
Ejemplo de función de Lambda ... 179
Formato de respuesta de función de Lambda para Amazon Pinpoint ... 184
Conceder permiso para invocar la función ... 185
Transmitir datos de eventos de aplicación ... 188
Configurar transmisiones de datos de eventos ... 189
Requisitos previos ... 189
AWS CLI ... 190
AWS SDK para Java ... 190
Flujo de datos de eventos de aplicación desde Amazon Pinpoint ... 191
Ejemplo de evento de aplicación ... 191
Atributos de los eventos de aplicación ... 192
Flujo de datos de eventos de campaña desde Amazon Pinpoint ... 198
Ejemplo de evento de campaña ... 198
Atributos de los eventos de campaña ... 199
Datos de eventos de recorrido de Amazon Pinpoint ... 206
Ejemplo de evento de recorrido ... 207
Atributos de eventos de recorrido ... 208
Flujo de datos de eventos de correo electrónico desde Amazon Pinpoint ... 213
Ejemplos de eventos de correo electrónico ... 214
Atributos de los eventos de correo electrónico ... 220
Flujo de datos de eventos de SMS desde Amazon Pinpoint ... 228
Ejemplo de evento de SMS ... 228
Atributos de los eventos de SMS ... 229
Eliminar un flujo de eventos ... 239
AWS CLI ... 239
AWS SDK para Java ... 239
Consultar datos de análisis ... 240
Consultar métricas en Amazon Pinpoint ... 241
Políticas de IAM ... 242
Métricas estándar para proyectos, campañas y recorridos ... 246
Métricas de aplicación para campañas ... 248
Métricas de aplicación para correos electrónicos ... 254

Métricas de aplicación para SMS ... 266
Métricas de campaña ... 274
Métricas de interacción del recorrido ... 286
Métricas de ejecución de recorrido ... 294
Métricas de ejecución de actividades de recorrido ... 296
Métricas sobre la ejecución del recorrido y la campaña ... 302
Consultar datos de campañas ... 304
Requisitos previos ... 305
Consultar datos de una campaña ... 306
Consultar datos de varias campañas ... 311
Consultar datos de mensajes transaccionales ... 318
Requisitos previos ... 318
Consultar datos de correos electrónicos transaccionales ... 319
Consultar datos de SMS transaccionales ... 324
Usar resultados de consultas JSON ... 330
Estructura JSON ... 331
Objetos y campos JSON ... 336
Registre las llamadas a la API con CloudTrail ... 339
Información de Amazon Pinpoint en CloudTrail ... 339
Acciones de la API en los archivos de CloudTrail registro ... 341
Acciones de la API de correo electrónico en los archivos de CloudTrail registro ... 345
Acciones de la API de voz y SMS v1 compatibles en los archivos de CloudTrail registro ... 346
CloudTrail ejemplos de entradas de registro ... 346
Usar un modelo de recomendaciones ... 352
Agregar recomendaciones a mensajes ... 352
Invocar una función de Lambda para un modelo de recomendaciones ... 355
Datos de eventos de entrada ... 355
Datos y requisitos de respuesta ... 357
Asignar una política para procesar datos de recomendaciones ... 362
Autorizar a Amazon Pinpoint a invocar una función de Lambda ... 364
Configurar un modelo de recomendaciones ... 365
Eliminar datos de un proyecto de Amazon Pinpoint ... 366
Eliminar todos los datos de un proyecto de Amazon Pinpoint ... 366
Ejemplos de código ... 368
Amazon Pinpoint ... 369
Conceptos básicos ... 370

API de SMS y voz de Amazon Pinpoint ... 462
Conceptos básicos ... 462
Seguridad ... 484
Protección de datos ... 485
Cifrado de datos ... 487
Privacidad del tráfico entre redes ... 488
Creación de un punto de conexión de VPC de interfaz para Amazon Pinpoint ... 489
Identity and Access Management ... 490
Público ... 491
Autenticación con identidades ... 491
Administración del acceso con políticas ... 493
Cómo funciona Amazon Pinpoint con IAM ... 495
Acciones de políticas de Amazon Pinpoint ... 501
Ejemplos de políticas basadas en identidades ... 533
Roles de IAM para tareas comunes ... 548
Resolución de problemas ... 565
Registro y supervisión ... 567
Validación de conformidad ... 569
Resiliencia ... 570
Seguridad de la infraestructura ... 570
Configuración y análisis de vulnerabilidades ... 571
Prácticas recomendadas de seguridad ... 572
Cuotas ... 573
Cuotas de proyectos ... 573
Cuotas de solicitudes de API ... 574
Cuotas de campaña ... 576
Cuotas de correo electrónico ... 579
Cuotas de mensajes de correo electrónico ... 579
Cuotas de los remitentes y destinatarios de correo electrónico ... 579
Cuotas de envío de correo electrónico ... 581
Cuotas de punto de enlace ... 582
Cuotas de importación de puntos de enlace ... 584
Cuotas de consumo de eventos ... 584
Cuotas de recorrido ... 586
Cuotas de Lambda ... 587
Cuotas de aprendizaje automático ... 588

Cuotas de plantillas de mensajes 589
Cuotas de notificaciones push 591
Cuotas de mensajes integrados en la aplicación 592
Cuotas de segmentos 592
Cuotas de SMS 593
Cuotas de 10DLC 593
Cuotas de voz 594
Solicitud de aumento de cuota 594
Historial de revisión 597
Actualizaciones anteriores 607

Aviso de fin de soporte: el 30 de octubre de 2026, AWS finalizará el soporte para Amazon Pinpoint. Después del 30 de octubre de 2026, ya no podrá acceder a la consola de Amazon Pinpoint ni a los recursos de Amazon Pinpoint (puntos de conexión, segmentos, campañas, recorridos y análisis). Para obtener más información, consulte Fin de soporte de Amazon Pinpoint. Nota: en lo APIs que respecta a los SMS, este cambio no afecta a los mensajes de voz, a las notificaciones push móviles, a las OTP y a la validación de números de teléfono, y son compatibles con la mensajería para el usuario AWS final.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.

¿Qué es Amazon Pinpoint?

Amazon Pinpoint es un servicio de AWS que puede utilizar para interactuar con los clientes a través de varios canales de mensajería. Puede utilizar Amazon Pinpoint para enviar notificaciones de inserción, mensajes de correo electrónico, mensajes de texto SMS o mensajes de voz.

La información de esta guía para desarrolladores está dirigida a los desarrolladores de aplicaciones. Esta guía contiene información sobre el uso de las características de Amazon Pinpoint mediante programación. También contiene información de especial interés para los desarrolladores de aplicaciones móviles, como procedimientos para integrar características de análisis y mensajería en la aplicación.

Amazon Pinpoint está disponible en varias regiones de AWS en América del Norte, Europa, Asia y Oceanía. Para obtener más información sobre las Regiones de AWS, consulte Managing Regiones de AWS en la Referencia general de Amazon Web Services. Para obtener una lista de todas las regiones en las que Amazon Pinpoint está disponible actualmente, consulte puntos de conexión y cuotas de Amazon Pinpoint y puntos de conexión de servicio de AWS en Referencia general de Amazon Web Services. Para obtener más información sobre la cantidad de zonas de disponibilidad de cada región, consulte Infraestructura global de AWS.

Para obtener más información sobre Amazon Pinpoint, consulte las siguientes guías:

- Referencia de la API de Amazon Pinpoint
- API de SMS y voz de Amazon Pinpoint
- Guía del usuario de Amazon Pinpoint

Utilizar Amazon Pinpoint para enviar mensajes a segmentos de público y analizar datos

Puede usar Amazon Pinpoint para definir segmentos de público, enviar campañas de mensajería y mensajes transaccionales y utilizar métricas para analizar el comportamiento de los usuarios.

Definición de los segmentos de público

Diríjase al público adecuado para sus mensajes definiendo segmentos de público. Un segmento designa qué usuarios reciben los mensajes que se envían desde una campaña. Puede definir

segmentos dinámicos en función de los datos notificados por la aplicación, como, por ejemplo, el sistema operativo o el tipo de dispositivo móvil. También puede importar segmentos estáticos que defina mediante otro servicio o aplicación.

Programar campañas de mensajería

Conecte con su segmento de público creando una campaña de mensajería. Una campaña envía mensajes personalizados según la programación que defina. Puede crear campañas que envíen notificaciones push en móviles, correo electrónico o mensajes SMS.

Para experimentar con estrategias de campaña alternativas, configure la campaña como una prueba A/B y analice le resultado con los análisis de Amazon Pinpoint.

Envío de mensajes de transacciones

Mantenga informados a los clientes enviando notificaciones de inserción en móviles y mensajes SMS transaccionales, como mensajes de activación de cuenta nueva, confirmaciones de pedidos y notificaciones de restablecimiento de contraseña, a usuarios concretos. Puede enviar mensajes transaccionales utilizando la API de REST de Amazon Pinpoint.

Utilizar informes de análisis y métricas

Obtenga información acerca del público y la eficacia de las campañas mediante los análisis que ofrece Amazon Pinpoint. Puede ver las tendencias del nivel de interacción, la actividad de compra y los datos demográficos de sus usuarios, entre otros datos. También puede monitorizar el tráfico de mensajes a través de métricas como el número total de mensajes que se han enviado o abierto para una campaña o aplicación. Mediante la API de Amazon Pinpoint, la aplicación puede informar de datos personalizados que Amazon Pinpoint pone a su disposición para su análisis y puede consultar los datos de análisis para determinadas métricas estándar.

Para analizar o almacenar datos de análisis fuera de Amazon Pinpoint, puede configurar Amazon Pinpoint para que transmita los datos a Amazon Kinesis.

Utilizar puntos de conexión para representar a su público en Amazon Pinpoint

En Amazon Pinpoint, cada miembro del público está representado por uno o varios puntos de conexión. Cuando se utiliza Amazon Pinpoint para enviar un mensaje, ese mensaje se dirige a los puntos de conexión que representan los miembros de su público objetivo. Cada definición de punto de conexión incluye un destino de mensaje; por ejemplo, un token de dispositivo, dirección de correo electrónico o número de teléfono. También incluye datos acerca de los usuarios y sus dispositivos. Antes de analizar, segmentar o atraer al público, el primer paso consiste en agregar puntos de conexión al proyecto de Amazon Pinpoint.

A medida que el público crece y cambia, también lo hacen los datos de punto de enlace. Para ver la información más reciente de que dispone Amazon Pinpoint acerca del público, puede buscar puntos de conexión de manera individual o puede exportar todos los puntos de conexión desde un proyecto de Amazon Pinpoint. Consultando los datos de sus puntos de conexión, puede conocer la siguiente información sobre sus usuarios:

- Su dispositivo y plataforma.
- Su zona horaria.
- Las versiones de su aplicación que están instaladas en su dispositivo.
- La ubicación de su ciudad y país.
- Cualquier otra métrica o atributo personalizado que registre.

La consola de Amazon Pinpoint también proporciona análisis de datos demográficos y atributos personalizados que se capturan en los puntos de conexión.

En los siguientes temas se explica cómo trabajar con puntos de conexión en Amazon Pinpoint. Para obtener información sobre cómo añadir puntos de conexión automáticamente mediante Android, iOS o un JavaScript cliente, consulteRegistrar puntos de conexión de Amazon Pinpoint en su aplicación.

Temas

- Agregar puntos de conexión a Amazon Pinpoint
- Asociar usuarios con puntos de conexión de Amazon Pinpoint
- Agregar un lote de puntos de conexión a Amazon Pinpoint

- Importar puntos de conexión a Amazon Pinpoint
- Exportar puntos de conexión de Amazon Pinpoint a buckets de Amazon S3
- Buscar puntos de conexión en un proyecto de Amazon Pinpoint
- Lista de ID de puntos de conexión con Amazon Pinpoint
- Administrar el número máximo de puntos de conexión en Amazon Pinpoint
- Eliminar puntos de conexión de Amazon Pinpoint mediante programación

Agregar puntos de conexión a Amazon Pinpoint

Un punto de conexión representa un destino al que puede enviar mensajes, como un dispositivo móvil, número de teléfono o dirección de correo electrónico. Antes de poder enviar un mensaje a un miembro de su público, debe definir uno o varios puntos de enlace para esa persona.

A medida que agrega puntos de conexión a Amazon Pinpoint, aumenta como repositorio de datos de público. Estos datos se componen de:

- Los puntos de conexión que agrega o actualiza utilizando la API de Amazon Pinpoint.
- Los puntos de enlace que su código de cliente añade o actualiza a medida que los usuarios llegan a la aplicación.

Al definir un punto de enlace, debe especificar el canal y la dirección. El canal es el tipo de plataforma que se utiliza para enviar un mensaje al punto de enlace. Entre los ejemplos de canales se incluyen un servicio de notificación de inserción, SMS o correo electrónico. La dirección especifica dónde enviar mensajes al punto de enlace, como, por ejemplo, un token de dispositivo, número de teléfono o dirección de correo electrónico.

Para añadir más detalles sobre el público, puede enriquecer los puntos de enlace con atributos personalizados y estándar. Estos atributos incluyen datos acerca de los usuarios, sus preferencias, sus dispositivos, las versiones del cliente que utilizan o sus ubicaciones. Al añadir este tipo de datos a los puntos de enlace, puede:

- Ver gráficos sobre el público en la consola de Amazon Pinpoint.
- Segmentar el público en función de los atributos de puntos de enlace para que pueda enviar sus mensajes al público de destino correcto.
- Personalizar sus mensajes incorporando variables de mensajes que se sustituyen por valores de atributos de punto de enlace.

Una aplicación móvil o de cliente de JavaScript registra los puntos de conexión automáticamente si integra Amazon Pinpoint mediante los AWS Mobile SDK o la biblioteca JavaScript de AWS Amplify. El cliente registra un punto de enlace para cada usuario nuevo y actualiza los puntos de enlace para los usuarios recurrentes. Para registrar puntos de enlace desde un cliente móvil o de JavaScript, consulte Registrar puntos de conexión de Amazon Pinpoint en su aplicación.

Ejemplos

Los siguientes ejemplos muestran cómo agregar un punto de conexión a un proyecto de Amazon Pinpoint. El punto de enlace representa a un miembro del público que vive en Seattle y utiliza un iPhone. Se puede enviar un mensaje a esta persona a través de Apple Push Notification Service (APN). La dirección del punto de enlace es token de dispositivo que proporciona APN.

AWS CLI

Puede utilizar Amazon Pinpoint ejecutando comandos con la AWS CLI.

Example Comando Update Endpoint

Para añadir o actualizar un punto de enlace, utilice el comando update-endpoint:

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

Donde:

- application-id es el ID del proyecto de Amazon Pinpoint en el que agrega o actualiza un punto de conexión.
- example-endpoint es el ID que está asignando a un nuevo punto de enlace o es el ID de un punto de enlace existente que está actualizando.
- endpoint-request-file.json es la ruta de archivo a un archivo JSON local que contiene la entrada para el parámetro `--endpoint-request`.

Example Archivo de solicitud de punto de enlace

El comando `update-endpoint` de ejemplo utiliza un archivo JSON como argumento para el parámetro `--endpoint-request`. Este archivo contiene una definición de punto de enlace similar a la siguiente:

```
{
  "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
  }
}
```

Para los atributos que puede utilizar para definir un punto de conexión, consulte el esquema EndpointRequest en la Referencia de la API de Amazon Pinpoint.

AWS SDK para Java

Puede usar la API de Amazon Pinpoint en las aplicaciones Java mediante el cliente que proporciona AWS SDK para Java.

Example Código

Para añadir un punto de conexión, inicialice un objeto `EndpointRequest` y transfiéralo al método `updateEndpoint` del cliente de `AmazonPinpoint`:

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

public class AddExampleEndpoint {

 public static void main(String[] args) {

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

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

  String applicationId = args[0];

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

  // Initializes an endpoint definition with channel type and address.
  EndpointRequest wangXiulansIphoneEndpoint = new EndpointRequest()
    .withChannelType(ChannelType.APNS)
    .withAddress(deviceToken);

  // Adds custom attributes to the endpoint.
  wangXiulansIphoneEndpoint.addAttributesEntry("interests", Arrays.asList(
```

```
    "technology",
    "music",
    "travel"));

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

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

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

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

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

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

 }
}
```

HTTP

Puede utilizar Amazon Pinpoint realizando solicitudes HTTP directamente a la API de REST.

Example Solicitud PUT al punto de enlace

Para añadir un punto de enlace, emita una solicitud PUT al recurso Punto de enlace en el URI siguiente:

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

Donde:

- application-id es el ID del proyecto de Amazon Pinpoint en el que agrega o actualiza un punto de conexión.
- endpoint-id es el ID que asigna a un nuevo punto de enlace o el ID de un punto de enlace existente que está actualizando.

En su solicitud, incluya los encabezados necesarios y proporcione el JSON EndpointRequest como cuerpo:

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

{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
```

```
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

Si la solicitud se realiza correctamente, recibirá una respuesta similar a la siguiente:

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

Información relacionada

Para obtener más información acerca del recurso de punto de conexión en la API de Amazon Pinpoint, incluidos los métodos HTTP compatibles y los parámetros de solicitud, consulte Punto de conexión en la Referencia de la API de Amazon Pinpoint.

Para obtener más información sobre la personalización de mensajes con variables, consulte Variables del mensaje en la Guía del usuario de Amazon Pinpoint.

Para obtener información sobre las cuotas que se aplican a los puntos de enlace, como el número de atributos que puede asignar, consulte the section called “Cuotas de punto de enlace”.

Asociar usuarios con puntos de conexión de Amazon Pinpoint

Un punto de enlace puede incluir atributos que definen un usuario, que representa una persona de su público. Por ejemplo, un usuario puede representar a alguien que ha instalado su aplicación móvil o alguien que tenga una cuenta en su sitio web.

Puede definir un usuario especificando un ID de usuario único y, opcionalmente, atributos de usuario personalizados. Si alguien utiliza la aplicación en varios dispositivos o si se pueden enviar mensajes a esa persona en varias direcciones, puede asignar el mismo ID de usuario a varios puntos de enlace. En este caso, Amazon Pinpoint sincroniza los atributos de usuario entre los puntos de conexión. Por lo tanto, si agrega un atributo de usuario a un punto de conexión, Amazon Pinpoint agrega dicho atributo a cada punto de conexión que incluye el mismo ID de usuario.

Puede añadir atributos de usuario para realizar un seguimiento de los datos que se aplican a una persona y no varía en función del dispositivo que la persona está utilizando. Por ejemplo, puede añadir atributos para el nombre de una persona, la edad o el estado de la cuenta.

> **Tip**
>
> Si la aplicación utiliza grupos de usuarios de Amazon Cognito para gestionar la autenticación del usuario, Amazon Cognito puede agregar los ID de usuario y atributos a los puntos de conexión de forma automática. Para el valor de ID de usuario de punto de conexión, Amazon Cognito asigna el valor `sub` que se ha asignado al usuario en el grupo de usuarios. Para obtener información sobre cómo agregar usuarios con Amazon Cognito, consulte Uso del análisis de Amazon Pinpoint con grupos de usuarios de Amazon Cognito en la Guía para desarrolladores de Amazon Cognito.

Después de añadir definiciones de usuario a los puntos de enlace, tiene más opciones acerca de cómo segmentar su público. Puede definir un segmento en función de atributos de usuario o puede definir un segmento importando una lista de ID de usuario. Cuando envía un mensaje a un segmento que se basa en usuarios, los destinos posibles incluyen todos los puntos de enlace asociados con cada usuario en el segmento.

También tiene más opciones acerca de cómo enviar mensajes a su público. Puede utilizar una campaña para enviar mensajes a un segmento de usuarios o puede enviar un mensaje directamente

a una lista de ID de usuario. Para personalizar su mensaje, puede incluir variables de mensajes que se sustituyen por valores de atributos de usuario.

Ejemplos

Los siguientes ejemplos muestran cómo añadir una definición de usuario a un punto de enlace.

AWS CLI

Puede utilizar Amazon Pinpoint ejecutando comandos con la AWS CLI.

Example Comando Update Endpoint

Para añadir un usuario a un punto de enlace, utilice el comando update-endpoint. Para el parámetro `--endpoint-request`, puede definir un nuevo punto de enlace, que puede incluir un usuario. O bien, para actualizar un punto de enlace existente, puede proporcionar solo los atributos que desee cambiar. En el siguiente ejemplo se añade un usuario a un punto de enlace existente proporcionando solo los atributos de usuario:

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

Donde:

- *`application-id`* es el ID del proyecto de Amazon Pinpoint en el que agrega o actualiza un punto de conexión.
- *`endpoint-id`* es el ID que asigna a un nuevo punto de enlace o el ID de un punto de enlace existente que está actualizando.
- *`endpoint-request-file.json`* es la ruta de archivo a un archivo JSON local que contiene la entrada para el parámetro `--endpoint-request`.

Example Archivo de solicitud de punto de enlace

El comando `update-endpoint` de ejemplo utiliza un archivo JSON como argumento para el parámetro `--endpoint-request`. Este archivo contiene una definición de usuario similar a la siguiente:

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

Para los atributos que puede utilizar para definir un usuario, consulte el objeto `User` en el esquema de EndpointRequest en la Referencia de la API de Amazon Pinpoint.

AWS SDK para Java

Puede usar la API de Amazon Pinpoint en las aplicaciones Java mediante el cliente que proporciona AWS SDK para Java.

Example Código

Para agregar un usuario a un punto de conexión, inicialice un objeto EndpointRequest y transfiéralo al método `updateEndpoint` del cliente de `AmazonPinpoint`. Puede utilizar este objeto para definir un nuevo punto de enlace, que puede incluir un usuario. O bien, para actualizar un punto de enlace existente, puede actualizar solo las propiedades que desee cambiar. En el siguiente ejemplo se añade un usuario a un punto de enlace existente añadiendo un objeto EndpointUser al objeto 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);
        }
    }
```

Para ver el ejemplo completo de SDK, consulte AddExampleUser.java en GitHub.

HTTP

Puede utilizar Amazon Pinpoint realizando solicitudes HTTP directamente a la API de REST.

Example Solicitud PUT al punto de enlace con definición de usuario

Para añadir un usuario a un punto de enlace, emita una solicitud PUT al recurso Endpoint en el URI siguiente:

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

Donde:

- *application-id* es el ID del proyecto de Amazon Pinpoint en el que agrega o actualiza un punto de conexión.
- *endpoint-id* es el ID que asigna a un nuevo punto de enlace o el ID de un punto de enlace existente que está actualizando.

En su solicitud, incluya los encabezados necesarios y proporcione el JSON EndpointRequest como cuerpo. El cuerpo de la solicitud puede definir un nuevo punto de enlace, que puede incluir un usuario. O bien, para actualizar un punto de enlace existente, puede proporcionar solo los atributos que desee cambiar. En el siguiente ejemplo se añade un usuario a un punto de enlace existente proporcionando solo los atributos de usuario:

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

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

Si la solicitud se realiza correctamente, recibirá una respuesta similar a la siguiente:

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

Información relacionada

Para obtener más información acerca del recurso de punto de conexión en la API de Amazon Pinpoint, incluidos los métodos HTTP compatibles y los parámetros de solicitud, consulte Punto de conexión en la Referencia de la API de Amazon Pinpoint.

Para obtener más información sobre la personalización de mensajes con variables, consulte Variables del mensaje en la Guía del usuario de Amazon Pinpoint.

Para obtener información sobre cómo definir un segmento mediante la importación de una lista de ID de usuario, consulte Importación de segmentos en la Guía del usuario de Amazon Pinpoint.

Para obtener información sobre cómo enviar un mensaje directo a un máximo de 100 ID de usuario, consulte Mensajes de usuarios en la Referencia de la API de Amazon Pinpoint.

Para obtener información sobre las cuotas que se aplican a los puntos de enlace, incluido el número de atributos de usuario que puede asignar, consulte the section called “Cuotas de punto de enlace”.

Agregar un lote de puntos de conexión a Amazon Pinpoint

Puede añadir o actualizar varios puntos de enlace en una sola operación proporcionando los puntos de enlace en lotes. Cada solicitud de lote puede incluir hasta 100 definiciones de punto de enlace.

Si desea añadir o actualizar más de 100 puntos de enlace en una sola operación, consulte Importar puntos de conexión a Amazon Pinpoint en su lugar.

Ejemplos

Los siguientes ejemplos muestran cómo añadir dos puntos de enlace a la vez incluyendo los puntos de enlace en una solicitud de lote.

AWS CLI

Puede utilizar Amazon Pinpoint ejecutando comandos con la AWS CLI.

Example Comando Update Endpoints Batch

Para enviar una solicitud de lote de puntos de enlace, utilice el comando `update-endpoints-batch`:

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

Donde:

- *`application-id`*es el ID del proyecto de Amazon Pinpoint en el que va a añadir o actualizar los puntos de enlace.
- *`endpoint_batch_request_file.json`*es la ruta del archivo a un archivo JSON local que contiene la entrada del parámetro. `--endpoint-batch-request`

Example Archivo de solicitud de lote de puntos de enlace

El comando `update-endpoints-batch` de ejemplo utiliza un archivo JSON como argumento para el parámetro `--endpoint-request`. Este archivo contiene un lote de definiciones de punto de enlace similar al siguiente:

```
{
```

```
    "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"
```

```
                }
            }
        }
    ]
}
```

Para ver los atributos que puede usar para definir un lote de puntos de enlace, consulte el EndpointBatchRequestesquema en la referencia de la API de Amazon Pinpoint.

AWS SDK para Java

Puede usar la API de Amazon Pinpoint en las aplicaciones Java mediante el cliente que proporciona AWS SDK para Java.

Example Código

Para enviar una solicitud de lote de puntos de conexión, inicialice un objeto `EndpointBatchRequest` y transfiéralo al método `updateEndpointsBatch` del cliente de `AmazonPinpoint`. En el siguiente ejemplo, se rellena un objeto `EndpointBatchRequest` con dos objetos `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);
                }
        }
}
```

Para ver el ejemplo completo del SDK, consulte AddExampleEndpoints.java en. GitHub

HTTP

Puede utilizar Amazon Pinpoint realizando solicitudes HTTP directamente a la API de REST.

Example Solicitud PUT a puntos de enlace

Para enviar una solicitud de lote de puntos de enlace, emita una solicitud PUT al recurso Puntos de enlace en el URI siguiente:

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

Dónde *application-id* está el ID del proyecto de Amazon Pinpoint en el que va a añadir o actualizar los puntos de enlace.

En su solicitud, incluya los encabezados necesarios y proporcione el EndpointBatchRequestJSON como cuerpo:

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

{
    "Item": [
        {
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
            },
```

```
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
            },
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
                }
            }
        }
    ]
}
```

Si la solicitud se realiza correctamente, recibirá una respuesta similar a la siguiente:

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

Información relacionada

Para obtener más información acerca del recurso de punto de conexión en la API de Amazon Pinpoint, incluidos los métodos HTTP compatibles y los parámetros de solicitud, consulte Punto de conexión en la Referencia de la API de Amazon Pinpoint.

Importar puntos de conexión a Amazon Pinpoint

Puede agregar o actualizar puntos de conexión a gran escala importándolos desde un bucket de Amazon S3. La importación de puntos de conexión resulta útil si tiene registros sobre el público fuera de Amazon Pinpoint y desea agregar esta información a un proyecto de Amazon Pinpoint. En este caso, debería hacer lo siguiente:

1. Crear definiciones de punto de enlace basadas en sus propios datos de público.
2. Guarde estas definiciones de punto de conexión en uno o más archivos y cargue los archivos en un bucket de Amazon S3.
3. Agregue los puntos de conexión al proyecto de Amazon Pinpoint importándolos desde el bucket.

Cada trabajo de importación puede transferir hasta 1 GB de datos. En un trabajo típico, en el que cada punto de enlace tiene 4 KB o menos, puede importar en torno a 250 000 puntos de enlace. Puede ejecutar hasta dos trabajos de importación simultáneos por AWS cuenta. Si necesita más ancho de banda para sus trabajos de importación, puede enviar una solicitud de aumento de la cuota de servicio a Soporte. Para obtener más información, consulte Solicitud de aumento de cuota.

Antes de empezar

Para poder importar los puntos de enlace, son precisos los siguientes recursos en su cuenta de AWS :

- Un bucket de Amazon S3. Para crear un bucket, consulte Crear un bucket en la Guía del usuario de Amazon Simple Storage Service.
- Un rol AWS Identity and Access Management (IAM) que otorga permisos de lectura a Amazon Pinpoint para su bucket de Amazon S3. Para crear la función, consulte Rol de IAM para importar puntos de conexión o segmentos.

Ejemplos

Los siguientes ejemplos muestran cómo agregar definiciones de punto de conexión al bucket de Amazon S3 y, a continuación, importar dichos puntos de conexión en un proyecto de Amazon Pinpoint.

Archivos con definiciones de puntos de enlace

Los archivos que agrega al bucket de Amazon S3 pueden contener definiciones de punto de conexión en formato CSV o JSON delimitado por saltos de línea. Para ver los atributos que puede usar para definir sus puntos de enlace, consulte el esquema EndpointRequestJSON en la referencia de la API de Amazon Pinpoint.

CSV

Puede importar puntos de enlace definidos en un archivo CSV, como se muestra en el siguiente ejemplo:

```
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 primera línea es el encabezado, que contiene los atributos del punto de enlace. Especifique los atributos anidados utilizando la notación de puntos, como en `Location.Country`.

En las líneas siguientes se definen los puntos de enlace mediante la inclusión de valores para cada uno de los atributos en el encabezado.

Para incluir una coma o comillas dobles en un valor, escriba el valor entre comillas dobles, por ejemplo, `"aaa,bbb"`.

No se admiten saltos de línea dentro de un valor de CSV.

JSON

Puede importar puntos de enlace definidos en un archivo JSON delimitado por saltos de línea, como se muestra en el siguiente ejemplo:

```
{"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"}}
```

En este formato, cada línea es un objeto JSON completo que contiene una definición de punto de enlace individual.

Solicitudes de trabajo de importación

Los siguientes ejemplos muestran cómo agregar definiciones de puntos de conexión a Amazon S3 cargando un archivo local en un bucket. A continuación, los ejemplos importan las definiciones de puntos de conexión en un proyecto de Amazon Pinpoint.

AWS CLI

Puede utilizar Amazon Pinpoint ejecutando comandos con la AWS CLI.

Example Comando CP de S3

Para cargar un archivo local a un bucket de Amazon S3, utilice el comando `cp` de Amazon S3:

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

Donde:

- ./endpoints-file es la ruta de archivo a un archivo local que contiene definiciones de punto de enlace.
- bucket-name/prefix es el nombre del bucket de Amazon S3 y, opcionalmente, un prefijo que le ayuda a organizar los objetos en el bucket de forma jerárquica. Por ejemplo, un prefijo útil puede ser `pinpoint/imports/endpoints/`.

Example Comando Create Import Job

Para importar las definiciones de puntos de conexión desde un bucket de Amazon S3, utilice el comando `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
```

Donde:

- application-id es el ID del proyecto de Amazon Pinpoint para el que está importando los puntos de conexión.
- bucket: name/prefix/key es la ubicación en Amazon S3 que contiene uno o más objetos para importar. La ubicación puede finalizar con la clave para un objeto individual o puede terminar con un prefijo que califica varios objetos.
- iam-import-role-arnes el nombre de recurso de Amazon (ARN) de una función de IAM que otorga a Amazon Pinpoint acceso de lectura al bucket.
- format puede ser `JSON` o bien `CSV`, en función del formato que haya utilizado para definir los puntos de enlace. Si la ubicación de Amazon S3 incluye varios objetos de formatos mixtos, Amazon Pinpoint solo importa los objetos que coinciden con el formato especificado.
- RegisterEndpoints puede ser `true` o `false`. Si se establece en verdadero, el trabajo de importación registra los puntos de conexión con Amazon Pinpoint, cuando se importan las definiciones de los puntos de conexión.

  RegisterEndpoints y combinaciones DefineSegments

RegisterEndpoints	DefineSegments	Description (Descripción)
true	true	Amazon Pinpoint importará los puntos de conexión y creará un segmento que los contenga.
true	false	Amazon Pinpoint importará los puntos de conexión y no creará un segmento.
false	true	Amazon Pinpoint importará los puntos de conexión y

RegisterEndpoints	DefineSegments	Description (Descripción)
		creará un segmento que los contenga. Los puntos de conexión no se guardarán ni sobrescribirán los puntos de conexión existentes.
false	false	Amazon Pinpoint rechazará esta solicitud.

La respuesta incluye detalles acerca del trabajo de importación:

```
{
    "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 respuesta proporciona el ID de trabajo con el atributo `Id`. Puede utilizar este ID para comprobar el estado actual del trabajo de importación.

Example Comando Get Import Job

Para comprobar el estado actual de un trabajo de importación, utilice el comando `get-import-job`:

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

```
> --job-id job-id
```

Donde:

- application-id es el ID del proyecto de Amazon Pinpoint para el que se inició el trabajo de importación.
- job-id es el ID del trabajo de importación que está comprobando.

La respuesta a este comando ofrece el estado actual del trabajo de importación:

```
{
    "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 respuesta proporciona el estado de trabajo con el atributo `JobStatus`.

AWS SDK para Java

Puede usar la API de Amazon Pinpoint en las aplicaciones Java mediante el cliente que proporciona AWS SDK para Java.

Example Código

Para cargar un archivo con definiciones de puntos de conexión en Amazon S3, utilice el método `putObject` del cliente de `AmazonS3`.

Para importar los puntos de conexión en un proyecto de Amazon Pinpoint, inicialice un objeto `CreateImportJobRequest`. A continuación, transfiera este objeto al método `createImportJob` del cliente de `AmazonPinpoint`.

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

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

public class ImportEndpoints {

    public static void main(String[] args) {

        final String USAGE = "\n" +
                "ImportEndpoints - Adds endpoints to an Amazon Pinpoint application
 by: \n" +
                "1.) Uploading the endpoint definitions to an Amazon S3 bucket. \n"
 +
                "2.) Importing the endpoint definitions from the bucket to an Amazon
 Pinpoint " +
                "application.\n\n" +
```

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

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

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

        Path endpointsFilePath = Paths.get(endpointsFileLocation);
        File endpointsFile = new
File(endpointsFilePath.toAbsolutePath().toString());
        uploadToS3(endpointsFile, s3BucketName);

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

    }

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

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

        // Checks whether the specified bucket exists. If not, attempts to create
one.
        if (!s3.doesBucketExistV2(s3BucketName)) {
```

```
            try {
                s3.createBucket(s3BucketName);
                System.out.format("Created S3 bucket %s.\n", s3BucketName);
            } catch (AmazonS3Exception e) {
                System.err.println(e.getErrorMessage());
                System.exit(1);
            }
        }

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

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

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

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

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

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

        try {

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

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

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

            if (jobStatus.equals("COMPLETED")) {
                System.out.println("Finished importing endpoints.");
            } else {
                System.err.println("Failed to import endpoints.");
                System.exit(1);
            }

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

```
            }

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

    }

}
```

HTTP

Puede utilizar Amazon Pinpoint realizando solicitudes HTTP directamente a la API de REST.

Example Solicitud PUT Object de S3

Para agregar las definiciones de puntos de conexión a un bucket, utilice la operación PUT object de Amazon S3 y proporcione las definiciones del punto de conexión como cuerpo:

```
PUT /prefix/key HTTP/1.1
Content-Type: text/plain
Accept: application/json
Host: bucket-name.s3.amazonaws.com
X-Amz-Content-Sha256:
 c430dc094b0cec2905bc88d96314914d058534b14e2bc6107faa9daa12fdff2d
X-Amz-Date: 20180605T184132Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/s3/aws4_request, SignedHeaders=accept;cache-control;content-
length;content-type;host;postman-token;x-amz-content-sha256;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

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

Donde:

- /prefix/key es el prefijo y nombre de clave para el objeto que contendrá las definiciones de puntos de enlace después de la carga. Puede utilizar el prefijo para organizar sus objetos jerárquicamente. Por ejemplo, un prefijo útil puede ser `pinpoint/imports/endpoints/`.
- bucket-name es el nombre del bucket de Amazon S3 al que está agregando las definiciones de puntos de conexión.

Example Solicitud POST de trabajo de importación

Para importar definiciones de puntos de conexión desde un bucket de Amazon S3, emita una solicitud POST al recurso Trabajos de importación. En su solicitud, incluya los encabezados necesarios y proporcione el ImportJobRequestJSON como cuerpo:

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

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

Donde:

- application-id es el ID del proyecto de Amazon Pinpoint para el que está importando los puntos de conexión.
- bucket: name/prefix/key es la ubicación en Amazon S3 que contiene uno o más objetos para importar. La ubicación puede finalizar con la clave para un objeto individual o puede terminar con un prefijo que califica varios objetos.
- iam-import-role-arnes el nombre de recurso de Amazon (ARN) de una función de IAM que otorga a Amazon Pinpoint acceso de lectura al bucket.

- format puede ser `JSON` o bien `CSV`, en función del formato que haya utilizado para definir los puntos de enlace. Si la ubicación de Amazon S3 incluye varios archivos de formatos mixtos, Amazon Pinpoint solo importa los archivos que coincidan con el formato especificado.

Si la solicitud se realiza correctamente, recibirá una respuesta similar a la siguiente:

```
{
    "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 respuesta proporciona el ID de trabajo con el atributo `Id`. Puede utilizar este ID para comprobar el estado actual del trabajo de importación.

Example Solicitud GET de trabajo de importación

Para comprobar el estado actual de un trabajo de importación, emita una solicitud `GET` al recurso Trabajo de importación:

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

Donde:

- application_id es el ID del proyecto de Amazon Pinpoint para el que se inició el trabajo de importación.
- job_id es el ID del trabajo de importación que está comprobando.

Si la solicitud se realiza correctamente, recibirá una respuesta similar a la siguiente:

```
{
    "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 respuesta proporciona el estado de trabajo con el atributo `JobStatus`.

Información relacionada

Para obtener más información acerca del recurso Trabajos de importación en la API de Amazon Pinpoint, incluidos los métodos HTTP compatibles y los parámetros de solicitud, consulte Trabajos de importación en la Referencia de la API de Amazon Pinpoint.

Exportar puntos de conexión de Amazon Pinpoint a buckets de Amazon S3

Para obtener toda la información que Amazon Pinpoint tiene acerca del público, puede exportar las definiciones de punto de conexión que pertenecen a un proyecto. Al exportar, Amazon Pinpoint coloca las definiciones de punto de conexión en un bucket de Amazon S3 que especifique. La exportación de puntos de enlace resulta útil cuando desea:

- Consulte los datos más recientes sobre puntos de conexión nuevos y existentes que la aplicación cliente registró con Amazon Pinpoint.
- Sincronizar los datos de puntos de conexión en Amazon Pinpoint con el propio sistema de administración de las relaciones con el cliente (CRM).
- Analizar o crear informes sobre datos de los clientes.

> **Note**
>
> El contenido entregado a los buckets de Amazon S3 puede contener contenido del cliente. Si necesita eliminar los datos de punto de conexión que ha exportado a un bucket de Amazon S3, deberá hacerlo en Amazon S3. Para obtener más información sobre cómo eliminar información confidencial, consulte cómo vaciar un bucket de S3 o cómo eliminar un bucket de S3.

Antes de empezar

Para poder exportar los puntos de enlace, son precisos los siguientes recursos en su cuenta de AWS:

- Un bucket de Amazon S3. Para crear un bucket, consulte Crear un bucket en la Guía del usuario de Amazon Simple Storage Service.
- Un rol de AWS Identity and Access Management (IAM) que conceda a Amazon Pinpoint permisos de escritura para el bucket de Amazon S3. Para crear la función, consulte Rol de IAM para exportar puntos de conexión o segmentos.

Ejemplos

Los siguientes ejemplos muestran cómo exportar puntos de conexión desde un proyecto de Amazon Pinpoint y, a continuación, descargar esos puntos de conexión desde el bucket de Amazon S3.

AWS CLI

Puede utilizar Amazon Pinpoint ejecutando comandos con la AWS CLI.

Example Comando Create Export Job

Para exportar los puntos de conexión en el proyecto de Amazon Pinpoint, utilice el comando `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
```

Donde:

- *application-id* es el ID del proyecto de Amazon Pinpoint que contiene los puntos de conexión.
- *bucket-name/prefix/* es el nombre del bucket de Amazon S3 y, opcionalmente, un prefijo que le ayuda a organizar los objetos en el bucket de forma jerárquica. Por ejemplo, un prefijo útil puede ser `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* es el nombre de recurso de Amazon (ARN) de un rol de IAM que concede a Amazon Pinpoint acceso de escritura al bucket.

La respuesta a este comando ofrece información detallada sobre el trabajo de exportación:

```
{
    "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 respuesta proporciona el ID de trabajo con el atributo `Id`. Puede utilizar este ID para comprobar el estado actual del trabajo de exportación.

Example Comando Get Export Job

Para comprobar el estado actual de un trabajo de exportación, utilice el comando `get-export-job`:

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

Donde:

- *`application-id`* es el ID del proyecto de Amazon Pinpoint desde el que exportó los puntos de conexión.
- *`job-id`* es el ID del trabajo que está comprobando.

La respuesta a este comando ofrece el estado actual del trabajo de exportación:

```
{
    "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 respuesta proporciona el estado de trabajo con el atributo `JobStatus`. Cuando el valor de estado del trabajo es `COMPLETED`, puede obtener los puntos de conexión exportados desde el bucket de Amazon S3.

Example Comando CP de S3

Para descargar los puntos de conexión exportados, utilice el comando cp de Amazon S3:

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

Donde:

- *`bucket-name/prefix/key`* es la ubicación del archivo .gz que Amazon Pinpoint agregó al bucket cuando exportó los puntos de conexión. Este archivo contiene las definiciones de punto de enlace exportadas. Por ejemplo, en la URL `https://PINPOINT-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/Exports/example.csv`, `PINPOINT-EXAMPLE-BUCKET` es el nombre del bucket y `Exports/example.csv` es la clave. Para obtener más información sobre claves, consulte Claves en la Guía del usuario de Amazon S3.
- */local/directory/* es la ruta del archivo al directorio local donde desea descargar los puntos de enlace.

AWS SDK para Java

Puede usar la API de Amazon Pinpoint en las aplicaciones Java mediante el cliente que proporciona AWS SDK para Java.

Example Código

Para exportar puntos de conexión desde un proyecto de Amazon Pinpoint, inicialice un objeto `CreateExportJobRequest`. A continuación, transfiera este objeto al método `createExportJob` del cliente `AmazonPinpoint`.

Para descargar los puntos de conexión exportados desde Amazon Pinpoint, utilice el método `getObject` del cliente de `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);
        }
    }
```

```
}
```

Para ver el ejemplo de SDK completo, consulte ExportEndpoints.java en GitHub.

HTTP

Puede utilizar Amazon Pinpoint realizando solicitudes HTTP directamente a la API de REST.

Example Solicitud POST de trabajo de exportación

Para exportar los puntos de conexión en el proyecto de Amazon Pinpoint, emita una solicitud POST al recurso Trabajos de exportación:

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

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

Donde:

- *application-id* es el ID del proyecto de Amazon Pinpoint que contiene los puntos de conexión.
- *bucket-name/prefix* es el nombre del bucket de Amazon S3 y, opcionalmente, un prefijo que le ayuda a organizar los objetos en el bucket de forma jerárquica. Por ejemplo, un prefijo útil puede ser `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* es el nombre de recurso de Amazon (ARN) de un rol de IAM que concede a Amazon Pinpoint acceso de escritura al bucket.

La respuesta a esta solicitud ofrece información detallada sobre el trabajo de exportación:

```
{
    "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 respuesta proporciona el ID de trabajo con el atributo `Id`. Puede utilizar este ID para comprobar el estado actual del trabajo de exportación.

Example Solicitud GET de trabajo de exportación

Para comprobar el estado actual de un trabajo de exportación, emita una solicitud `GET` al recurso Trabajo de exportación:

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

Donde:

- *application-id* es el ID del proyecto de Amazon Pinpoint desde el que exportó los puntos de conexión.
- *job-id* es el ID del trabajo que está comprobando.

La respuesta a esta solicitud ofrece el estado actual del trabajo de exportación:

```
{
    "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 respuesta proporciona el estado de trabajo con el atributo `JobStatus`. Cuando el valor de estado del trabajo es `COMPLETED`, puede obtener los puntos de conexión exportados desde el bucket de Amazon S3.

Información relacionada

Para buscar el ID de un punto de conexión específico, debe determinar a qué segmento pertenece el punto de conexión y, a continuación, exportar el segmento desde Amazon Pinpoint. Los datos exportados incluyen el ID del punto de enlace para cada punto de enlace. Puede exportar un segmento a un archivo mediante la consola de Amazon Pinpoint. Para obtener más información sobre la exportación de segmentos, consulte Exporting Segments en la Guía del usuario de Amazon Pinpoint.

Para obtener más información acerca del recurso Trabajos de exportación en la API de Amazon Pinpoint, incluidos los métodos HTTP compatibles y los parámetros de solicitud, consulte Trabajos de exportación en la Referencia de la API de Amazon Pinpoint.

Buscar puntos de conexión en un proyecto de Amazon Pinpoint

Puede buscar los detalles de cualquier punto de conexión individual que se agregó a un proyecto de Amazon Pinpoint. Estos detalles pueden incluir la dirección de destino de sus mensajes, el canal de mensajería, datos acerca del dispositivo del usuario, datos acerca de la ubicación del usuario y cualquier otro atributo personalizado que registre en sus puntos de enlace.

Para buscar un punto de enlace, necesita el ID del punto de enlace. Si no conoce el ID, puede obtener los datos del punto de enlace exportándolo en su lugar. Para exportar puntos de enlace,

consulte the section called “Exportar puntos de conexión de Amazon Pinpoint a buckets de Amazon S3”.

Ejemplos

En los siguientes ejemplos puede ver cómo buscar un punto de enlace individual especificando su ID.

AWS CLI

Puede utilizar Amazon Pinpoint ejecutando comandos con la AWS CLI.

Example Comando Get Endpoint

Para buscar un punto de enlace, utilice el comando `get-endpoint`:

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

Donde:

- *application-id* es el ID del proyecto de Amazon Pinpoint que contiene el punto de conexión.
- *endpoint-id* es el ID del punto de enlace que está buscando.

La respuesta a este comando es la definición JSON del punto de enlace, como en el ejemplo siguiente:

```
{
    "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 para Java

Puede usar la API de Amazon Pinpoint en las aplicaciones Java mediante el cliente que proporciona AWS SDK para Java.

Example Código

Para buscar un punto de conexión, inicialice un objeto `GetEndpointRequest`. A continuación, transfiera este objeto al método `getEndpoint` del cliente de`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");
    }
}
```

Para imprimir los datos de punto de enlace en un formato legible, en este ejemplo se utiliza la biblioteca GSON de Google para convertir el objeto `EndpointResponse` en una cadena JSON.

HTTP

Puede utilizar Amazon Pinpoint realizando solicitudes HTTP directamente a la API de REST.

Example Solicitud GET al punto de enlace

Para buscar un punto de enlace, emita una solicitud `GET` al recurso Punto de enlace:

```
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
```

Donde:

- *`application-id`* es el ID del proyecto de Amazon Pinpoint que contiene el punto de conexión.
- *`endpoint-id`* es el ID del punto de enlace que está buscando.

La respuesta a esta solicitud es la definición JSON del punto de enlace, como en el ejemplo siguiente:

```
{
    "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"
}
```

Información relacionada

Para obtener más información sobre el recurso de punto de conexión de la API de Amazon Pinpoint, consulte Punto de conexión en la referencia de la API de Amazon Pinpoint.

Lista de ID de puntos de conexión con Amazon Pinpoint

Para actualizar o eliminar un punto de enlace, necesita el ID del punto de enlace. Por lo tanto, si desea realizar estas operaciones en todos los puntos de conexión en un proyecto de Amazon

Pinpoint, el primer paso consiste en mostrar todos los ID de los puntos de conexión que pertenezcan a ese proyecto. A continuación, puede iterar en estos ID para, por ejemplo, añadir un atributo globalmente o eliminar todos los puntos de enlace del proyecto.

En el siguiente ejemplo se utiliza la AWS SDK para Java y hace lo siguiente:

1. Llama al método `exportEndpointsToS3` de ejemplo desde el código de ejemplo de Export endpoints from Amazon Pinpoint. Este método exporta las definiciones de punto de conexión desde un proyecto de Amazon Pinpoint. Las definiciones de punto de conexión se agregan como archivos gzip a un bucket de Amazon S3.
2. Descarga los archivos gzip exportados.
3. Lee los archivos gzip y obtiene el ID del punto de enlace desde la definición JSON de cada punto de enlace.
4. Imprime los ID de punto de enlace a la consola.
5. Se realiza una limpieza eliminando los archivos que Amazon Pinpoint ha agregado a 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);
        }
    }
}
```

Para ver el ejemplo completo de SDK, consulte ListEndpointIs.java en GitHub.

Administrar el número máximo de puntos de conexión en Amazon Pinpoint

Cada miembro de su audiencia puede tener un máximo de 15 puntos finales asociados a su audiencia UserId, consulteCuotas de punto de enlace. Si intentas añadir un decimosexto punto final, dependiendo del mismo ChannelType, conseguirás BadRequestExceptiono tendrás éxito eliminando el punto final con el más antiguo EffectiveDate.

Agregar un punto de conexión número 16

- Si el nuevo tipo de canal para el terminal es SMS, PUSH, VOICE, EMAIL, CUSTOM o IN_APP, BadRequestExceptionse devuelve porque el miembro de la audiencia se encuentra en su número máximo de puntos finales. Debe eliminar un punto de conexión asociado al miembro de la audiencia y volver a intentarlo, consulte Eliminar puntos de conexión de Amazon Pinpoint mediante programación.
- Si el nuevo tipo de canal para el punto de conexión es ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX o BAIDU:
  - Compruebe que al menos uno de los terminales actualmente asociados al miembro de la audiencia sea ADM, GCM, APNS, ChannelTypeAPNS_VOICE, APNS_VOIP_SANDBOX o BAIDU. Si no lo hay, se devuelve y es necesario eliminar un punto final antes de volver a BadRequestExceptionintentarlo, consulte. Eliminar puntos de conexión de Amazon Pinpoint mediante programación
  - De lo contrario, el punto final más antiguo EffectiveDatese establece en ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX o BAIDU. `INACTIVE` ChannelType
    - Se elimina el del punto final anterior. UserId

- El nuevo punto de conexión está asociado al miembro de la audiencia y este sigue teniendo el número máximo de puntos de conexión.

El punto final se puede volver a habilitar configurando el estado en el punto final `ACTIVE` y UserIdvolviendo a añadirlo al mismo.

Eliminar puntos de conexión de Amazon Pinpoint mediante programación

Un punto de enlace representa un único método de ponerse en contacto con uno de sus clientes. Cada punto de enlace puede hacer referencia a la dirección de correo electrónico de un cliente, al identificador del dispositivo móvil, al número de teléfono o a otro tipo de destino al que puede enviar mensajes. En muchas jurisdicciones, este tipo de información puede considerarse personal. Puede eliminar puntos de conexión cuando ya no desee enviar mensajes a un destino determinado, por ejemplo, si el destino deja de estar disponible o cuando un cliente cierra una cuenta.

Ejemplos

En los siguientes ejemplos puede ver cómo eliminar un punto de enlace.

AWS CLI

Puede utilizar Amazon Pinpoint ejecutando comandos con la AWS CLI.

Example Comando Delete Endpoint

Para eliminar un punto de enlace, utilice el comando `delete-endpoint`:

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

Donde:

- application-id es el ID del proyecto de Amazon Pinpoint que contiene el punto de conexión.
- endpoint-id es el ID del punto de enlace que está eliminando.

La respuesta a este comando es la definición JSON del punto de enlace que eliminó.

AWS SDK para Java

Puede usar la API de Amazon Pinpoint en las aplicaciones Java mediante el cliente que proporciona AWS SDK para Java.

Example Código

Para eliminar un punto de conexión, utilice el método `deleteEndpoint` del cliente de `AmazonPinpoint`. Proporcione un objeto `DeleteEndpointRequest` como argumento de método:

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

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

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

                Usage:   <appName> <endpointId >

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

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

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

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

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

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

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

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

Para ver el ejemplo completo del SDK, consulte DeleteEndpoint.java en GitHub.

HTTP

Puede utilizar Amazon Pinpoint realizando solicitudes HTTP directamente a la API de REST.

Example Solicitud DELETE al punto de enlace

Para eliminar un punto de enlace, emita una solicitud `DELETE` al recurso Punto de enlace:

```
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
```

Donde:

- application-id es el ID del proyecto de Amazon Pinpoint que contiene el punto de conexión.
- endpoint-id es el ID del punto de enlace que está eliminando.

La respuesta a esta solicitud es la definición JSON del punto de enlace que eliminó.

Crear o importar segmentos en Amazon Pinpoint.

Un segmento de usuarios representa un subconjunto de usuarios basado en ciertas características compartidas, como, por ejemplo, la forma en la que han usado recientemente la aplicación o qué plataforma de dispositivos utilizan. Un segmento designa qué usuarios reciben los mensajes enviados por una campaña. Defina segmentos para llegar al público correcto cuando desee invitar a usuarios a volver a la aplicación, hacer ofertas especiales o aumentar de otro modo las compras de los usuarios y su implicación.

Después de crear un segmento, puede utilizarlo en una o varias campañas. Una campaña envía mensajes personalizados a los usuarios del segmento.

Para obtener más información, consulte Segmentos.

Temas

- Generar segmentos en Amazon Pinpoint
- Importar segmentos en Amazon Pinpoint.
- Personalice los segmentos de Amazon Pinpoint mediante una función AWS Lambda

Generar segmentos en Amazon Pinpoint

Para llegar al público previsto de una campaña, cree un segmento basado en los datos notificados por la aplicación. Por ejemplo, para llegar a los usuarios que no hayan usado la aplicación recientemente, puede definir un segmento para los usuarios que no la hayan utilizado en los últimos 30 días.

Para ver más ejemplos de código, consulte Ejemplos de código.

Cree segmentos con AWS SDK para Java

El siguiente ejemplo muestra cómo crear un segmento con AWS SDK para Java. El ejemplo crea un segmento de usuarios cuyo equipo es los `Lakers` y han estado activos durante los últimos 30 días. Una vez creado el segmento, puede usarlo como parte de una campaña o un recorrido. Para ver un ejemplo del uso de un segmento con una campaña, consulte Crear campañas de Amazon Pinpoint mediante programación.

```
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;
        }
}
```

Cuando se ejecuta este ejemplo, se muestra lo siguiente en la ventana de la consola de su IDE:

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

Para ver el ejemplo completo del SDK, consulta CreateSegment.java en GitHub.

Importar segmentos en Amazon Pinpoint.

Con Amazon Pinpoint, puede definir un segmento de usuarios mediante la importación de información acerca de los puntos de conexión que pertenecen al segmento. Un punto de enlace es un único destino de mensajería, como un token de inserción en dispositivos móviles, un número de teléfono móvil o una dirección de correo electrónico.

La importación de segmentos resulta útil si ya se han creado segmentos de los usuarios fuera de Amazon Pinpoint, pero se desea conectar con los usuarios en campañas de Amazon Pinpoint.

Al importar un segmento, Amazon Pinpoint obtiene los puntos de conexión del segmento desde Amazon Simple Storage Service (Amazon S3). Antes de importar, se agregan los puntos de conexión a Amazon S3 y se crea un rol de IAM que concede a Amazon Pinpoint acceso a Amazon S3. A continuación, se proporciona a Amazon Pinpoint la ubicación de Amazon S3 en la que se almacenan los puntos de conexión y Amazon Pinpoint agrega cada punto de conexión al segmento.

Para crear el rol de IAM, consulte Rol de IAM para importar puntos de conexión o segmentos. Para obtener información sobre la importación de un segmento mediante la consola de Amazon Pinpoint, consulte Importación de segmentos en la Guía del usuario de Amazon Pinpoint.

Para ver más ejemplos de código, consulte Ejemplos de código.

Importar un segmento con AWS SDK para Java

En el siguiente ejemplo se muestra cómo importar un segmento mediante AWS SDK para 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;
```

```
    }
}
```

Para ver el ejemplo completo del SDK, consulte ImportingSegments.java en GitHub.

Personalice los segmentos de Amazon Pinpoint mediante una función AWS Lambda

> Esta es una documentación preliminar para una característica en versión beta pública. Está sujeta a cambios.

Puede utilizarla AWS Lambda para personalizar la forma en que una campaña de Amazon Pinpoint atrae a su público objetivo. Con AWS Lambda, puede modificar el segmento de la campaña en el momento en que Amazon Pinpoint envía el mensaje de la campaña.

AWS Lambda es un servicio informático que puede utilizar para ejecutar código sin aprovisionar ni administrar servidores. Puede empaquetar el código y cargarlo en Lambda como funciones de Lambda. Lambda ejecuta una función cuando se invoca la función, lo que se podría hacer de forma manual o automática en respuesta a eventos. Para obtener más información, consulte la Guía para desarrolladores de AWS Lambda.

Para asignar una función de Lambda a una campaña, defina la configuración `CampaignHook` de la campaña mediante el recurso de campaña en la API de Amazon Pinpoint. Entre estos ajustes se incluye el nombre de la función de Lambda. También se incluye el modo `CampaignHook`, que establece si Amazon Pinpoint recibe un valor de retorno de la función.

Una función de Lambda que asigne a una campaña se conoce como una extensión de Amazon Pinpoint.

Con la configuración de `CampaignHook` definida, Amazon Pinpoint invoca automáticamente la función de Lambda cuando ejecuta la campaña, antes de enviar el mensaje de la campaña. Cuando Amazon Pinpoint invoca la función, proporciona datos de eventos sobre el envío de mensajes. Estos datos incluyen el segmento de la campaña, que es la lista de puntos de conexión a los que Amazon Pinpoint envía el mensaje.

Si el modo `CampaignHook` está establecido en `FILTER`, Amazon Pinpoint permite que la función modifique y devuelva el segmento antes de enviar el mensaje. Por ejemplo, la función podría

actualizar las definiciones del punto de conexión con atributos que contienen datos de un origen externo a Amazon Pinpoint. O bien, la función podría filtrar el segmento quitando ciertos puntos de enlace dependiendo de las condiciones del código de la función. Una vez Amazon Pinpoint reciba el segmento modificado de la función, envía el mensaje a cada uno de los puntos de conexión del segmento utilizando el canal de entrega de la campaña.

Al procesar tus segmentos con ellos AWS Lambda, tienes más control sobre a quién envías los mensajes y qué contienen esos mensajes. Puede adaptar sus campañas en tiempo real, en el momento en que los mensajes de la campaña se envíen. Filtrar segmentos le permite atraer a subconjuntos de los segmentos definidos más concretamente. Agregar o actualizar los atributos de los puntos de enlace también permite que haya nuevos datos disponibles para las variables de mensajes.

> **Note**
>
> También puede utilizar la configuración de `CampaignHook` para asignar una función de Lambda que administre la entrega de mensajes. Este tipo de función resulta útil para la entrega de mensajes a través de canales personalizadas que Amazon Pinpoint no admite, como, por ejemplo, las plataformas de redes sociales. Para obtener más información, consulte Crear un canal personalizado en Amazon Pinpoint mediante un webhook o una función de Lambda.
>
> Al invocar un enlace de Lambda mediante Amazon Pinpoint, la función de Lambda también debe estar en la misma región que el proyecto de Amazon Pinpoint.

Para modificar los segmentos de la campaña AWS Lambda, cree primero una función que procese los datos de eventos enviados por Amazon Pinpoint y devuelva un segmento modificado. A continuación, autorice a Amazon Pinpoint a invocar la función asignando una política de función de Lambda. Por último, asigne la función a una o varias campañas definiendo configuraciones de `CampaignHook`.

Para ver más ejemplos de código, consulte Ejemplos de código.

Datos de evento

Cuando Amazon Pinpoint invoca la función de Lambda, proporciona la siguiente carga como los datos de eventos:

```
{
```

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

AWS Lambda pasa los datos del evento a su código de función. Los datos de eventos proporciona los siguientes atributos:

- `MessageConfiguration`: tiene la misma estructura que el objeto `DirectMessageConfiguration` del recurso de mensajes en la API de Amazon Pinpoint.
- `ApplicationId`: el ID del proyecto de Amazon Pinpoint al que pertenece la campaña.
- `CampaignId`: el ID de la campaña de Amazon Pinpoint para la que se invocó la función.
- `TreatmentId`— El ID de una variante de la campaña que se utiliza para A/B realizar pruebas.
- `ActivityId`: el ID de la actividad que está realizando la campaña.
- `ScheduledTime`: la fecha y hora, en formato ISO 8601, en que se enviarán los mensajes de la campaña.
- `Endpoints`— Un mapa que asocia el punto final IDs con las definiciones del punto final. Cada carga de datos de eventos contiene hasta 50 puntos de enlace. Si el segmento de la campaña contiene más de 50 puntos de conexión, Amazon Pinpoint invoca la función reiteradamente, de 50 en 50 puntos de conexión, hasta que se hayan procesado todos ellos.

Crear una función de Lambda

Para obtener información sobre cómo crear una función de Lambda, consulte Introducción en la Guía para desarrolladores de AWS Lambda . Al crear la función, recuerde que la entrega de mensajes produce un error en las siguientes condiciones:

- La función de Lambda tarda más de 15 segundos en devolver el segmento modificado.
- Amazon Pinpoint no puede decodificar el valor de retorno de la función.
- La función requiere más de 3 intentos de Amazon Pinpoint para invocarla correctamente.

Amazon Pinpoint solo acepta las definiciones de punto de conexión del valor de retorno de la función. La función no puede modificar otros elementos de los datos de eventos.

Ejemplo de función de Lambda

La función de Lambda procesa los datos de eventos enviados por Amazon Pinpoint y devuelve los puntos de conexión modificados, tal y como se muestra en el siguiente controlador de ejemplo, escrito 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 transfiere los datos de eventos al controlador según el parámetro `event`.

En este ejemplo, el controlador itera mediante cada punto de enlace en el objeto `event.Endpoints` y añade un nuevo atributo, `CreditScore`, en el punto de enlace. El valor del atributo `CreditScore` es simplemente un número aleatorio.

La `console.log()` declaración registra el evento en CloudWatch los registros.

La instrucción `callback()` devuelve los puntos de conexión modificados a Amazon Pinpoint. Normalmente, el parámetro `callback` es opcional en las funciones de Lambda de Node.js, pero es obligatorio en este contexto porque la función debe devolver los puntos de conexión actualizados a Amazon Pinpoint.

La función debe devolver los puntos finales en el mismo formato que proporcionan los datos del evento, es decir, un mapa que asocia los puntos finales IDs con las definiciones de los puntos finales, como en el siguiente ejemplo:

```
{
    "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 función de ejemplo modifica y devuelve el objeto `event.Endpoints` que recibió en los datos de evento.

Si lo prefiere, puede incluir los atributos `BodyOverride` y `TitleOverride` en las definiciones de los puntos de enlace que se devuelven.

> **Note**
>
> Cuando se utiliza esta solución para enviar mensajes, Amazon Pinpoint solo respeta los atributos `TitleOverride` y `BodyOverride` de los puntos de conexión cuando el valor del atributo `ChannelType` es uno de los siguientes: `ADM`, `APNS`, `APNS_SANDBOX`, `APNS_VOIP`, `APNS_VOIP_SANDBOX`, `BAIDU`, `GCM` o `SMS`.

Amazon Pinpoint no respeta estos atributos de los puntos de conexión cuando el valor del atributo `ChannelType` es `EMAIL`.

Asignar una política de función de Lambda

Antes de poder utilizar la función de Lambda para procesar los puntos de conexión, debe autorizar a Amazon Pinpoint a invocar la función de Lambda. Para conceder permiso de invocación, asigne una política de función de Lambda a la función. Una política de función de Lambda consiste en una política de permisos basada en el recurso que designa qué entidades pueden usar la función y qué acciones pueden realizar dichas entidades.

Para obtener más información, consulte Uso de políticas basadas en recursos para AWS Lambda en la Guía del desarrollador de AWS Lambda .

Política de función de ejemplo

La siguiente política otorga permiso al director del servicio Amazon Pinpoint para usar la `lambda:InvokeFunction` acción en una campaña específica ()*campaign-id*:

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

La política de función requiere un bloque `Condition` que incluye una clave `AWS:SourceArn`. Este código indica qué campaña de Amazon Pinpoint tiene permitido invocar la función. En este ejemplo,

la política concede permiso solo a una única campaña. El `Condition` bloqueo también debe incluir una `AWS:SourceAccount` clave que controle qué AWS cuenta puede invocar la acción.

Para escribir una política más genérica, use caracteres comodín (*) que sustituyan a varios caracteres. Por ejemplo, puede usar el siguiente `Condition` bloque para permitir que cualquier campaña de un proyecto específico de Amazon Pinpoint (*`application-id`*) invoque la función:

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

Si desea que la función de Lambda sea la función predeterminada que utilizan todas las campañas de un proyecto, le recomendamos que configure el bloque `Condition` para la política de la forma anterior. Para obtener información sobre la configuración de una función de Lambda como predeterminada para todas las campañas de un proyecto, consulte Asignar una función de Lambda a una campaña.

Conceder el permiso de invocación de Amazon Pinpoint

Puede usar AWS Command Line Interface (AWS CLI) para añadir permisos a la política de funciones de Lambda asignada a su función de Lambda. Para permitir que Amazon Pinpoint invoque una función para una campaña específica, use el comando `add-permission` de Lambda, como se muestra en el siguiente ejemplo:

```
$ 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
```

Puedes buscar tu campaña IDs mediante el comando get-campaigns de. AWS CLI También puede buscar el ID de la aplicación mediante el comando get-apps.

Cuando ejecuta el comando `add-permission` de Lambda, Lambda devuelve la siguiente salida:

```
{
  "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\"}}}
}
```

El valor de `Statement` es una versión de cadena JSON de la instrucción que se agregó a la política de la función de Lambda.

Asignar una función de Lambda a una campaña

Puede asignar una función de Lambda a una campaña de Amazon Pinpoint individual. O bien, puede configurar la función de Lambda como la predeterminada que utilizan todas las campañas de un proyecto, excepto en el caso de las campañas a las que se le asigne una función de manera individual.

Para asignar una función de Lambda a una campaña individual, use la API de Amazon Pinpoint a fin de crear o actualizar un objeto `Campaign` y defina su atributo `CampaignHook`. Para establecer una función de Lambda como la predeterminada de todas las campañas de un proyecto, cree o actualice el recurso `Settings` de ese proyecto y defina su objeto `CampaignHook`.

En ambos casos, establezca los siguientes atributos `CampaignHook`:

- `LambdaFunctionName`: el nombre o ARN de la función de Lambda que Amazon Pinpoint invoca antes de enviar mensajes para la campaña.

- `Mode`: se establece en `FILTER`. Con este modo, Amazon Pinpoint invoca la función y espera a que devuelva los puntos de conexión modificados. Tras recibirlos, Amazon Pinpoint envía el mensaje. Amazon Pinpoint espera hasta 15 segundos antes de que se produzca un error en el envío del mensaje.

Con la configuración de `CampaignHook` definida para una campaña, Amazon Pinpoint invoca la función de Lambda especificada antes de enviar los mensajes de la campaña. Amazon Pinpoint espera a recibir los puntos de conexión modificados de la función. Si Amazon Pinpoint recibe los puntos de conexión actualizados, continúa con el envío de mensajes, mediante los datos de punto de conexión actualizados.

Crear campañas de Amazon Pinpoint mediante programación

Para ayudar a aumentar la conexión entre la aplicación y los usuarios, utilice Amazon Pinpoint para crear y administrar campañas de notificaciones de inserción que se dirijan a segmentos de usuarios concretos.

Por ejemplo, la campaña podría invitar a los usuarios que no han ejecutado la aplicación recientemente a que regresen a ella u ofrecer promociones especiales a los usuarios que no han comprado últimamente.

Una campaña envía un mensaje personalizado a un segmento de usuario que se especifique. La campaña puede enviar el mensaje a todos los usuarios del segmento, o bien puede asignar exclusiones, que es un porcentaje de los usuarios que no recibirán ningún mensaje.

Puede establecer la programación de la campaña para enviar el mensaje una vez o con una frecuencia recurrente, como, por ejemplo, una vez a la semana. Para evitar que los usuarios reciban el mensaje a horas poco apropiadas, la programación puede incluir una horas de silencio durante las que no se enviará ningún mensaje.

Para experimentar con estrategias de campaña alternativas, configura tu campaña de forma que una A/B test. An A/B prueba incluya dos o más versiones del mensaje o la programación. Los tratamientos son variaciones del mensaje o la programación. A medida que los usuarios respondan a la campaña, podrá ver análisis de dicha campaña para comparar la eficacia de cada tratamiento.

Para obtener más información, consulte Campaigns en la Guía de API de REST de Amazon Pinpoint o Campaigns en la Guía del usuario de Amazon Pinpoint.

Crear una campaña estándar de Amazon Pinpoint

Una campaña estándar envía una notificación de inserción personalizada a un segmento especificado de acuerdo con una programación que se define. El siguiente ejemplo muestra cómo crear una campaña con AWS SDK para Java. Para ver un ejemplo de cómo crear un segmento para transferirlo, consulte Generar segmentos en Amazon Pinpoint.

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

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

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

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

        final String usage = """

                Usage:   <appId> <segmentId>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

        return null;
    }
}
```

Cuando se ejecuta este ejemplo, se muestra lo siguiente en la ventana de la consola de su IDE:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Para ver el ejemplo completo del SDK, consulta CreateCampaign.java en GitHub.

Cree una campaña A/B de prueba de Amazon Pinpoint con el AWS SDK para Java

Una campaña de A/B prueba se comporta como una campaña estándar, pero te permite definir distintos tratamientos para el mensaje o el calendario de la campaña. Una prueba A/B incluye dos o más tratamientos del mensaje o la programación. Los tratamientos son variaciones del mensaje o la programación. A medida que los usuarios respondan a la campaña, podrá ver análisis de dicha campaña para comparar la eficacia de cada tratamiento.

En el siguiente ejemplo se muestra cómo crear una campaña A/B de prueba con. AWS SDK para 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();
    }

}
```

Cuando se ejecuta este ejemplo, se muestra lo siguiente en la ventana de la consola de su IDE:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Administración de etiquetas de recurso de Amazon Pinpoint

Una etiqueta es una etiqueta que puede definir y asociar de forma opcional a AWS los recursos, incluidos determinados tipos de recursos de Amazon Pinpoint. Las etiquetas pueden ayudarle a clasificar y administrar recursos de distintas formas, como, por finalidad, propietario, entorno u otros criterios. Por ejemplo, puede usar etiquetas para aplicar políticas o automatización, o bien para identificar recursos que están sujetos a determinados requisitos de conformidad. Puede agregar etiquetas a los siguientes tipos de recursos de Amazon Pinpoint:

- Campaigns (Campañas)
- Plantillas de mensaje
- Proyectos (aplicaciones)
- Segmentos

Un recurso puede tener hasta 50 etiquetas. Cada etiqueta está formada por una clave de etiqueta y un valor de etiqueta opcional, ambos definidos por el usuario. Un clave de etiqueta es una etiqueta general que actúa como una categoría para valores de etiqueta más específicos. Un valor de etiqueta actúa como descriptor de una clave de etiqueta.

Una clave de etiqueta puede incluir hasta 128 caracteres. Un valor de etiqueta puede incluir hasta 256 caracteres. Los caracteres pueden ser letras Unicode, números, espacios en blanco o uno de los siguientes símbolos: _ . : / = + -. Las siguientes restricciones adicionales se aplican a las etiquetas:

- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- Para cada recurso asociado, cada clave de etiqueta debe ser única y solo puede tener un valor.
- El `aws:` prefijo está reservado para que lo usen AWS; no puede usarlo en ninguna clave de etiqueta o valor que defina. Además, las claves o valores de etiqueta que utilizan este prefijo no se pueden editar ni quitar. Las etiquetas que usan este prefijo no cuentan para la cuota de 50 etiquetas por recurso.
- No puede actualizar ni eliminar un recurso en función únicamente de sus etiquetas. También debe especificar el ID de recurso o nombre de recurso de Amazon (ARN), dependiendo de la operación que use.
- Puede asociar etiquetas a recursos públicos o compartidos. Sin embargo, las etiquetas solo están disponibles para tu AWS cuenta, no para ninguna otra cuenta que comparta el recurso.

Además, las etiquetas solo están disponibles para los recursos que se encuentran en la AWS región especificada para su AWS cuenta.

Para añadir, mostrar, actualizar y eliminar claves y valores de etiquetas de los recursos de Amazon Pinpoint, puede utilizar AWS Command Line Interface (AWS CLI), la API de Amazon Pinpoint, Grupos de recursos de AWS la API de etiquetado o un SDK. AWS Para administrar las claves y los valores de las etiquetas en todos los AWS recursos que se encuentran en una AWS región específica de su AWS cuenta (incluidos los recursos de Amazon Pinpoint), utilice la API de Grupos de recursos de AWS etiquetado.

Para obtener más información acerca de los comandos de la CLI que puede usar para administrar recursos de Amazon Pinpoint, consulte la sección de Amazon Pinpoint de la Referencia de comandos de la AWS CLI.

Para obtener más información acerca de los recursos de la API de Amazon Pinpoint, incluidos los esquemas, parámetros y métodos HTTP(S) admitidos, consulte la Referencia de la API de Amazon Pinpoint.

Utilizar etiquetas de Amazon Pinpoint en las políticas de IAM y las operaciones de API

Una vez que comience a implementar etiquetas, podrá aplicar permisos basados en etiquetas a nivel de recursos a las políticas AWS Identity and Access Management (IAM) y las operaciones de la API. Esto incluye operaciones que admiten agregar etiquetas a los recursos cuando se crean recursos. Al usar las etiquetas de esta manera, puede implementar un control pormenorizado sobre qué grupos y usuarios de su AWS cuenta tienen permiso para crear y etiquetar recursos, y qué grupos y usuarios tienen permiso para crear, actualizar y eliminar etiquetas de manera más general.

Por ejemplo, puede crear una política que permita a un usuario tener acceso completo a todos los recursos de Amazon Pinpoint en los que su nombre sea un valor en la etiqueta `Owner` del recurso:

JSON

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

```
            "Effect": "Allow",
            "Action": "mobiletargeting:*",
            "Resource": "*",
            "Condition": {
               "StringEqualsIgnoreCase": {
                  "aws:ResourceTag/Owner": "${aws:username}"
               }
            }
         }
      ]
}
```

Si define los permisos de nivel de recurso basados en etiquetas, estos entrarán en vigor inmediatamente. Esto significa que sus recursos están más seguros en cuanto se crean y que puede empezar a aplicar el uso de etiquetas de nuevos recursos rápidamente. También puede usar permisos de nivel de recurso para controlar las claves y valores de etiqueta que se pueden asociar a recursos nuevos y existentes. Para obtener más información, consulte Control del acceso mediante etiquetas en la AWS Guía del usuario de IAM de .

Agregar etiquetas a los recursos de Amazon Pinpoint mediante programación

En los ejemplos siguientes se muestra cómo agregar una etiqueta a un recurso de Amazon Pinpoint mediante la AWS CLI y la API de REST de Amazon Pinpoint. También puedes usar cualquier AWS SDK compatible para añadir una etiqueta a un recurso.

Para añadir una etiqueta a varios recursos de Amazon Pinpoint en una sola operación, utilice las operaciones de etiquetado de grupos de recursos de la API de etiquetado AWS CLI o la Grupos de recursos de AWS API de etiquetado.

Agregar etiquetas mediante la API

Para crear un recurso nuevo y agregarle una etiqueta mediante la API de REST de Amazon Pinpoint, envíe una solicitud POST al URI de recurso apropiado. En el cuerpo de la solicitud, incluya el parámetro `tags` y los valores. En el siguiente ejemplo se muestra cómo especificar una etiqueta al crear un nuevo proyecto.

```
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"
   }
}
```

Para agregar una etiqueta a un recurso existente, envíe una solicitud POST al URI de Etiquetas. Incluya el nombre de recurso de Amazon (ARN) del recurso en el URI. El ARN debe estar codificado en la URL. En el cuerpo de la solicitud, incluya el parámetro `tags` y los valores, tal y como se muestra en el siguiente ejemplo.

```
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"
   }
}
```

Añada etiquetas mediante la AWS CLI

Para crear un nuevo recurso y añadirle una etiqueta mediante el AWS CLI, utilice el `create` comando correspondiente al recurso. Incluya el parámetro `tags` y los valores. En el siguiente ejemplo se muestra cómo especificar etiquetas al crear un nuevo proyecto.

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}
```

En el ejemplo anterior, realice el siguiente procedimiento:

- *MyProject*Sustitúyalo por el nombre que desee asignar al proyecto.
- Sustituya *key1* y *key2* por las claves de las etiquetas que desee añadir al recurso.
- Sustituya *value1* y *value2* por los valores de las etiquetas que desee añadir para las claves respectivas.

Para obtener información acerca de los comandos que puede usar para crear un recurso de Amazon Pinpoint, consulte la Referencia de comandos de la AWS CLI.

Para agregar una etiqueta a un recurso existente, utilice el comando `tag-resource` y especifique los valores adecuados para los parámetros requeridos:

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}
```

En el ejemplo anterior, realice el siguiente procedimiento:

- *resource-arn*Sustitúyalo por el nombre de recurso de Amazon (ARN) del recurso al que quieres añadir una etiqueta.
- Sustituya *key1* y *key2* por las claves de las etiquetas que desee añadir al recurso.
- Sustituya *value1* y *value2* por los valores de las etiquetas que desee añadir para las claves respectivas.

Mostrar etiquetas de los recursos de Amazon Pinpoint mediante programación

Los ejemplos siguientes muestran cómo utilizar la AWS CLI y la API de REST de Amazon Pinpoint para mostrar una lista de todas las etiquetas (claves y valores) asociadas a un recurso de Amazon Pinpoint. También puedes usar cualquier AWS SDK compatible para mostrar las etiquetas asociadas a un recurso.

Mostrar etiquetas mediante la API

Para utilizar la API de REST de Amazon Pinpoint para mostrar todas las etiquetas que están asociadas a un recurso específico, envíe una solicitud GET al URI Etiquetas e incluya el nombre de recurso de Amazon (ARN) del recurso en el URI. El ARN debe estar codificado en la URL. Por ejemplo, la siguiente solicitud recupera todas las etiquetas asociadas a una campaña específica (*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 respuesta JSON a la solicitud incluye un objeto `tags`. El objeto `tags` enumera todas las claves de etiqueta y los valores asociados a la campaña.

Para mostrar todas las etiquetas asociadas a más de un recurso del mismo tipo, envíe una solicitud GET al URI adecuado para dicho tipo de recurso. Por ejemplo, la siguiente solicitud recupera información sobre todas las campañas del proyecto especificado ()*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 respuesta JSON a la solicitud muestra todas las campañas del proyecto. El objeto `tags` de cada campaña muestra todas las claves y valores de etiqueta asociados a la campaña.

Muestre las etiquetas mediante el AWS CLI

Para usar el AWS CLI para mostrar una lista de las etiquetas asociadas a un recurso específico, ejecute el `list-tags-for-resource` comando y especifique el nombre del recurso de Amazon (ARN) del recurso para el `resource-arn` parámetro, como se muestra en el siguiente ejemplo.

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
```

Para mostrar una lista de todos los recursos de Amazon Pinpoint que tienen etiquetas y todas las etiquetas asociadas a cada uno de esos recursos, utilice el comando get-resources de la API de etiquetado. Grupos de recursos de AWS Establezca el parámetro `resource-type-filters` en `mobiletargeting`, como se muestra en el siguiente ejemplo.

Linux, macOS, or Unix

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

Windows Command Prompt

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

El resultado del comando es una lista de todos ARNs los recursos de Amazon Pinpoint que tienen etiquetas. En la lista se incluyen todas las claves y valores de etiqueta asociados a cada recurso.

Actualizar o sobrescribir etiquetas de recursos de Amazon Pinpoint mediante programación

Existen varias formas de actualizar (sobrescribir) una etiqueta para un recurso de Amazon Pinpoint. La mejor manera de actualizar una etiqueta depende de:

- El tipo de recurso para el que desea actualizar etiquetas.
- Si desea actualizar una etiqueta para uno o varios recursos al mismo tiempo.
- Si desea actualizar una clave de etiqueta, un valor de etiqueta, o ambos.

Para actualizar una etiqueta para un proyecto de Amazon Pinpoint o para varios recursos al mismo tiempo, utilice las operaciones de etiquetado de grupos de recursos de la API de etiquetado AWS CLI o de la API de etiquetado.Grupos de recursos de AWS En la actualidad, la API de Amazon Pinpoint no admite directamente ninguna de esas tareas.

Para actualizar una etiqueta para un recurso, puede eliminar la etiqueta actual y agregar una nueva etiqueta mediante la API de Amazon Pinpoint.

Quitar etiquetas de los recursos de Amazon Pinpoint mediante programación

Los siguientes ejemplos muestran cómo eliminar una etiqueta (tanto la clave como el valor) de un recurso de Amazon Pinpoint mediante la AWS CLI y la API de REST de Amazon Pinpoint. También puede usar cualquier AWS SDK compatible para eliminar una etiqueta de un recurso.

Para eliminar una etiqueta de varios recursos de Amazon Pinpoint en una sola operación, utilice las operaciones de etiquetado de grupos de recursos de la API de etiquetado AWS CLI o la Grupos de recursos de AWS API de etiquetado. Para quitar solo un valor de etiqueta específico (no una clave de etiqueta) de un recurso, puede actualizar la etiqueta para el recurso.

Quitar etiquetas mediante la API

Para eliminar una etiqueta de un recurso mediante la API de REST de Amazon Pinpoint, envíe una solicitud DELETE al URI de las etiquetas. En el URI, incluya el nombre de recurso de Amazon (ARN)

del recurso del que desea quitar una etiqueta, seguida del parámetro `tagKeys` y la etiqueta que se va a quitar. Por ejemplo:

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

Donde:

- *endpoint*es el punto de enlace de Amazon Pinpoint para la AWS región que aloja el recurso.
- *resource-arn*es el ARN del recurso del que desea eliminar una etiqueta.
- *key*es la etiqueta que desea eliminar del recurso.

Todos los parámetros deben tener codificación URL.

Para quitar varias claves de etiquetas y sus valores asociados de un recurso, añada el parámetro `tagKeys` y el argumento para cada una de las etiquetas adicionales que se van a quitar, separadas por un signo ampersand (&). Por ejemplo:

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

Todos los parámetros deben tener codificación URL.

Elimine las etiquetas mediante el AWS CLI

Para eliminar una etiqueta de un recurso mediante el AWS CLI, ejecute el `untag-resource` comando. Incluya el argumento y el parámetro `tag-keys`, como se muestra en el siguiente ejemplo.

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
```

En el ejemplo anterior, realice los siguientes cambios:

- *resource-arn*Sustitúyalo por el ARN del recurso del que desea eliminar las etiquetas.
- Sustituya *key1* y *key2* por las claves de las etiquetas que desee eliminar del recurso.

Integrar Amazon Pinpoint con la aplicación

Integre Amazon Pinpoint con el código del cliente para comprender a los usuarios e interactuar con ellos.

Una vez realizada la integración, a medida que los usuarios ejecuten la aplicación, esta se conectará al servicio de Amazon Pinpoint para agregar o actualizar los puntos de conexión. Los puntos de conexión representan los destinos a los que se pueden enviar mensajes, como dispositivos de usuario, direcciones de correo electrónico o números de teléfono.

La aplicación puede después proporcionar datos de uso o eventos. Consulte los datos de eventos en la consola de Amazon Pinpoint para saber cuántos usuarios tiene, la frecuencia con la que utilizan la aplicación, cuándo lo hacen, etc.

Una vez que la aplicación haya proporcionado puntos de enlace y eventos, puede utilizar esta información para personalizar campañas de mensajería para públicos específicos, o segmentos. (También puede enviar mensajes directamente a listas sencillas de destinatarios sin crear campañas.)

Utilice los temas de esta sección para integrar Amazon Pinpoint con una aplicación móvil o web. Estos temas incluyen ejemplos de código y procedimientos para realizar la integración con una aplicación de JavaScript, Android, Swift o Flutter. Para iniciar la integración de sus aplicaciones, consulte the section called “Conectar la aplicación frontend mediante Amplify”.

Aparte del cliente, puede utilizar los AWS SDK compatibles o la API de Amazon Pinpoint para importar puntos de conexión, exportar datos de eventos, definir segmentos de clientes, crear y ejecutar campañas, etc.

Temas

- Uso de Amazon Pinpoint con un AWS SDK
- Conecte su aplicación de interfaz a Amazon AWS Pinpoint mediante Amplify
- Registrar puntos de conexión de Amazon Pinpoint en su aplicación
- Notificar eventos de Amazon Pinpoint en la aplicación

Uso de Amazon Pinpoint con un AWS SDK

Los kits de desarrollo de software (SDK) de AWS se encuentran disponibles en muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK para C++	AWS SDK para C++ Ejemplos de código de la
AWS CLI	AWS CLI Ejemplos de código de la
AWS SDK para Go	AWS SDK para Go Ejemplos de código de la
AWS SDK para Java	AWS SDK para Java Ejemplos de código de la
AWS SDK para JavaScript	AWS SDK para JavaScript Ejemplos de código de la
AWS SDK para Kotlin	AWS SDK para Kotlin Ejemplos de código de la
AWS SDK para .NET	AWS SDK para .NET Ejemplos de código de la
AWS SDK para PHP	AWS SDK para PHP Ejemplos de código de la
Herramientas de AWS para PowerShell	Herramientas de AWS para PowerShell Ejemplos de código de la
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) Ejemplos de código de la
AWS SDK para Ruby	AWS SDK para Ruby Ejemplos de código de la
AWS SDK para Rust	AWS SDK para Rust Ejemplos de código de la
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP Ejemplos de código de la
AWS SDK para Swift	AWS SDK para Swift Ejemplos de código de la

Para obtener ejemplos específicos de Amazon Pinpoint, consulte Ejemplos de código para Amazon Pinpoint con AWS SDK.

> **Ejemplo de disponibilidad**
>
> ¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Conecte su aplicación de interfaz a Amazon AWS Pinpoint mediante Amplify

Usa AWS Amplify para integrar tu aplicación con. AWS Para las aplicaciones de Swift, consulte Introducción en la documentación de Amplify para Swift. Para las aplicaciones de Android, consulte Introducción en la documentación de Amplify para Android SDK. Para la aplicación React Native, consulta Cómo empezar en la documentación de Amplify JavaScript . Para las aplicaciones de Flutter, consulte Introducción en la documentación de Flutter SDK. Estos temas le ayudan a:

- Configure los recursos de backend.
- Conecte la aplicación a los recursos de backend mediante las bibliotecas de Amplify.

Para obtener más información sobre cómo conectar la aplicación frontend a Amazon Pinpoint para análisis, mensajería integrada en la aplicación y notificaciones push, consulte AWS Amplify.

Siguiente paso

Tras integrar AWS Amplify con la aplicación, actualiza el código para registrar los dispositivos de los usuarios como puntos finales. Para obtener más información, consulte Registrar puntos de conexión de Amazon Pinpoint en su aplicación.

Registrar puntos de conexión de Amazon Pinpoint en su aplicación

Cuando un usuario inicia una sesión (por ejemplo, lanzando la aplicación móvil), la aplicación web o móvil puede registrar (o actualizar) automáticamente un punto de conexión con Amazon Pinpoint. El punto de enlace representa el dispositivo con el que se inicia la sesión. Incluye atributos que describen el dispositivo, y también puede incluir los atributos personalizados que se definan. Los

puntos de enlace también pueden representar otros métodos de comunicarse con los clientes, como direcciones de correo electrónico o números de teléfono móvil.

Una vez que la aplicación haya registrado los puntos de enlace, puede segmentar el público en función de los atributos de los puntos de enlace. A continuación, puede atraer a estos segmentos con campañas de mensajería personalizadas. También puede utilizar la página Análisis de la consola de Amazon Pinpoint para ver gráficos sobre el registro y la actividad de los puntos de conexión, por ejemplo, Puntos de conexión nuevos y Puntos de conexión activos diarios.

Puede asignar un único ID de usuario a varios puntos de enlace. Un ID de usuario representa a un único usuario, mientras que cada uno de los puntos de enlace a los que se asocia el ID de usuario representa uno de los dispositivos del usuario. Después de asignar un usuario IDs a sus puntos de conexión, puede ver gráficos sobre la actividad de los usuarios en la consola, como los usuarios activos diarios y los usuarios activos mensuales.

Antes de empezar

Si aún no lo has hecho, integra el AWS Mobile SDK for Android o iOS o integra la JavaScript biblioteca AWS Amplify con tu aplicación. Para obtener más información, consulte Conecte su aplicación de interfaz a Amazon AWS Pinpoint mediante Amplify.

Registre los puntos finales con el AWS móvil SDKs para Android o iOS

Puede usar AWS Mobile SDKs para Android o iOS para registrar y personalizar los puntos finales. Para obtener más información y ver ejemplos de código, consulte los documentos siguientes:

- Registro de puntos de enlace en la aplicación en la documentación de Android SDK.
- Registro de puntos de enlace en la aplicación en la documentación de iOS SDK.

Registre puntos finales con la biblioteca AWS Amplify JavaScript

Puedes usar la JavaScript biblioteca AWS Amplify para registrar y actualizar los puntos finales de tus aplicaciones. Para obtener más información y ver ejemplos de código, consulta Actualizar punto final en la documentación de AWS Amplify JavaScript .

Siguientes pasos

Después de la actualización de la aplicación para registrar puntos de conexión, se proporciona información sobre los dispositivos y atributos personalizados a Amazon Pinpoint cuando los usuarios

lanzan la aplicación. Puede utilizar esta información para definir segmentos de público. También puede usar la consola para ver las métricas de los puntos finales y los usuarios a los que se les ha asignado un usuario IDs. También puede seguir los pasos que se indican en Notificar eventos de Amazon Pinpoint en la aplicación para actualizar la aplicación de tal forma que notifique los datos de uso.

Notificar eventos de Amazon Pinpoint en la aplicación

En su aplicación móvil o web, puede usar AWS Mobile SDKs o la API de eventos de Amazon Pinpoint para reportar datos de uso o eventos a Amazon Pinpoint. Puede notificar eventos para capturar información como las horas de las sesiones de la aplicación, el comportamiento de compra de los usuarios, los intentos de inicio de sesión o cualquier tipo de evento personalizado que necesite.

Una vez que la aplicación informe de los eventos, puede consultar los análisis en la consola de Amazon Pinpoint. Los gráficos de la página Analytics (Análisis) proporcionan métricas para muchos aspectos de comportamiento de los usuarios. Para obtener más información, consulte la referencia de gráficos para los análisis de Amazon Pinpoint en la Guía del usuario de Amazon Pinpoint.

Para analizar y almacenar los datos de eventos fuera de Amazon Pinpoint, puede configurar Amazon Pinpoint para que transmita los datos a Amazon Kinesis. Para obtener más información, consulte Transmitir datos de eventos de aplicación a través de Kinesis y Firehose con Amazon Pinpoint.

Al utilizar las JavaScript bibliotecas AWS Mobile SDKs y AWS Amplify, puede llamar a la API Amazon Pinpoint para informar de los siguientes tipos de eventos:

Eventos de sesión

Indican cuándo y con qué frecuencia los usuarios abren y cierran la aplicación.

Después de que la aplicación informe de los eventos de sesión, use la página Análisis en la consola de Amazon Pinpoint para ver los gráficos para Sesiones, Puntos de conexión activos diariamente, Tasas de permanencia de 7 días, etc.

Eventos personalizados

Son eventos no estándar que se definen asignando un tipo de evento personalizado. Es posible añadir métricas y atributos personalizados a un evento personalizado.

En la página Analytics (Análisis) de la consola, la pestaña Events (Eventos) muestra métricas para todos los eventos personalizados notificados por la aplicación.

Eventos de monetización

Informan sobre los ingresos que genera la aplicación y el número de elementos que compran los usuarios.

En la página Análisis, la pestaña Ingresos muestra gráficos para Ingresos, Usuarios de pago, Unidades vendidas, etc.

Eventos de autenticación

Indican la frecuencia con la que los usuarios se autentican en la aplicación.

En la página Analytics (Análisis), la pestaña Users (Usuarios) muestra gráficos para Sign-ins (Inicios de sesión), Sign-ups (Inscripciones) y Authentication failures (Errores de autenticación).

Antes de empezar

Si aún no lo ha hecho, haga lo siguiente:

- Integre su aplicación con AWS Amplify. Consulte Conecte su aplicación de interfaz a Amazon AWS Pinpoint mediante Amplify.
- Actualice la aplicación para registrar puntos de enlace. Consulte Registrar puntos de conexión de Amazon Pinpoint en su aplicación.

Reporta eventos con el AWS móvil SDKs para Android o iOS

Puede activar una aplicación móvil para informar de eventos a Amazon Pinpoint mediante la aplicación AWS móvil para SDKs iOS y Android.

Para obtener más información sobre cómo actualizar su aplicación para grabar y enviar eventos a Amazon Pinpoint, consulte las siguientes páginas de la documentación de Amplify AWS :

- Analytics en la documentación de iOS SDK
- Analytics en la documentación de Android SDK

Reporta eventos con la biblioteca AWS Amplify JavaScript

Puede habilitar JavaScript las aplicaciones nativas de React para que notifiquen los eventos de uso de las aplicaciones a Amazon Pinpoint mediante la biblioteca Amplify AWS . JavaScript Para obtener

más información sobre cómo actualizar su aplicación para enviar eventos a Amazon Pinpoint, consulte Analytics en la documentación de Amplify AWS . JavaScript

Notificar eventos con la API de Amazon Pinpoint

Puede utilizar la API de Amazon Pinpoint o un AWS SDK para enviar eventos a Amazon Pinpoint de forma masiva. Para obtener más información, consulte Eventos en la Referencia de la API de Amazon Pinpoint.

Siguientes pasos

Después de actualizar la aplicación para notificar eventos, esta enviará datos de uso a Amazon Pinpoint. Puede ver estos datos en la consola y transmitirlos a Amazon Kinesis. También puede actualizar la aplicación para controlar las notificaciones push que envíe con Amazon Pinpoint. Para obtener más información, consulte los siguientes temas en la Guía del usuario de Mensajes push para usuarios finales de AWS.

- Configuración de las notificaciones push
- Configuración de notificaciones push rápidas
- Configuración de notificaciones push para Android
- Configuración de notificaciones push para Flutter
- Configuración de notificaciones push para React Native
- Crear un proyecto
- Gestión de notificaciones push

Enviar mensajes transaccionales desde la aplicación mediante Amazon Pinpoint

Puede usar la API Amazon Pinpoint y AWS SDKs enviar mensajes transaccionales directamente desde su aplicación. Los mensajes transaccionales son mensajes que envía a destinatarios específicos, en lugar de mensajes que se envían a los segmentos. Existen varios motivos por los que es posible que desee enviar mensajes de transacciones en lugar de mensajes basados en campañas. Por ejemplo, puede enviar una confirmación por correo electrónico cuando un cliente realiza un pedido. También puede enviar una contraseña de un solo uso por SMS o voz que un cliente puede utilizar para completar el proceso de creación de una cuenta para su servicio.

En esta sección se incluye código de ejemplo en varios lenguajes de programación que puede utilizar para iniciar el envío de correos electrónicos transaccionales, mensajes SMS y mensajes de voz.

Para obtener más ejemplos de código sobre puntos de conexión, segmentos y canales, consulte Ejemplos de código.

Temas de esta sección:

- Enviar correos electrónicos transaccionales mediante Amazon Pinpoint
- Envío de mensajes SMS transaccionales mediante Amazon Pinpoint
- Enviar mensajes de voz mediante Amazon Pinpoint

Enviar correos electrónicos transaccionales mediante Amazon Pinpoint

En esta sección se proporcionan ejemplos de código completos que puede utilizar para enviar mensajes de correo electrónico transaccionales a través de Amazon Pinpoint:

- Mediante la SendMessages operación de la API de Amazon Pinpoint: puede utilizar la `SendMessages` operación de la API de Amazon Pinpoint para enviar mensajes en todos los canales compatibles con Amazon Pinpoint, incluidos los canales de notificaciones push, SMS, voz y correo electrónico.

  La ventaja de utilizar esta operación es que la sintaxis de solicitud para enviar mensajes es muy similar en todos los canales. De ese modo, resulta más sencillo replanificar el código existente.

La `SendMessages` operación también le permite sustituir el contenido de sus mensajes de correo electrónico y enviar correos electrónicos al punto de conexión Amazon Pinpoint IDs en lugar de a direcciones de correo electrónico específicas.

En esta sección se incluye código de ejemplo en varios lenguajes de programación que puede utilizar para iniciar el envío de correos electrónicos transaccionales.

Para obtener más ejemplos de código sobre puntos de conexión, segmentos y canales, consulte Ejemplos de código.

Elegir un método de envío de correos electrónicos

El mejor método que se puede utilizar para enviar correos electrónicos transaccionales depende del caso de uso. Por ejemplo, si necesita enviar correos electrónicos mediante una aplicación de terceros o si no hay un AWS SDK disponible para su lenguaje de programación, puede que tenga que usar la interfaz SMTP. Si desea enviar mensajes en otros canales admitidos por Amazon Pinpoint y desea utilizar código coherente para realizar esas solicitudes, debe utilizar la operación `SendMessages` de la API de Amazon Pinpoint.

Elegir entre Amazon Pinpoint y Amazon SES

Si envía un gran número de correos electrónicos transaccionales, como confirmaciones de compra o mensajes de restablecimiento de contraseña, considere la posibilidad de utilizar Amazon SES. Amazon SES cuenta con una API y una interfaz SMTP que son adecuadas para enviar correo electrónico desde las aplicaciones o servicios. También dispone de otras funciones de correo electrónico, como características para la recepción de mensajes, conjuntos de configuración y funciones de autorización de envío.

Amazon SES cuenta, además, con una interfaz SMTP que se puede integrar con aplicaciones de terceros existentes; por ejemplo, con servicios de gestión de relaciones con el cliente (CRM), como Salesforce. Para obtener más información sobre el envío de correo electrónico a través de Amazon SES, consulte la Guía del desarrollador de Amazon Simple Email Service.

Envío de correo electrónico mediante la API de Amazon Pinpoint

En esta sección se incluyen ejemplos de código completos que puede utilizar para enviar correos electrónicos a través de la API de Amazon Pinpoint mediante un AWS SDK. Deberá verificar una dirección de correo electrónico o un dominio antes de poder enviar correos electrónicos.

C#

Utilice este ejemplo para enviar un correo electrónico mediante AWS SDK para .NET. En este ejemplo se presupone que ya ha instalado y configurado el SDK para .NET. Para obtener más información, consulte Introducción a AWS SDK para .NET en la Guía para desarrolladores de AWS SDK para .NET.

En este ejemplo se supone que utiliza un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario existente. Para obtener más información, consulte Configuración de credenciales de AWS en la Guía para desarrolladores de AWS SDK para .NET.

Este ejemplo de código se ha probado utilizando la versión 3.3.29.13 de AWS SDK para .NET y la versión 2.1.2 de tiempo de ejecución de .NET Core.

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

namespace SendEmailMessage;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Java

Utilice este ejemplo para enviar un correo electrónico mediante AWS SDK para Java. En este ejemplo se presupone que ya ha instalado y configurado el AWS SDK for Java 2.x. Para obtener más información, consulte Introducción en la Guía para desarrolladores de AWS SDK for Java 2.x.

En este ejemplo se supone que utiliza un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario existente. Para obtener más información, consulte Configuración de credenciales y regiones predeterminadas en la Guía para desarrolladores de AWS SDK para Java.

Este ejemplo de código se ha probado utilizando la versión 2.3.1 de AWS SDK para Java y la versión 11.0.1 de 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);
        }
    }
```

```
}
```

Para ver el ejemplo completo del SDK, consulte SendEmailMessage.java en GitHub.

JavaScript (Node.js)

Use este ejemplo para enviar un correo electrónico mediante el AWS SDK para JavaScript en Node.js. En este ejemplo se presupone que ya ha instalado y configurado SDK para JavaScript en Node.js. Para obtener más información, consulte Introducción en la Guía para desarrolladores de AWS SDK para JavaScript en Node.js.

En este ejemplo se supone que utiliza un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario existente. Para obtener más información, consulte Configuración de credenciales en la Guía para desarrolladores de AWS SDK para JavaScript en Node.js.

Este ejemplo de código se ha probado utilizando la versión 2.388.0 de Node.jd y la versión 11.7.0 de Node.js.

```
"use strict";

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Python

Utilice este ejemplo para enviar un correo electrónico mediante AWS SDK para Python (Boto3). En este ejemplo se presupone que ya ha instalado y configurado el SDK para Python (Boto3). Para obtener más información, consulte Quickstart en la Referencia de la API del AWS SDK para Python (Boto3).

```
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()
```

También puede usar plantillas de mensajes para enviar mensajes de correo electrónico, como se muestra en el siguiente ejemplo:

```
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()
```

En estos ejemplos se supone que está utilizando un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario existente. Para obtener más información, consulte Credenciales en la Referencia de la API del AWS SDK para Python (Boto3).

Agregar encabezados de cancelación de suscripción al correo electrónico mediante Amazon Pinpoint

> **Note**
>
> Antes de poder usar encabezados de correo electrónico, debe configurar un rol de envío de orquestación de correo electrónico si va a enviar correos electrónicos desde una campaña o un recorrido. Para correos electrónicos de envío directo, debe tener permisos para `ses:SendEmail` y `ses:SendRawEmail`. Para obtener más información, consulte Creating an email orchestration sending role en la Guía del usuario de Amazon Pinpoint.

Incluir un enlace para cancelar la suscripción en el correo electrónico es una práctica recomendada y, en algunos países, es un requisito legal. Para agregar un enlace para cancelar la suscripción con un solo clic, añada los siguientes encabezados:

1. Establezca el Nombre del encabezado como `List-Unsubscribe` y en Valor ingrese el enlace para cancelar la suscripción. El enlace debe admitir solicitudes HTTP POST para procesar la solicitud de cancelación de suscripción de los destinatarios.

2. Establezca el nombre del encabezado como `List-Unsubscribe-Post` y el valor en `List-Unsubscribe=One-Click`.

Puede agregar hasta 15 encabezados a un mensaje de correo electrónico. Para obtener una lista de los encabezados compatibles, consulte Amazon SES header fields en la Guía para desarrolladores de Amazon Simple Email Service.

En el siguiente ejemplo se muestra cómo enviar un mensaje de correo electrónico con encabezados de cancelación de suscripción mediante la AWS Command Line Interface. Para obtener más información sobre la configuración de AWS CLI, consulte Configurar el AWS CLI en la Guía del AWS Command Line Interface usuario.

En el siguiente comando, haga esto:

- *AppId*Sustitúyalo por el identificador de la aplicación.
- *richard_roe@example.com*Sustitúyalo por la dirección de correo electrónico del destinatario.
- *https://example.com/unsub*Sustitúyelo por el enlace para cancelar la suscripción.
- *example123456*Sustitúyalo por un identificador único para el destinatario.

```
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"
          }
        ]
      }
    }
  }
}'
```

Envío de mensajes SMS transaccionales mediante Amazon Pinpoint

Puede utilizar la API de Amazon Pinpoint para enviar mensajes SMS (mensajes de texto) a determinados números de teléfono o ID de punto de conexión. En esta sección se incluyen ejemplos de código completos que puede utilizar para enviar mensajes SMS a través de la API de Amazon Pinpoint mediante un AWS SDK. Su cuenta debe estar en producción y usted debe tener una identidad de origen activa que pueda enviar mensajes SMS.

Para obtener más ejemplos de código sobre puntos de conexión, segmentos y canales, consulte Ejemplos de código.

C#

Utilice este ejemplo para enviar un mensaje SMS mediante el AWS SDK para .NET. En este ejemplo se presupone que ya ha instalado y configurado el SDK para .NET. Para obtener más información, consulte Introducción en la Guía para desarrolladores de AWS SDK para .NET.

En este ejemplo se supone que está utilizando un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario de IAM existente.

Para obtener más información, consulte Configuración de credenciales de AWS en la Guía para desarrolladores de AWS SDK para .NET.

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

namespace SendSmsMessage;

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

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

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

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

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

        // The type of SMS message that you want to send. If you plan to send
```

```
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;

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

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

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
 {response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
 {messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
 originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";
```

```
        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

Java

Utilice este ejemplo para enviar un mensaje SMS mediante el AWS SDK para Java. En este ejemplo se presupone que ya ha instalado y configurado el SDK para Java. Para obtener más información, consulte Introducción en la Guía para desarrolladores de AWS SDK para Java.

En este ejemplo se supone que está utilizando un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario de IAM

existente. Para obtener más información, consulte Configuración de credenciales y regiones predeterminadas en la Guía para desarrolladores de AWS SDK para Java.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

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

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
```

```
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

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

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

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application ID
 to use when you send this message.
                                  originationNumber - The phone number or short code
 that you specify has to be associated with your Amazon Pinpoint account. For best
 results, specify long codes in E.164 format (for example, +1-555-555-5654).
                                  destinationNumber - The recipient's phone number.
 For best results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).\s
                                  """;

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

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();
```

```
                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request = SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();
```

```
                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" + v)));

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

Para ver el ejemplo completo del SDK, consulte SendMessage.java en GitHub.

JavaScript (Node.js)

Use este ejemplo para enviar un mensaje SMS mediante el AWS SDK para JavaScript en Node.js. En este ejemplo se presupone que ya ha instalado y configurado SDK para JavaScript en Node.js. Para obtener más información, consulte Introducción en la Guía para desarrolladores de AWS SDK para JavaScript en Node.js.

En este ejemplo se supone que está utilizando un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario de IAM existente. Para obtener más información, consulte Configuración de credenciales en la Guía para desarrolladores de AWS SDK para JavaScript en Node.js.

```
"use strict";

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

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

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";
```

```
// The recipient's phone number.  For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

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

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

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

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

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

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

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

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
```

```
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

Python

Utilice este ejemplo para enviar un mensaje SMS mediante el AWS SDK para Python (Boto3). En este ejemplo se presupone que ya ha instalado y configurado el SDK para Python. Para obtener más información, consulte Inicio rápido en Introducción a AWS SDK para Python (Boto3).

```
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()
```

También puede usar plantillas de mensajes para enviar mensajes SMS, como se muestra en el siguiente ejemplo:

```
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()
```

En estos ejemplos se supone que está utilizando un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario de IAM existente. Para obtener más información, consulte Credenciales en la Referencia de la API del AWS SDK para Python (Boto3).

Enviar mensajes de voz mediante Amazon Pinpoint

Puede utilizar la API de Amazon Pinpoint para enviar mensajes de voz a determinados números de teléfono. En esta sección se incluyen ejemplos de código completos que puede utilizar para enviar mensajes de voz a través de la API de SMS y voz de Amazon Pinpoint mediante AWS SDK. Su cuenta debe estar en producción y usted debe tener una identidad de origen activa que pueda enviar mensajes de voz.

Para obtener más ejemplos de código sobre puntos de conexión, segmentos y canales, consulte Ejemplos de código.

Java

Utilice este ejemplo para enviar un mensaje de voz mediante AWS SDK para Java. En este ejemplo se presupone que ya ha instalado y configurado el SDK para Java. Para obtener más información, consulte Introducción en la Guía para desarrolladores de AWS SDK para Java.

En este ejemplo se supone que utiliza un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario existente. Para obtener más información, consulte Configurar las credenciales y la región de AWS para el desarrollo en la Guía para desarrolladores de AWS SDK para 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.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";

    // The language to use when sending the message. For a list of supported
    // languages, see
    // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
    static final String languageCode = "en-US";

    // The content of the message. This example uses SSML to customize and control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
    static final String ssmlMessage = "<speak>This is a test message sent from "
            + "<emphasis>Amazon Pinpoint</emphasis> "
            + "using the <break strength='weak'/>AWS "
            + "SDK for Java. "
            + "<amazon:effect phonation='soft'>Thank "
            + "you for listening.</amazon:effect></speak>";

    public static void main(String[] args) {

        final String usage = """
                Usage:   <originationNumber> <destinationNumber>\s

                Where:
                  originationNumber - The phone number or short code that you
 specify has to be associated with your Amazon Pinpoint account. For best results,
 specify long codes in E.164 format (for example, +1-555-555-5654).
```

```
                destinationNumber - The recipient's phone number.  For best
results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
               """;

       if (args.length != 2) {
           System.out.println(usage);
           System.exit(1);
       }
       String originationNumber = args[0];
       String destinationNumber = args[1];
       System.out.println("Sending a voice message");

       // Set the content type to application/json.
       List<String> listVal = new ArrayList<>();
       listVal.add("application/json");
       Map<String, List<String>> values = new HashMap<>();
       values.put("Content-Type", listVal);

       ClientOverrideConfiguration config2 = ClientOverrideConfiguration.builder()
               .headers(values)
               .build();

       PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
               .overrideConfiguration(config2)
               .region(Region.US_EAST_1)
               .build();

       sendVoiceMsg(client, originationNumber, destinationNumber);
       client.close();
   }

   public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                                   String destinationNumber) {
       try {
           SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                   .languageCode(languageCode)
                   .text(ssmlMessage)
                   .voiceId(voiceName)
                   .build();

           VoiceMessageContent content = VoiceMessageContent.builder()
                   .ssmlMessage(ssmlMessageType)
```

```
                    .build();

            SendVoiceMessageRequest voiceMessageRequest =
 SendVoiceMessageRequest.builder()
                    .destinationPhoneNumber(destinationNumber)
                    .originationPhoneNumber(originationNumber)
                    .content(content)
                    .build();

            client.sendVoiceMessage(voiceMessageRequest);
            System.out.println("The message was sent successfully.");

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

Para ver el ejemplo completo del SDK, consulte SendVoiceMessage.java en GitHub.

JavaScript (Node.js)

Use este ejemplo para enviar un mensaje de voz mediante el AWS SDK para JavaScript en Node.js. En este ejemplo se presupone que ya ha instalado y configurado SDK para JavaScript en Node.js.

En este ejemplo se supone que utiliza un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario existente. Para obtener más información, consulte Configuración de credenciales en la Guía para desarrolladores de AWS SDK para JavaScript en 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

Utilice este ejemplo para enviar un mensaje de voz mediante AWS SDK para Python (Boto3). En este ejemplo se presupone que ya ha instalado y configurado el SDK para Python (Boto3).

En este ejemplo se supone que utiliza un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario existente. Para obtener más información, consulte Credenciales en la Referencia de la API del AWS SDK para Python (Boto3).

```
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()
```

Utilice la API de mensajería SMS y voz para el usuario AWS final, versión 2

Amazon Pinpoint incluye una API (denominada API de SMS y voz, versión 2) que se ha diseñado para enviar mensajes SMS y de voz. Mientras la API de Amazon Pinpoint se centra en enviar mensajes a través de campañas y recorridos programados y basados en eventos, la API de SMS y voz ofrece nuevas características y capacidades para enviar mensajes SMS y de voz directamente a destinatarios individuales. Puede usar la API de SMS y voz independientemente de las características de campaña y recorrido de Amazon Pinpoint o puede usar ambas al mismo tiempo para adaptarse a diferentes casos de uso. Si ya usa Amazon Pinpoint para enviar mensajes SMS o de voz, la cuenta ya está configurada para usar esta API.

Esta API es una buena solución para los usuarios que tienen una arquitectura multiusuario, como los proveedores de software independientes (ISVs). Esta API facilita garantizar que los datos de los eventos, los números de teléfono de origen y las listas de exclusión estén separados para los distintos inquilinos.

Cuando utilice la API de SMS y voz, le recomendamos que configure los conjuntos de configuración y los destinos de los eventos. La API de SMS y voz no emite automáticamente datos de eventos para los mensajes que envía. La configuración de los destinos de los eventos garantiza la captura de datos importantes sobre los eventos, como la entrega de mensajes y los eventos de error.

La versión 1 precedió a la versión 2 de esta API. Si actualmente utiliza la versión 1 de esta API, seguirá estando disponible y podrá seguir utilizándola. Si migra a la versión 2, obtendrá características adicionales, como la posibilidad de crear grupos de números de teléfono, solicitar nuevos números de teléfono mediante programación y habilitar o desactivar determinadas funciones de los números de teléfono.

> **Note**
>
> Hay algunas tareas que actualmente solo se pueden completar con la consola de Amazon Pinpoint. Por ejemplo, verificar un número de teléfono para usarlo mientras la cuenta está en el entorno de pruebas de SMS y registrarse para usar 10DLC.

Para obtener más información acerca de la versión 2 de la API de SMS y voz de Amazon Pinpoint, consulte la referencia de la API de SMS y voz, versión 2. Para obtener información sobre cómo crear,

configurar y administrar sus recursos de mensajería, SMS y voz para usuarios AWS finales, consulte la Guía del usuario de mensajes SMS para usuarios AWS finales

Genere contraseñas de un solo uso (OTPs) con Amazon Pinpoint

Amazon Pinpoint incluye una característica de administración de contraseñas de un solo uso (OTP) que puede utilizar para generar contraseñas nuevas de un solo uso y enviarlas a los destinatarios como mensajes SMS.

> **Important**
>
> Para utilizar esta característica, la cuenta debe tener acceso a producción y una identidad de origen activa. Para obtener más información, consulte Acerca del entorno limitado de SMS/MMS and Voice y Solicitud de un número de teléfono en la Guía del usuario AWS final de mensajería por SMS.

En algunos países y regiones, debe obtener un número de teléfono dedicado o un ID de origen antes de poder enviar mensajes SMS. Por ejemplo, cuando envía mensajes a destinatarios en Estados Unidos, debe tener un número gratuito dedicado, un número 10DLC o un código corto. Cuando envíe mensajes a destinatarios en India, debe tener un ID de remitente registrado, que incluya un ID de entidad principal (PEID) y un ID de plantilla. Se siguen aplicando estos requisitos cuando utiliza la característica de OTP.

Para utilizar esta característica, necesita permisos para enviar y verificar mensajes de OTP, consulte Contraseñas únicas. Si necesita ayuda para determinar los permisos, consulte Solución de problemas de la administración de identidades y accesos de Amazon Pinpoint.

Puede utilizar la operación `SendOtpMessages` en la API de Amazon Pinpoint para enviar un código de OTP a un usuario de la aplicación. Cuando utiliza esta API, Amazon Pinpoint genera un código aleatorio y se lo envía al usuario como mensaje SMS. La solicitud puede incluir los siguientes parámetros:

- `Channel`: el canal de comunicación por el que se envía el código de OTP. Actualmente, solo se admiten los mensajes SMS, por lo que el único valor aceptable es el SMS.
- `BrandName`: el nombre de la marca, empresa o producto asociado al código de OTP. Este nombre puede contener hasta 20 caracteres.

> **Note**
>
> Cuando Amazon Pinpoint envía el mensaje de OTP, el nombre de la marca se inserta automáticamente en la siguiente plantilla de mensaje:
>
> ```
> This is your One Time Password: {{otp}} from {{brand}}
> ```
>
> Por lo tanto, si especificas ExampleCorp el nombre de tu marca y Amazon Pinpoint genera una contraseña de un solo uso llamada 123456, enviará el siguiente mensaje a tu usuario:
>
> ```
> This is your One Time Password: 123456 from ExampleCorp
> ```

- `CodeLength`: el número de dígitos que habrá en el código de OTP que se envía al destinatario. Los códigos de OTP pueden contener entre 5 y 8 dígitos, ambos incluidos.
- `ValidityPeriod`: la cantidad de tiempo, en minutos, durante el que el código de OTP será válido. El periodo de validez puede oscilar entre 5 y 60 minutos, ambos incluidos.
- `AllowedAttempts`: el número de veces que el destinatario puede intentar verificar la OTP sin éxito. Si el número de intentos supera este valor, la OTP deja de ser válida automáticamente. El número máximo de intentos permitidos es 5.
- `Language`: el idioma, en formato IETF BCP-47, que se utilizará al enviar el mensaje. Los valores aceptables son:
  - `de-DE`: alemán
  - `en-GB`: inglés (Reino Unido)
  - `en-US`: inglés (EE. UU.)
  - `es-419`: español (Latinoamérica)
  - `es-ES`: español
  - `fr-CA`: francés (Canadá)
  - `fr-FR`: francés
  - `it-IT`: italiano
  - `ja-JP`: japonés
  - `ko-KR`: coreano
  - `pt-BR`: portugués (Brasil)
  - `zh-CN`: chino (simplificado)

  - `zh-TW`: chino (tradicional)
- `OriginationIdentity`: la identidad de origen (como un código largo, un código corto o un ID del remitente) que se utiliza para enviar el código de OTP. Si utiliza un código largo o un número gratuito para enviar la OTP, el número de teléfono debe tener el formato E.164.
- `DestinationIdentity`: el número de teléfono en formato E.164 al que se envió el código de OTP.
- `ReferenceId`: un ID de referencia único para la solicitud. El ID de referencia coincide exactamente con el ID de referencia que proporciona al verificar la OTP. El ID de referencia puede contener entre 1 y 48 caracteres, ambos incluidos.
- `EntityId`: un ID de entidad registrado con una agencia reguladora. Actualmente, este parámetro solo se usa cuando se envían mensajes a destinatarios en India. Si no envía a destinatarios en India, puede omitir este parámetro.
- `TemplateId`: un ID de plantilla registrado con una agencia reguladora. Actualmente, este parámetro solo se usa cuando se envían mensajes a destinatarios en India. Si no envía a destinatarios en India, puede omitir este parámetro.

> **Note**
>
> Para obtener más información sobre cómo enviar mensajes SMS a destinatarios en la India, consulte India sender ID registration process en la Guía del usuario de Amazon Pinpoint.

Para asegurarse de que su cuenta de Amazon Pinpoint está configurada correctamente para enviar mensajes OTP, puede utilizarla AWS CLI para enviar un mensaje de prueba. Para obtener más información al respecto AWS CLI, consulte la Guía del AWS Command Line Interface usuario.

Para enviar un mensaje OTP de prueba mediante el AWS CLI, ejecute el send-otp-messagecomando en la 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
```

En el comando anterior, haga lo siguiente.

- *7353f53e6885409fa32d07cedexample*Sustitúyalo por el identificador de tu aplicación.

- *ExampleCorp*Sustitúyalo por el nombre de su empresa.
- *5*Sustitúyelo por el número de dígitos que habrá en el código OTP que se envía al destinatario. `CodeLegth`
- *20*Sustitúyalo por el tiempo, en minutos, durante el que el código OTP será válido. `ValidityPeriod`
- *5*Sustitúyalo por el número de veces que el destinatario puede intentar verificar la OTP sin éxito. `AllowedAttempts`
- *+18555550142*`OriginationIdentity`Sustitúyalo por la identidad de origen que se utiliza para enviar el código OTP.
- *+12065550007*`DestinationIdentity`Sustitúyalo por el número de teléfono al que se va a enviar el código OTP.
- *SampleReferenceId*`ReferenceId`Sustitúyalo por un identificador de referencia único para la solicitud.

Respuesta de **SendOtpMessage**

Cuando envía correctamente un mensaje de OTP, recibe una respuesta similar al siguiente ejemplo:

```
{
    "MessageResponse": {
        "ApplicationId": "7353f53e6885409fa32d07cedexample",
        "RequestId": "255d15ea-75fe-4040-b919-096f2example",
        "Result": {
            "+12065550007": {
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "nvrmgq9kq4en96qgp0tlqli3og1at6aexample",
                "StatusCode": 200,
                "StatusMessage": "MessageId: nvrmgq9kq4en96qgp0tlqli3og1at6aexample"
            }
        }
    }
}
```

Validar mensajes de OTP en Amazon Pinpoint

Después de enviar un one-time-password, la aplicación puede llamar a la API Amazon Pinpoint para verificarlo. Para verificar un código de OTP, llame a la API `VerifyOtpMessages`. La solicitud debe incluir los siguientes parámetros:

- `DestinationIdentity`: el número de teléfono en formato E.164 al que se envió el código de OTP.
- `ReferenceId`: el ID de referencia que utilizó al enviar el código de OTP al destinatario. El ID de referencia debe coincidir de forma exacta.
- `Otp`: el código de OTP que está validando.

Puede utilizarla AWS CLI para probar el proceso de validación. Para obtener más información sobre la instalación y configuración del AWS CLI, consulte la Guía del AWS Command Line Interface usuario.

Para verificar una OTP mediante la AWS CLI, ejecute el verify-otp-messagecomando en la terminal:

```
aws pinpoint verify-otp-message --application-id 7353f53e6885409fa32d07cedexample --
verify-otp-message-request-parameters
 DestinationIdentity=+12065550007,ReferenceId=SampleReferenceId,Otp=01234
```

En el comando anterior, haga lo siguiente.

- *7353f53e6885409fa32d07cedexample*Sustitúyala por el identificador de tu aplicación.
- *+12065550007*`DestinationIdentity`Sustitúyalo por el número de teléfono al que se envió el código OTP.
- *SampleReferenceId*`ReferenceId`Sustitúyalo por un identificador de referencia único para la solicitud. Este valor debe coincidir con el `ReferenceID` que se utilizó para enviar la solicitud.
- *01234*Sustitúyalo por un Otp que se envió al`DestinationIdentity`. `Otp`

Respuesta de **`VerifyOtpMessage`**

Cuando envía una solicitud a la API de `VerifyOTPMessage`, esta devuelve un objeto `VerificationResponse`, que contiene una sola propiedad, `Valid`. Si el ID de referencia, el número de teléfono y la OTP coinciden con los valores que Amazon Pinpoint espera y si la OTP no

ha caducado, el valor de `Valid` es `true`; de lo contrario, es `false`. A continuación, se muestra un ejemplo de una respuesta para una verificación de la OTP correcta:

```
{
    "VerificationResponse": {
        "Valid": true
    }
}
```

Ejemplos de Amazon Pinpoint para usar SDK para Python (Boto3) en Amazon Pinpoint

Esta sección contiene ejemplos de código que muestran cómo usar el SDK para Python (Boto3) para enviar y verificar códigos de OTP.

Generar un ID de referencia

La siguiente función genera un ID de referencia único para cada destinatario, basado en el número de teléfono del destinatario, el producto o la marca para el que el destinatario recibe una OTP y el origen de la solicitud (que podría ser el nombre de una página de un sitio o una aplicación, por ejemplo). Al verificar el código de OTP, debe pasar un ID de referencia idéntico para que la validación se realice correctamente. Los ejemplos de código de envío y validación utilizan esta función de utilidad.

Esta función no es obligatoria, pero es una forma útil de limitar el proceso de envío y verificación de la OTP a una transacción específica, de forma que se pueda volver a enviar fácilmente durante el paso de verificación. Puede usar cualquier ID de referencia que desee; este es solo un ejemplo básico. Sin embargo, los demás ejemplos de código de esta sección se basan en esta función.

```
# 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()
```

Enviar códigos de OTP

El siguiente ejemplo de código muestra cómo utilizar el SDK para Python (Boto3) para enviar un código de 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")
```

Validar códigos de OTP

El siguiente ejemplo de código muestra cómo utilizar el SDK para Python (Boto3) para verificar un código de OTP que ya ha enviado. Para que el paso de validación se realice correctamente, la solicitud debe incluir un ID de referencia que coincida exactamente con el ID de referencia que se utilizó para enviar el mensaje.

```
# 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")
```

Personalizar mensajes integrados en la aplicación con Amazon Pinpoint y Amplify

Puede usar los mensajes en la aplicación para enviar mensajes dirigidos a los usuarios de las aplicaciones. Los mensajes en la aplicación son altamente personalizables. Pueden incluir botones que abren sitios web o llevan a los usuarios a partes específicas de la aplicación. Puede configurar los colores de fondo y del texto, colocar el texto y agregar botones e imágenes a la notificación. Puede enviar un solo mensaje o crear un carrusel que contenga hasta cinco mensajes únicos. Para obtener información general de los mensajes integrados en la aplicación, incluidas las instrucciones para crear plantillas de mensajes integrados en la aplicación, consulte Creación de plantillas integradas en la aplicación en la Guía del usuario de Amazon Pinpoint.

Puede utilizarlo AWS Amplify para integrar sin problemas las funciones de mensajería integradas en la aplicación de Amazon Pinpoint en su aplicación. Amplify puede gestionar automáticamente los procesos de obtención de mensajes, procesamiento de mensajes y envío de datos analíticos a Amazon Pinpoint. Actualmente, esta integración es compatible con las aplicaciones de React Native. Para obtener más información, consulte Mensajería en la aplicación en la documentación de Amplify Framework.

Recuperar mensajes integrados en la aplicación para un punto de conexión mediante programación con Amazon Pinpoint

Sus aplicaciones pueden llamar a la GetInAppMessagesAPI para recuperar todos los mensajes integrados en la aplicación a los que tiene derecho un punto final determinado. Cuando se llama a la API de `GetInAppMessages`, se proporcionan los siguientes parámetros:

- `ApplicationId`: el ID único del proyecto de Amazon Pinpoint con el que está asociada la campaña del mensaje integrada en la aplicación.
- `EndpointId`: el ID único del punto de conexión para el que estás recuperando los mensajes.

Cuando se llama a la API con estos valores, devuelve una lista de mensajes. Para obtener más información acerca de la respuesta producida por esta operación, consulte Ejemplo de JSON de respuesta de la API `GetInAppMessages` de Amazon Pinpoint.

Puede utilizarla AWS SDKs para llamar a la `GetInAppMessages` operación. Los siguientes ejemplos de código incluyen funciones que recuperan mensajes integrados en la aplicación.

JavaScript

Cree el cliente en un módulo separado y expórtelo:

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
const REGION = "us-east-1";
const pinClient = new PinpointClient({ region: REGION });
export { pinClient };
```

Recupere mensajes dentro de la aplicación para un punto de conexión:

```
// 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()
```

Ejemplo de JSON de respuesta de la API **GetInAppMessages** de Amazon Pinpoint

Cuando llamas a la operación de la GetInAppMessagesAPI, devuelve una lista de mensajes a los que tiene derecho el punto final especificado. Después, la aplicación puede procesar el mensaje en función de los valores de la respuesta.

A continuación, se ofrece un ejemplo del objeto JSON que se devuelve cuando llame a la API de `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"
      }
    ]
  }
}
```

En las siguientes secciones se proporciona más información sobre los componentes de esta respuesta y sus atributos.

Objeto **InAppMessageCampaigns**

El objeto `InAppMessageCampaigns` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`CampaignId`	Una cadena que contiene el nombre y el ID de campaña exclusivo de la campaña de	Amazon Pinpoint la crea automáticamente al crear la campaña.

Atributo	Description (Descripción)	Dónde tiene lugar
	Amazon Pinpoint desde la que se envió el mensaje. El nombre precede al ID de la campaña. Los dos valores se separan con un guion (-).	
`TreatmentId`	Un número entero que representa el ID del tratamiento de campaña de este mensaje. Si la campaña solo tiene un tratamiento, el valor es `0`.	
`Priority`	La prioridad del mensaje en la aplicación, expresada como un número entero entre 1 y 5, ambos inclusive, donde 1 indica la prioridad más alta y 5 indica la más baja.	Paso 1 del proceso de creación de la campaña.
`InAppMessage`	Y Objeto `InAppMessage` que contiene información sobre cómo se representa el mensaje.	En función del contenido de la plantilla de mensajes integrada en la aplicación que se especificó para la campaña.
`Schedule`	Un objeto de programación que contiene información sobre cuándo se envió el mensaje.	El paso 4 del proceso de creación de la campaña (si la campaña se creó en la consola) o el objeto de `Schedule` (si la campaña se creó mediante la API o un SDK).

Atributo	Description (Descripción)	Dónde tiene lugar
`DailyCap`	El número de veces, que se muestra como un número entero, que se puede mostrar al usuario un mensaje integrado en la aplicación durante un periodo de 24 horas.	Se hereda de la configuración a nivel de proyecto. Si la campaña incluye ajustes que invalidan los ajustes del proyecto, se utilizan en su lugar.
`SessionCap`	El número de veces, expresado como un entero, que se puede mostrar al usuario un mensaje integrado en la aplicación durante una sesión de la aplicación.	
`TotalCap`	El número total de veces, expresado como un entero, que se puede mostrar un mensaje integrado en la aplicación a un punto de conexión por campaña.	

Objeto **`InAppMessage`**

El objeto `InAppMessage` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`Content`	Matriz que contiene un InAppMessageContentobjeto que describe el contenido del mensaje.	En función del contenido de la plantilla de mensajes integrada en la aplicación que se especificó para la campaña.
`Layout`	Una cadena que describe cómo aparecerá el mensaje	

Atributo	Description (Descripción)	Dónde tiene lugar
	integrado en la aplicación en el dispositivo del destinatario. Los valores posibles son los que se indican a continuación. • `BOTTOM_BANNER` : un mensaje que aparece como un banner en la parte inferior de la página. • `TOP_BANNER` : un mensaje que aparece como un banner en la parte superior de la página. • `OVERLAYS`: un mensaje que abarca toda la pantalla. • `MOBILE_FEED` : un mensaje que aparece en una ventana delante de la página. • `MIDDLE_BANNER` : un mensaje que aparece como un banner en el centro de la página. • `CAROUSEL`: un formato desplazable de hasta cinco mensajes únicos.	

Objeto **`HeaderConfig`**

El objeto `HeaderConfig` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`Alignment`	Una cadena que especific a la alineación del texto del encabezado. Los posibles valores son `LEFT`, `CENTER` y `RIGHT`.	En función del contenido de la plantilla de mensajes integrada en la aplicación que se especificó para la campaña.
`Header`	El texto del encabezado del mensaje.	
`TextColor`	El color del texto del encabezado, expresado como una cadena que describe el código de color hexadecimal (por ejemplo, “#000000” para el negro).	

Objeto **`BodyConfig`**

El objeto `BodyConfig` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`Alignment`	Una cadena que especifica la alineación del texto del cuerpo del mensaje. Los posibles valores son `LEFT`, `CENTER` y `RIGHT`.	En función del contenido de la plantilla de mensajes integrada en la aplicación que se especificó para la campaña.
`Body`	El texto principal del cuerpo del mensaje.	
`TextColor`	El color del texto del cuerpo, expresado como una cadena que contiene un código de color hexadecimal (por	

Atributo	Description (Descripción)	Dónde tiene lugar
	ejemplo, “#000000” para el color negro).	

Objeto **`InAppMessageContent`**

El objeto `InAppMessageContent` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`BackgroundColor`	El color de fondo del mensaje integrado en la aplicación, expresado como una cadena que contiene un código de color hexadecimal (por ejemplo, “#000000” para el negro).	En función del contenido de la plantilla de mensajes integrada en la aplicación que se especificó para la campaña.
`BodyConfig`	BodyConfigObjeto que contiene información relaciona da con el contenido del cuerpo principal del mensaje.	
`HeaderConfig`	Un HeaderConfigobjeto que contiene información relaciona da con el encabezado o el título del mensaje.	
`ImageUrl`	La URL de la imagen que aparece en el mensaje.	
`PrimaryBtn`	InAppMessageButtonObjeto que contiene información sobre el botón principal del mensaje.	

Atributo	Description (Descripción)	Dónde tiene lugar
`SecondaryBtn`	InAppMessageButtonObjeto que contiene información sobre el botón secundario del mensaje. No está presente si la plantilla de mensaje integrada en la aplicación no especifica un botón secundario.	

Objeto **`Schedule`**

El objeto `Schedule` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`EndDate`	La hora programada, en formato ISO 8601, cuando la campaña terminará.	El paso 4 del proceso de creación de la campaña (si la campaña se creó en la consola) o el objeto de `Schedule` (si la campaña se creó mediante la API o un SDK).
`EventFilter`	Información sobre el evento que provoca que se muestre el mensaje integrado en la aplicación. Cuando genera un evento que coincide con una campaña integrada en la aplicación de Amazon Pinpoint, se muestra el mensaje.	

Objeto **`InAppMessageButton`**

Un objeto `InAppMessageButton` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`DefaultConfig`	DefaultButtonConfigObjeto que contiene informaci ón sobre la configuración predeterminada de un botón en un mensaje integrado en la aplicación.	En función del contenido de la plantilla de mensajes integrada en la aplicación que se especificó para la campaña.
`Android`	OverrideButtonConfigObjeto que especifica el comportam iento del botón en los dispositi vos Android. Esto invalida la configuración de botones predeterminada detallada en el objeto de `DefaultCo nfig` .	
`IOS`	OverrideButtonConfigObjeto que especifica el comportam iento del botón en los dispositi vos iOS. Esto invalida la configuración de botones predeterminada detallada en el objeto de `DefaultCo nfig` .	
`Web`	OverrideButtonConfigObjeto que especifica el comportam iento del botón en las aplicacio nes web. Esto invalida la configuración de botones predeterminada detallada en el objeto de `DefaultCo nfig` .	

Objeto **DefaultButtonConfig**

Un objeto `DefaultButtonConfig` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`BackgroundColor`	El color de fondo del botón, expresado como una cadena que contiene un código de color hexadecimal (por ejemplo, “#000000” para el negro).	En función del contenido de la plantilla de mensajes integrada en la aplicación que se especificó para la campaña.
`BorderRadius`	El radio del borde del botón en píxeles, expresado como un número entero. Un número mayor da como resultado esquinas más redondeadas.	
`ButtonAction`	Una cadena que describe la acción que se produce cuando un destinatario elige un botón en el mensaje integrado en la aplicación. Los valores posibles son los que se indican a continuación. • `LINK`: un enlace a un destino web. • `DEEP_LINK` : un enlace a una página específica de una aplicación. • `CLOSE`: omite el mensaje.	
`Link`	La URL de destino de un botón. No está presente en los	

Atributo	Description (Descripción)	Dónde tiene lugar
	botones donde ButtonAction está`CLOSE`.	
`Text`	El texto que aparece en el botón.	
`TextColor`	El color del texto en el botón, expresado como una cadena que contiene un código de color hexadecimal (por ejemplo, “#000000” para el negro).	

Objeto **`OverrideButtonConfig`**

El objeto `OverrideButtonConfig` solo está presente si la plantilla de mensajes integrada en la aplicación utiliza botones de invalidación. Un botón de invalidación es un botón que tiene una configuración específica para un tipo de dispositivo concreto, como un dispositivo iOS, un dispositivo Android o un navegador web.

Un objeto `OverrideButtonConfig` contiene los siguientes atributos:

Atributo	Description (Descripción)	Dónde tiene lugar
`ButtonAction`	La acción que se produce cuando un destinatario elige un botón en el mensaje en la aplicación. Los valores posibles son los que se indican a continuación. • `LINK`: un enlace a un destino web. • `DEEP_LINK` : un enlace a una página específica de una aplicación.	En función del contenido de la plantilla de mensajes integrada en la aplicación que se especificó para la campaña.

Atributo	Description (Descripción)	Dónde tiene lugar
	• `CLOSE`: omite el mensaje.	
`Link`	La URL de destino de un botón. No está presente para los botones en los que `ButtonAction` es `CLOSE`.	
`Text`	El texto que aparece en el botón.	
`TextColor`	El color del texto en el botón, expresado como una cadena que contiene un código de color hexadecimal (por ejemplo, “#000000” para el negro).	

Utilizar el servicio de validación de números de teléfono de Amazon Pinpoint

Amazon Pinpoint incluye un servicio de validación de números de teléfono que puede utilizar para determinar si un número de teléfono es válido y para obtener información adicional sobre el propio número de teléfono. Por ejemplo, cuando se usa el servicio de validación de números de teléfono, este devuelve la siguiente información:

- El número de teléfono en formato E.164.
- El tipo de número de teléfono (por ejemplo, móvil, fijo o VoIP).
- La ciudad y el país del número de teléfono.
- El proveedor de servicios asociado al número de teléfono.

Se aplica un cargo adicional por el uso del servicio de validación de números de teléfono. Para obtener más información, consulte Precios de Amazon Pinpoint.

> **Important**
>
> Para los números de teléfono originarios de Estados Unidos y Canadá, la API de validación de números de teléfono ya no devolverá los datos de `City`, `County`, `Timezone` y `ZipCode`.

Casos de uso de validación de números de teléfono de Amazon Pinpoint

Puede utilizar el servicio de validación de números de teléfono para varios casos de uso, entre los que se incluyen los siguientes:

- Verificación de los números de teléfono proporcionados en un formulario web: si utiliza formularios basados en la web para recopilar la información de contacto de los clientes, valida los números de teléfono que proporcionan los clientes antes de enviar el formulario. Utilice el backend de su sitio web para validar el número mediante la API de Amazon Pinpoint. La respuesta de la API indica si el número no es válido, por ejemplo, si el número de teléfono está formateado incorrectamente. Si determina que el número de teléfono que el cliente ha proporcionado no es válido, el formulario web puede pedir al cliente que proporcione otro número.

- Limpieza de la base de datos de contactos existentes: si tiene una base de datos de números de teléfono de clientes, puede validar cada número de teléfono y, a continuación, actualizar la base de datos en función de los resultados. Por ejemplo, si encuentra puntos de enlace con números de teléfono que no pueden recibir mensajes SMS, puede cambiar la propiedad `ChannelType` del punto de enlace `SMS` a `VOICE`. Puede validar primero el número de teléfono y, a continuación, actualizar la propiedad `ChannelType` para los puntos de conexión nuevos o existentes siguiendo las instrucciones de Agregar puntos de conexión a Amazon Pinpoint para un único punto de conexión o Agregar un lote de puntos de conexión a Amazon Pinpoint para varios puntos de conexión.
- Elección del canal correcto antes de enviar un mensaje: si va a enviar un mensaje SMS pero determina que el número de destino no es válido, puede enviar un mensaje al destinatario a través de un canal diferente. Por ejemplo, si el punto de enlace no puede recibir mensajes SMS, puede enviar un mensaje de voz en su lugar.

Valide un número de teléfono mediante el AWS CLI

En el ejemplo siguiente, se muestra cómo validar un número de teléfono mediante la AWS CLI. Para obtener más información, consulte phone-number-validate en la Referencia de comandos de la AWS CLI . Para ver ejemplos de respuestas de validación, consulte Respuesta de validación de números de teléfono. Para obtener más información sobre la configuración del AWS CLI, consulte Configurar el AWS CLI en la Guía del AWS Command Line Interface usuario.

Para utilizar el servicio de validación de números de teléfono mediante el AWS CLI

- En la línea de comandos, introduzca el comando siguiente:

```
aws pinpoint phone-number-validate --number-validate-request
 PhoneNumber=+442079460881,IsoCountryCode=GB
```

En el comando anterior, *+442079460881* sustitúyalo por el número de teléfono que deseas validar y *GB* por el código ISO de país o región de dos dígitos.

> **Note**
>
> Cuando proporciona un número de teléfono en el servicio de validación de números de teléfono, debe incluir siempre el código de país. Si no incluye el código de país, el

servicio podría devolver información de un número de teléfono de un país diferente. Puede incluir guiones en el número de teléfono, por ejemplo*+44-207-946-0881*.

Respuesta de validación de números de teléfono

La información que el servicio de validación de números de teléfono proporciona varía ligeramente en función de los datos que estén disponibles para el número de teléfono proporcionado. En esta sección se incluyen ejemplos de las respuestas que devuelve el servicio de validación de números de teléfono.

> **Note**
>
> Los datos que proporciona el servicio de validación de números de teléfono depende de la información proporcionada por los proveedores del servicio de telecomunicaciones y otras entidades de todo el mundo. Los proveedores de algunos países podrían actualizar esta información con menos frecuencia que los de otros países. Por ejemplo, si emite una solicitud para validar un número de teléfono móvil y el número que proporciona se ha transferido de un operador móvil a otro, la respuesta del servicio de validación de números de teléfono podría incluir el nombre del operador original, en lugar del actual.

Números de teléfono móvil válidos

Cuando envía una solicitud al servicio de validación de números de teléfono y el número de teléfono es un número de teléfono móvil válido, el servicio devuelve información con un aspecto similar al del siguiente ejemplo:

```
{
    "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"
    }
}
```

Números de teléfono fijo válidos

Si su solicitud contiene un número de teléfono fijo válido, el servicio de validación de números de teléfono devuelve información similar a la del siguiente ejemplo:

```
{
    "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"
}
```

Números de teléfono VoIP válidos

Si su solicitud contiene un número de teléfono VoIP (protocolo de voz a través de Internet) válido, el servicio de validación de números de teléfono devuelve información con un aspecto similar al del siguiente ejemplo:

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+441514960001",
        "CleansedPhoneNumberNational": "1514960001",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+441514960001",
```

```
        "PhoneType": "VOIP",
        "PhoneTypeCode": 2
    }
}
```

Números de teléfono no válidos

Si su solicitud contiene un número de teléfono no válido, el servicio de validación de números de teléfono devuelve información similar a la del siguiente ejemplo:

```
{
    "NumberValidateResponse": {
        "CleansedPhoneNumberE164": "+44163296076",
        "CleansedPhoneNumberNational": "163296076",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+440163296076",
        "PhoneType": "INVALID",
        "PhoneTypeCode": 3
    }
}
```

Tenga en cuenta que la propiedad `PhoneType` de esta respuesta indica que este número de teléfono es `INVALID` y que no incluye información sobre el operador o la ubicación asociada con el número de teléfono. Debe evitar el envío de mensajes SMS o de voz a números de teléfono en los que `PhoneType` sea `INVALID`, ya que es muy poco probable que estos números pertenezcan a destinatarios reales.

Otros números de teléfono

A veces, la respuesta del servicio de validación de números de teléfono incluye el valor `PhoneType` `OTHER`. El servicio puede devolver este tipo de respuesta en las siguientes situaciones:

- El número de teléfono es un número de teléfono gratuito.
- El número de teléfono está reservado para su uso en series de televisión y películas, como los números de teléfono de los Estados Unidos que comienzan por 555.
- El número de teléfono incluye un código de área que actualmente no se encuentra en uso, como el código de área 999 en Norteamérica.
- El número de teléfono está reservado para cualquier otro fin.

En el siguiente ejemplo se muestra la respuesta que el servicio de validación de números de teléfono proporciona cuando su solicitud incluye un número de teléfono de Norteamérica ficticio:

```
{
    "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"
    }
}
```

Números de teléfono de prepago

Si la solicitud contiene un número de teléfono prepagado válido, el servicio de validación de números de teléfono devuelve información similar a la del siguiente ejemplo:

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "PREPAID",
        "PhoneTypeCode": 5
    }
}
```

Para obtener más información acerca de la información incluida en estas respuestas, consulte Validación de números de teléfono en la Referencia de la API de Amazon Pinpoint.

Crear un canal personalizado en Amazon Pinpoint mediante un webhook o una función de Lambda

Amazon Pinpoint incluye soporte integrado para enviar mensajes a través de la notificación de inserción, correo electrónico, SMS y canales de voz. También puede configurar Amazon Pinpoint para enviar mensajes a través de otros canales creando canales personalizados. Los canales personalizados de Amazon Pinpoint le permiten enviar mensajes a través de cualquier servicio que tenga una API, incluidos servicios de terceros. Puede interactuar con ellos APIs mediante un webhook o llamando a una AWS Lambda función.

Los segmentos a los que envía campañas de canal personalizadas pueden contener puntos de enlace de todos los tipos (es decir, puntos de enlace donde el valor del atributo `ChannelType` es EMAIL, VOICE, SMS, CUSTOM o uno de los distintos tipos de puntos de enlace de notificación push).

Usar un webhook

Si utilizas un webhook para enviar mensajes de canales personalizados, la URL del webhook debe empezar por «play with»https://". The webhook URL can only contain alphanumeric characters, plus the following symbols: hyphen (-), period (.), underscore (_), tilde (~), question mark (?), slash or solidus (/), pound or hash sign (#), and semicolon (:). The URL has to com. RFC3986

Cuando crea una campaña que especifica una dirección URL de webhook, Amazon Pinpoint emite un HTTP `HEAD` a esa URL. La respuesta a la solicitud `HEAD` debe contener un encabezado llamado `X-Amz-Pinpoint-AccountId`. El valor de este encabezado debe ser igual al ID de tu AWS cuenta.

Utilizar una función de Lambda

Si opta por enviar mensajes de canal personalizados mediante la creación de una función de Lambda, es mejor que se familiarice primero con los datos que emite Amazon Pinpoint. Cuando una campaña de Amazon Pinpoint envía mensajes a través de un canal personalizado, envía una carga a la función de Lambda de destino similar al siguiente ejemplo:

```
{
  "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"
    }
  }
}
```

Los datos de eventos proporciona los siguientes atributos:

- `ApplicationId`: el ID del proyecto de Amazon Pinpoint al que pertenece la campaña.
- `CampaignId`: el ID de la campaña de Amazon Pinpoint que invocó la función de Lambda.
- `TreatmentId`: el ID de la variante de la campaña. Si ha creado una campaña estándar, este valor siempre es 0. Si has creado una campaña de A/B prueba, este valor es un número entero entre 0 y 4.
- `ActivityId`: el ID de la actividad que está realizando la campaña.

- `ScheduledTime`: la hora en que Amazon Pinpoint ejecutó la campaña, que se muestra en formato ISO 8601.
- `Endpoints`: una lista de los puntos de conexión a los que se dirigió la campaña. Cada carga útil puede contener hasta 50 puntos de enlace. Si el segmento al que se envió la campaña contiene más de 50 puntos de conexión, Amazon Pinpoint invoca la función reiteradamente, de 50 en 50 puntos de conexión, hasta que se hayan procesado todos ellos.

Puede utilizar estos datos de ejemplo al crear y probar la función de Lambda de canal personalizado.

Puede asignar una función de Lambda o un webhook a una campaña de Amazon Pinpoint individual.

Para asignar una función de Lambda o webhook a una campaña individual, use la API de Amazon Pinpoint a fin de crear o actualizar un objeto de campaña.

El objeto `MessageConfiguration` de la campaña también debe contener un objeto `CustomMessage`. Este objeto tiene un miembro: `Data`. El valor de `Data` es una cadena JSON que contiene la carga útil del mensaje que desea enviar al canal personalizado.

La campaña tiene que contener un objeto `CustomDeliveryConfiguration`. En el objeto `CustomDeliveryConfiguration`, especifique lo siguiente:

- `EndpointTypes`: una matriz que contiene todos los tipos de puntos de conexión a los que se debe enviar la campaña de canales personalizados. Puede contener alguno o todos los tipos de canales siguientes:
  - `ADM`
  - `APNS`
  - `APNS_SANDBOX`
  - `APNS_VOIP`
  - `APNS_VOIP_SANDBOX`
  - `BAIDU`
  - `CUSTOM`
  - `EMAIL`
  - `GCM`
  - `SMS`

  - `VOICE`
- `DeliveryUri`: el destino al que se envían los puntos de conexión. Puede especificar solo una de las siguientes opciones:
  - La dirección URL del webhook al que desea enviar datos de punto de enlace cuando se ejecute la campaña.
  - El nombre de recurso de Amazon (ARN) de una función de Lambda que desea ejecutar cuando se ejecute la campaña.

> **Note**
>
> El objeto `Campaign` también puede contener un objeto `Hook`. Este objeto solo se utiliza para crear segmentos personalizados por una función de Lambda cuando se ejecuta una campaña. Para obtener más información, consulte Personalice los segmentos de Amazon Pinpoint mediante una función AWS Lambda.

Crear y configurar una función de Lambda para una campaña de Amazon Pinpoint

En esta sección se proporciona información general de los pasos para crear una función de Lambda que envía mensajes a través de un canal personalizado. Primero, cree la función. A continuación, agregue una política de ejecución a la función. Esta política permite a Amazon Pinpoint ejecutarla cuando se ejecuta una campaña.

Para obtener una introducción a la creación de funciones de Lambda, consulte Creación de funciones de Lambda en la Guía para desarrolladores de AWS Lambda .

Ejemplo de función de Lambda

El siguiente ejemplo de código procesa la carga útil y registra el número de puntos finales de cada tipo de punto final. 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)
```

Cuando una campaña de Amazon Pinpoint ejecuta esta función de Lambda, Amazon Pinpoint envía a la función una lista de miembros del segmento. La función cuenta el número de puntos de enlace de cada `ChannelType`. Luego envía esos datos a Amazon CloudWatch. Puedes ver estas métricas en la sección Métricas de la CloudWatch consola. Las métricas están disponibles en el espacio de PinpointCustomChannelExecutionnombres.

Puede modificar este ejemplo de código para que también se conecte a la API de un servicio externo para enviar mensajes a través de ese servicio.

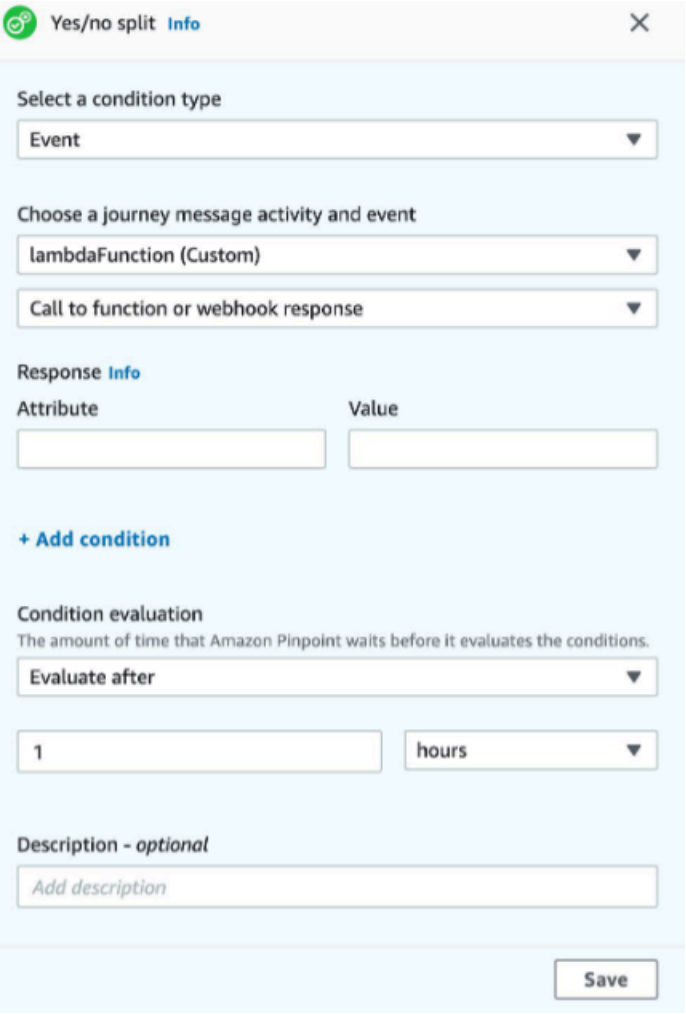
Formato de respuesta de función de Lambda para Amazon Pinpoint

Si desea utilizar el recorrido de forma multivariante o yes/no dividida para determinar la ruta del punto final después de una actividad de canal personalizada, debe estructurar la respuesta de la función Lambda en un formato que Amazon Pinpoint pueda entender y, a continuación, enviar los puntos de enlace por la ruta correcta.

La estructura de la respuesta debe estar en el siguiente formato:

```
{
    <Endpoint ID 1>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
    <Endpoint ID 2>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
...
}
```

Esto le permitirá seleccionar una clave y el valor que desee para determinar la ruta de los puntos de conexión.



Conceder a Amazon Pinpoint permiso para invocar la función de Lambda

Puede usar AWS Command Line Interface (AWS CLI) para añadir permisos a la política de funciones de Lambda asignada a su función de Lambda. Para permitir a Amazon Pinpoint invocar una función, utilice el comando add-permission de Lambda, como se muestra en el ejemplo siguiente:

```
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
```

En el comando anterior, haga lo siguiente.

- *myFunction*Sustitúyalo por el nombre de la función Lambda.
- *us-east-1*Sustitúyalo por la AWS región en la que utiliza Amazon Pinpoint.
- Reemplace *111122223333* por su ID de cuenta de AWS .

Cuando ejecuta el comando `add-permission`, Lambda devuelve la siguiente salida:

```
{
  "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\"}}}
}
```

El valor de `Statement` es una versión de cadena JSON de la instrucción que se agrega a la política de la función de Lambda.

Restricción adicional de la política de ejecución

Puede modificar la política de ejecución restringiéndola a un proyecto de Amazon Pinpoint específico. Para ello, sustituya * en el ejemplo anterior por el ID único del proyecto. Puede restringir aún más la política limitándola a una campaña específica. Por ejemplo, para restringir la política de modo que solo permita una campaña con el ID de campaña `95fee4cd1d7f5cd67987c1436example` en un proyecto con el ID de proyecto `dbaf6ec2226f0a9a8615e3ea5example`, utilice el valor siguiente para el atributo `source-arn`:

```
arn:aws:mobiletargeting:us-east-1:111122223333:apps/dbaf6ec2226f0a9a8615e3ea5example/
campaigns/95fee4cd1d7f5cd67987c1436example
```

> **Note**
>
> Si restringe la ejecución de la función de Lambda a una campaña específica, primero tiene que crear la función con una política menos restrictiva. A continuación, debe crear la campaña en Amazon Pinpoint y elegir la función. Por último, debe actualizar la política de ejecución para hacer referencia a la campaña especificada.

Transmitir datos de eventos de aplicación a través de Kinesis y Firehose con Amazon Pinpoint

En Amazon Pinpoint, un evento es una acción que se produce cuando un usuario interactúa con una de las aplicaciones, cuando envía un mensaje desde una campaña o recorrido o cuando envía un mensaje de correo electrónico o SMS transaccional. Por ejemplo, si envía un mensaje de correo electrónico, se producen varios eventos:

- Cuando envía el mensaje, se produce un evento de envío.
- Cuando el mensaje llega a la bandeja de entrada del destinatario, se produce un evento de entrega.
- Cuando el destinatario abre el mensaje, se produce un evento de apertura.

Puede configurar Amazon Pinpoint para enviar información sobre eventos a Amazon Kinesis. La plataforma Kinesis ofrece servicios que puede utilizar para recopilar, procesar y analizar datos de los servicios de AWS en tiempo real. Amazon Pinpoint puede enviar datos de eventos a Firehose, que transmite estos datos a almacenes de datos de AWS como Amazon S3 o Amazon Redshift. Amazon Pinpoint también puede transmitir datos a Kinesis Data Streams, que ingiere y almacena varios flujos de datos para que los procesen las aplicaciones de análisis.

La secuencia de eventos de Amazon Pinpoint incluye información acerca de las interacciones de los usuarios con las aplicaciones (apps) que conecte a Amazon Pinpoint. También incluye información sobre todos los mensajes que envíe desde campañas, a través de cualquier canal y de recorridos. Esto también puede incluir cualquier evento personalizado que haya definido. Por último, incluye información sobre los correos electrónicos y los mensajes SMS de transacciones que envíe.

> **Note**
>
> Amazon Pinpoint no transmite información sobre mensajes de voz ni notificaciones de inserción de transacciones.

En este capítulo se proporciona información acerca de la configuración de Amazon Pinpoint para transmitir datos de eventos a Kinesis. También contiene ejemplos de los datos de eventos transmitidos por Amazon Pinpoint.

Temas

- Configurar Amazon Pinpoint para transmitir datos de eventos de aplicación a través de Amazon Kinesis o Amazon Data Firehose
- Flujo de datos de eventos de aplicación desde Amazon Pinpoint
- Flujo de datos de eventos de campaña desde Amazon Pinpoint
- Datos de eventos de recorrido de Amazon Pinpoint
- Flujo de datos de eventos de correo electrónico desde Amazon Pinpoint
- Flujo de datos de eventos de SMS desde Amazon Pinpoint
- Eliminar un flujo de eventos de Amazon Pinpoint

Configurar Amazon Pinpoint para transmitir datos de eventos de aplicación a través de Amazon Kinesis o Amazon Data Firehose

Puede configurar Amazon Pinpoint para enviar datos de eventos a un flujo de Amazon Kinesis o un flujo de entrega de Amazon Data Firehose. Amazon Pinpoint puede enviar datos de eventos para campañas, recorridos y mensajes transaccionales de correo electrónico y SMS.

En esta sección se incluye información sobre la configuración del streaming de eventos mediante programación. También puede utilizar la consola de Amazon Pinpoint para configurar la transmisión de eventos. Para obtener información acerca de la configuración de la transmisión de eventos mediante la consola de Amazon Pinpoint, consulte Configuración de la transmisión de eventos en la Guía del usuario de Amazon Pinpoint.

Requisitos previos

Los ejemplos de esta sección requieren los datos siguientes:

- El ID de aplicación de una aplicación que está integrada con Amazon Pinpoint y eventos de notificación. Para obtener información acerca de cómo efectuar la integración, consulte Integrar Amazon Pinpoint con la aplicación.
- El nombre del recurso de Amazon (ARN) de una transmisión de Kinesis o Firehose de su cuenta. AWS Para obtener información acerca de la creación de estos recursos, consulte Creating and Managing Streams en la Guía para desarrolladores de Amazon Kinesis Data Streams o Creating an Amazon Data Firehose delivery stream en la Guía para desarrolladores de Amazon Data Firehose.

- El ARN de un rol AWS Identity and Access Management (IAM) que autoriza a Amazon Pinpoint a enviar datos a la transmisión. Para obtener información acerca de la creación de un rol, consulte Rol de IAM para la transmisión de eventos a Kinesis.

AWS CLI

En el siguiente AWS CLI ejemplo, se utiliza el comando. put-event-stream Este comando configura Amazon Pinpoint para enviar eventos a una secuencia de Kinesis:

```
aws pinpoint put-event-stream \
--application-id projectId \
--write-event-stream DestinationStreamArn=streamArn,RoleArn=roleArn
```

AWS SDK para Java

El siguiente ejemplo de Java configura Amazon Pinpoint para enviar eventos a una secuencia de 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);
}
```

En este ejemplo, se crea un `WriteEventStream` objeto que almacena la transmisión ARNs de Kinesis y la función de IAM. El objeto `WriteEventStream` se transfiere a un objeto `PutEventStreamRequest` para configurar Amazon Pinpoint de forma que transmita los eventos de una aplicación específica. El objeto `PutEventStreamRequest` se transfiere al método `putEventStream` del cliente de Amazon Pinpoint.

Puede asignar una secuencia de Kinesis a varias aplicaciones. Si lo hace, Amazon Pinpoint envía datos de los eventos codificados en base64 desde cada aplicación a la secuencia, lo que le

permite analizar los datos como una colección. El siguiente método de ejemplo acepta una lista de aplicaciones (aplicaciones) IDs y utiliza el método del ejemplo anterior para asignar una transmisión a cada aplicación: `createEventStream`

```
public List<PutEventStreamResult> createEventStreamFromAppList(
        AmazonPinpoint pinClient, List<String> appIDs,
        String streamArn, String roleArn) {

    return appIDs.stream()
            .map(appId -> createEventStream(pinClient, appId, streamArn,
                    roleArn))
            .collect(Collectors.toList());
}
```

Aunque es posible asignar una secuencia a varias aplicaciones, no es posible asignar varias secuencias a una aplicación.

Flujo de datos de eventos de aplicación desde Amazon Pinpoint

Después de integrar la aplicación (app) con Amazon Pinpoint y configurar la transmisión de eventos, Amazon Pinpoint recupera los datos de actividad del usuario, eventos personalizados y entregas de mensajes de la aplicación desde el destino que haya especificado durante la configuración para que pueda verlos. Para obtener información sobre cómo configurar transmisiones de eventos para poder ver los datos de los eventos, consulte Configurar Amazon Pinpoint para transmitir datos de eventos de aplicación a través de Amazon Kinesis o Amazon Data Firehose.

Ejemplo de evento de aplicación

El objeto JSON de un evento de aplicación contiene los datos que se muestran en el siguiente ejemplo.

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

Atributos de los eventos de aplicación

En esta sección se definen los atributos que se han incluido en el ejemplo anterior de la secuencia de eventos de la aplicación.

Atributo	Descripción
`event_type`	El tipo de evento. Los valores posibles son los siguientes:

Atributo	Descripción
	• _session.start: el punto de conexión inició una nueva sesión. • _session.stop: el punto de conexión terminó una sesión. • _userauth.sign_in: el punto de conexión se registró en la aplicación. • _userauth.sign_up: un nuevo punto de conexión completó el proceso de registro en la aplicación. • _userauth.auth_fail: el punto de conexión intentó iniciar sesión en la aplicación, pero no pudo completar el proceso. • _monetization.purchase: el punto de conexión realizó una compra en la aplicació n. • _session.pause: el punto de conexión detuvo una sesión. Las sesiones en pausa se pueden reanudar para poder seguir recopilan do métricas sin iniciar una sesión totalmente nueva. • _session.resume: el punto de conexión reanudó una sesión.
`event_timestamp`	La hora a la que se notificó el evento, que se muestra en formato de tiempo Unix en milisegundos.
`arrival_timestamp`	La hora a la que Amazon Pinpoint recibió el evento, que se muestra en formato de tiempo Unix en milisegundos.

Atributo	Descripción
`event_version`	La versión del esquema JSON del evento. **Tip** Compruebe esta versión en la aplicación de procesamiento de eventos para saber cuándo actualizar la aplicación en respuesta a una actualización del esquema.
`application`	Información sobre el proyecto de Amazon Pinpoint asociado al evento. Consulte la tabla de la sección Application (Aplicación) para obtener más información.
`client`	Información sobre el punto de enlace que notificó el evento. Consulte la tabla de la sección Client (Cliente) para obtener más información.
`device`	Información sobre el dispositivo que notificó el evento. Para obtener más información, consulte la tabla Dispositivos.
`session`	Información sobre la sesión que generó el evento. Consulte la tabla de la sección Session (Sesión) para obtener más información.
`attributes`	Atributos asociados al evento. Para los eventos notificados por las aplicaciones, este objeto incluye los atributos personalizados que usted defina.

Atributo	Descripción
`metrics`	Métricas relacionadas con el evento. Si lo desea, puede configurar las aplicaciones para que envíen métricas personalizadas a Amazon Pinpoint.

Aplicación

Incluye información sobre el proyecto de Amazon Pinpoint al que está asociado el evento.

Atributo	Descripción
`app_id`	El ID único del proyecto de Amazon Pinpoint que notificó el evento.
`cognito_identity_pool_id`	El ID del grupo de identidades de Amazon Cognito al que está asociado el punto de conexión.
`package_name`	El nombre del paquete de la aplicación, como `com.example.my_app` .
`sdk`	Información sobre el SDK que se utilizó para notificar el evento. Consulte la tabla de la sección SDK para obtener más información.
`title`	El nombre de la aplicación.
`version_name`	El nombre de la versión de la aplicación, como, por ejemplo, `V2.5`.
`version_code`	El número de versión de la aplicación, como, por ejemplo, `3`.

SDK

Incluye información sobre el SDK que se utilizó para notificar el evento.

Atributo	Descripción
`name`	El nombre del SDK que se utilizó para notificar el evento.
`version`	La versión del SDK.

Cliente

Incluye información sobre el punto de enlace que generó el evento.

Atributo	Descripción
`client_id`	ID del punto de enlace.
`cognito_id`	El token del ID de Amazon Cognito asociado al punto de conexión.

Dispositivo

Incluye información sobre el dispositivo del punto de enlace que generó el evento.

Atributo	Descripción
`locale`	Información sobre la configuración de idioma y región del dispositivo del punto de enlace. Consulte la tabla de la sección Locale (Local) para obtener más información.
`make`	El fabricante del dispositivo del punto de enlace.
`model`	El identificador del modelo del dispositivo del punto de enlace.
`platform`	Información sobre el sistema operativo en el dispositivo del punto de enlace. Consulte la

Atributo	Descripción
	tabla de la sección Platform (Plataforma) para obtener más información.

Locale

Incluye información sobre la configuración de idioma y región del dispositivo del punto de enlace.

Atributo	Descripción
`code`	El identificador de configuración regional asociado al dispositivo.
`country`	El país o la región que está asociado a la configuración regional del dispositivo.
`language`	El idioma asociado a la configuración regional del dispositivo.

Plataforma

Incluye información sobre el sistema operativo en el dispositivo del punto de enlace.

Atributo	Descripción
`name`	El nombre del sistema operativo del dispositivo.
`version`	La versión del sistema operativo del dispositivo.

Sesión

Incluye información sobre la sesión que generó el evento.

Atributo	Descripción
`session_id`	ID único que identifica la sesión.

Atributo	Descripción
`start_timestamp`	La fecha y la hora en que comenzó la sesión, que se muestra en formato de tiempo Unix en milisegundos.
`stop_timestamp`	La fecha y la hora en que finalizó la sesión, que se muestra en formato de tiempo Unix en milisegundos.

Flujo de datos de eventos de campaña desde Amazon Pinpoint

Si utiliza Amazon Pinpoint para enviar campañas a través de un canal, Amazon Pinpoint puede transmitir datos de eventos sobre esas campañas. Tras configurar la transmisión de eventos, Amazon Pinpoint recupera los datos de los eventos de la aplicación para los mensajes de correo electrónico o SMS que usted envíe desde una campaña desde el destino que haya especificado durante la configuración para que pueda verlos. Para obtener información detallada sobre los datos que Amazon Pinpoint transmite para mensajes de correo electrónico y SMS, consulte the section called “Flujo de datos de eventos de correo electrónico desde Amazon Pinpoint” y the section called “Flujo de datos de eventos de SMS desde Amazon Pinpoint”. Para obtener información sobre cómo configurar transmisiones de eventos, consulte Configurar Amazon Pinpoint para transmitir datos de eventos de aplicación a través de Amazon Kinesis o Amazon Data Firehose.

Ejemplo de evento de campaña

El objeto JSON para un evento de campaña contiene los datos que se muestran en el siguiente ejemplo.

```
{
  "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"
}
```

Atributos de los eventos de campaña

En esta sección se definen los atributos que se incluyen en la secuencia de eventos de la campaña.

Atributo	Description (Descripción)
`event_type`	El tipo de evento. Los valores posibles son los que se indican a continuación. • _campaign.send: Amazon Pinpoint ejecutó la campaña. • _campaign.opened_notification: para las campañas de notificaciones de inserción, este tipo de evento indica que el destinatario tocó la notificación para abrirla. • _campaign.received_foreground: para las campañas de notificaciones de inserción, este tipo de evento indica que el destinatario

Atributo	Description (Descripción)
	recibió el mensaje como una notificación en primer plano. • _campaign.received_background: para las campañas de notificaciones de inserción, este tipo de evento indica que el destinatario recibió el mensaje como una notificación en primer plano. **Note** _campaign.opened_notification, _campaign.received_foreground y _campaign.received_background se devuelven solo si usa AWS Amplify. Para obtener más información sobre la integración de su aplicación con AWS Amplify. Consulte Conecte su aplicación de interfaz a Amazon AWS Pinpoint mediante Amplify.
`event_timestamp`	La hora a la que se notificó el evento, que se muestra en formato de tiempo Unix en milisegundos.
`arrival_timestamp`	La hora a la que Amazon Pinpoint recibió el evento, que se muestra en formato de tiempo Unix en milisegundos.

Atributo	Description (Descripción)
`event_version`	La versión del esquema JSON del evento. **Tip** Compruebe esta versión en la aplicación de procesamiento de eventos para saber cuándo actualizar la aplicación en respuesta a una actualización del esquema.
`application`	Información sobre el proyecto de Amazon Pinpoint asociado al evento. Consulte la tabla de la sección Application (Aplicación) para obtener más información.
`client`	Información sobre el punto de enlace al que está asociado el evento. Consulte la tabla de la sección Client (Cliente) para obtener más información.
`device`	Información sobre el dispositivo que notificó el evento. Para los mensajes de campaña y de transacciones, este objeto está vacío.
`session`	Información sobre la sesión que generó el evento. Para las campañas, este objeto está vacío.

Atributo	Description (Descripción)
`attributes`	Atributos asociados al evento. Para los eventos notificados por una de sus aplicaciones, este objeto puede incluir atributos personalizados definidos por la aplicación. Para los eventos que se crean al enviar una campaña, este objeto contiene atributos asociados a la campaña. Para los eventos que se generan al enviar mensajes de transacciones, este objeto contiene información relacionada con el propio mensaje. Consulte la tabla de la sección Attributes (Atributos) para obtener más información.
`client_context`	Contiene un objeto `custom`, que contiene una propiedad `endpoint`. La propiedad `endpoint` contiene el contenido del registro del punto de enlace al que se envió la campaña.
`awsAccountId`	El ID de la AWS cuenta que se utilizó para enviar el mensaje.

Aplicación

Incluye información sobre el proyecto de Amazon Pinpoint al que está asociado el evento.

Atributo	Description (Descripción)
`app_id`	El ID único del proyecto de Amazon Pinpoint que notificó el evento.
`sdk`	El SDK que se utilizó para notificar el evento.

Atributos

Incluye información sobre la campaña que produjo el evento.

Atributo	Description (Descripción)
`treatment_id`	Si el mensaje se envió mediante una campaña de A/B prueba, este valor representa el número de tratamiento del mensaje. Para las campañas estándar, este valor es `0`.
`campaign_activity_id`	El ID único que Amazon Pinpoint genera cuando se produce el evento.
`delivery_type`	El método de entrega de la campaña. No confunda este atributo con el campo `ChannelType` especificado en la propiedad `endpoint` de `client_context` . El campo `ChannelType` se basa normalmente en el punto de conexión al que se envía el mensaje. En el caso de los canales que admiten solo un tipo de punto de conexión, los campos `delivery_type` y `ChannelType` tienen el mismo valor. Por ejemplo, para el canal de correo electrónico, los campos `delivery_type` y `ChannelType` tienen el mismo valor de CORREO ELECTRÓNICO. Sin embargo, esta condición no siempre se cumple en los canales que admiten distintos tipos de puntos de conexión, como los canales personalizados. Puede usar un canal personalizado para distintos puntos de conexión, como EMAIL, SMS, CUSTOM, etc. En este caso, `delivery_type` identifica un evento de entrega personalizado, CUSTOM, y `ChannelType` especifica el tipo de punto de

Atributo	Description (Descripción)
	conexión al que se envió la campaña, como EMAIL, SMS, CUSTOM, etc. Para obtener más información sobre la creación de canales personalizados, consulte Crear un canal personalizado. Los valores posibles son los que se indican a continuación. • CORREO ELECTRÓNICO • SMS • ADM • APNS • APNS_SANDBOX • APNS_VOIP • APNS_VOIP_SANDBOX • VOZ • GCM • BAIDU • INSERCIÓN • PERSONALIZAR
`campaign_id`	El ID único de la campaña desde la que se envió el mensaje.

Atributo	Description (Descripción)
`campaign_send_status`	Indica el estado de la campaña para el punto de enlace de destino. Los valores posibles son: • SUCCESS: la campaña se envió correctam ente al punto de conexión. • FAILURE: la campaña no se envió al punto de conexión. • DAILY_CAP: la campaña no se envió al punto de conexión porque ya se ha enviado el número máximo de mensajes diarios al punto de conexión. • EXPIRED: la campaña no se envió al punto de conexión porque su envío superaría la duración máxima o la velocidad de envío configuradas para la campaña. • QUIET_TIME: la campaña no se envió al punto de conexión debido a las restricciones de tiempo de inactividad. • HOLDOUT: la campaña no se envió al punto de conexión porque el punto de conexión era miembro del grupo de espera. • DUPLICATE_ADDRESS: hay direccion es de punto de conexión duplicadas en el segmento. La campaña se envió una vez a la dirección del punto de conexión. • QUIET_TIME: la campaña no se envió al punto de conexión debido a las restricciones de tiempo de inactividad. • CAMPAIGN_CAP: la campaña no se envió al punto de conexión porque ya se ha enviado el número máximo de mensajes al punto de conexión de esta campaña.

Atributo	Description (Descripción)
	• FAILURE_PERMANENT: se ha producido un error permanente al enviar al punto de conexión. • TRANSIENT_FAILURE: se ha producido un error transitorio al enviar al punto de conexión. • THROTTLED: el envío estaba limitado. • UNKNOWN: error desconocido. • HOOK_FAILURE: se ha producido un error en el enlace de la campaña. • CUSTOM_DELIVERY_FAILURE: se ha producido un error al realizar entrega personalizada. • RECOMMENDATION_FAILURE: no se ha podido realizar la recomendación. • UNSUPPORTED_CHANNEL: el canal no es compatible.

Cliente

Incluye información sobre el punto de enlace al que se dirigía la campaña.

Atributo	Description (Descripción)
`client_id`	El ID del punto de enlace al que se envió la campaña.

Datos de eventos de recorrido de Amazon Pinpoint

Cuando publica un recorrido, Amazon Pinpoint puede transmitir datos de eventos de los mensajes de correo electrónico, SMS, push y personalizados que envíe desde el recorrido. Tras configurar la transmisión de eventos, Amazon Pinpoint recupera los datos desde el destino que haya especificado

durante la configuración para que pueda verlos. Para obtener información detallada sobre los datos que Amazon Pinpoint transmite para mensajes de correo electrónico y SMS, consulte the section called “Flujo de datos de eventos de correo electrónico desde Amazon Pinpoint” y the section called “Flujo de datos de eventos de SMS desde Amazon Pinpoint”. Para obtener información sobre cómo configurar transmisiones de eventos, consulte Configurar Amazon Pinpoint para transmitir datos de eventos de aplicación a través de Amazon Kinesis o Amazon Data Firehose.

Ejemplo de evento de recorrido

El objeto JSON de un evento de recorrido contiene los datos que se muestran en el siguiente ejemplo.

```
{
   "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"
}
```

Atributos de eventos de recorrido

En esta sección se definen los atributos que se incluyen en los datos de la secuencia de eventos que Amazon Pinpoint genera para un recorrido.

Atributo	Descripción
`event_type`	El tipo de evento. Para los eventos de recorrido, el valor de este atributo es siempre `_journey.send` , lo que indica que Amazon Pinpoint ejecutó el recorrido.
`event_timestamp`	La hora a la que se notificó el evento, que se muestra en formato de tiempo Unix en milisegundos.
`arrival_timestamp`	La hora a la que Amazon Pinpoint recibió el evento, que se muestra en formato de tiempo Unix en milisegundos.
`event_version`	La versión del esquema JSON del evento. **Tip** Compruebe esta versión en la aplicación de procesamiento de eventos para saber cuándo actualizar la aplicación

Atributo	Descripción
	en respuesta a una actualización del esquema.
`application`	Información sobre el proyecto de Amazon Pinpoint asociado al evento. Consulte la tabla de la sección Application (Aplicación) para obtener más información.
`client`	Información sobre el punto de enlace que está asociado al evento. Consulte la tabla de la sección Client (Cliente) para obtener más información.
`device`	Información sobre el dispositivo que notificó el evento. Para recorridos, este objeto está vacío.
`session`	Información sobre la sesión que generó el evento. Para recorridos, este objeto está vacío.
`attributes`	Los atributos que están asociados con el recorrido y la actividad de recorrido que generó el evento. Consulte la tabla de la sección Attributes (Atributos) para obtener más información.
`client_context`	Contiene un objeto `custom`, que contiene una propiedad `endpoint`. La propiedad `endpoint` contiene el contenido del registro de punto de enlace del punto de enlace asociado al evento.
`awsAccountId`	El ID de la cuenta de AWS que se utilizó para ejecutar el recorrido.

Aplicación

Incluye información sobre el proyecto de Amazon Pinpoint al que está asociado el evento.

Atributo	Descripción
`app_id`	El ID único del proyecto de Amazon Pinpoint que notificó el evento.
`sdk`	El SDK que se utilizó para notificar el evento.

Cliente

Incluye información sobre el punto de enlace al que está asociado con el evento.

Atributo	Descripción
`client_id`	ID del punto de enlace.

Atributos

Incluye información sobre el recorrido que generó el evento.

Atributo	Descripción
`journey_run_id`	El ID único del recorrido ejecutado que generó el evento. Amazon Pinpoint genera y asigna este ID automáticamente a cada nueva ejecución de un recorrido.
`journey_send_status`	Indica el estado de entrega del mensaje que está asociado al evento. Los valores posibles son: • SUCCESS: el mensaje se envió correctam ente al punto de conexión. • FAILURE: el mensaje no se envió al punto de conexión porque se produjo un error.

Atributo	Descripción
	• CUSTOM_DELIVERY_FAILURE: se ha producido un error al realizar entrega personalizada. • FAILURE_PERMANENT: se ha producido un error permanente al enviar al punto de conexión. **Tip** Puede filtrar los eventos con el estado FAILURE_PERMANENT y `journey_send_status_code` establecido en 403 para determinar si existe una infracción de la política de acceso o del rol. En el caso de las campañas salientes con voz, estas excepciones son habituales en los casos en que el rol de ejecución de la campaña de Connect que vincula los recorridos de Amazon Pinpoint a las campañas de Amazon Connect se elimina sin querer para las ejecuciones de recorridos en proceso. • THROTTLED: el envío estaba limitado. • UNSUPPORTED_CHANNEL: el canal no es compatible. • DAILY_CAP: el mensaje no se envió al punto de conexión porque el envío del mensaje superaría el número máximo de mensajes que el recorrido o el proyecto pueden enviar a un único punto de conexión en un periodo de 24 horas.

Atributo	Descripción
	• QUIET_TIME: el mensaje no se envió debido a las restricciones de tiempo de silencio durante el recorrido o el proyecto. • QUIET_TIME_MISSING_TIMEZONE: el mensaje no se envió porque la estimación de la zona horaria no pudo estimar un zona horaria para el punto de conexión y el modo silencioso está habilitado.
`journey_id`	El identificador único del recorrido que generó el evento.
`journey_activity_id`	El identificador único de la actividad de recorrido que generó el evento.
`journey_activity_type`	El tipo de actividad de recorrido del evento. Esto puede ser EMAIL, SMS, PUSH, CONTACT_CENTER o CUSTOM. **Note** VOICE no es un tipo de actividad de recorrido compatible. El campo `journey_activity_type` no está presente cuando `journey_send_status` se establece en QUIET_TIME_WAIT_FINISHED.
`journey_send_status_message`	La descripción del estado del evento de envío.
`journey_send_status_code`	El código de estado HTTP de la solicitud.

Flujo de datos de eventos de correo electrónico desde Amazon Pinpoint

Si utiliza Amazon Pinpoint para enviar correos electrónicos, Amazon Pinpoint puede transmitir datos de eventos sobre esos correos electrónicos. Tras configurar la transmisión de eventos, Amazon Pinpoint recupera los datos de los eventos desde el destino que haya especificado durante la configuración para que pueda verlos. Para obtener información sobre cómo configurar transmisiones de eventos, consulte Configurar Amazon Pinpoint para transmitir datos de eventos de aplicación a través de Amazon Kinesis o Amazon Data Firehose. Amazon Pinpoint transmite datos sobre los siguientes tipos de eventos para los mensajes de correo electrónico:

- Envíos
- Entregas
- Rebotes
- Reclamos
- Aperturas
- Clics
- Rechazos
- Cancelaciones de suscripciones
- Fallos de representación

Estos tipos de eventos se explican en detalle en Atributos de los eventos de correo electrónico.

En función de la API y de la configuración que utilice para enviar mensajes de correo electrónico, es posible que vea tipos de eventos adicionales o datos diferentes. Por ejemplo, si envía mensajes mediante conjuntos de configuración que publican datos de eventos en Amazon Kinesis, como los proporcionados por Amazon Simple Email Service (Amazon SES), los datos también pueden incluir eventos para errores de representación de plantillas. Para obtener información acerca de esos datos, consulte Monitoreo mediante la publicación de eventos de Amazon SES en la Guía para desarrolladores de Amazon Simple Email Service. Antes de poder ver sus eventos, debe configurar la transmisión de eventos; consulte Configurar Amazon Pinpoint para transmitir datos de eventos de aplicación a través de Amazon Kinesis o Amazon Data Firehose. Al configurar la transmisión de eventos, especifique un destino en el que se guardarán los datos del evento; después, podrá usar el destino para recuperar los datos del evento para su visualización.

Ejemplos de eventos de correo electrónico

Envío de correo electrónico

El objeto JSON referente a un evento de envío de correo electrónico contiene los datos que se muestran en el siguiente ejemplo.


```
{
  "event_type": "_email.send",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622025,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "9a311b17-6f8e-4093-be61-4d0bbexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "received"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
```

```
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "send": {}
      }
    }
  }
}
```

Entrega de correo electrónico

El objeto JSON referente a un evento de entrega de correo electrónico contiene los datos que se muestran en el siguiente ejemplo.

```
{
  "event_type": "_email.delivered",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622690,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "e9a3000d-daa2-40dc-ac47-1cd34example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
```

```
    "feedback": "delivered"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "delivery": {
          "smtp_response": "250 ok:  Message 82080542 accepted",
          "reporting_mta": "a8-53.smtp-out.amazonses.com",
          "recipients": ["recipient@example.com"],
          "processing_time_millis": 1310
        }
      }
    }
  }
```

```
}
```

Clic en correo electrónico

El objeto JSON referente a un evento de clic en correo electrónico contiene los datos que se muestran en el siguiente ejemplo.

```
{
  "event_type": "_email.click",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618713751,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "49c1413e-a69c-46dc-b1c4-6470eexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "https://aws.amazon.com/pinpoint/"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
```

```
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "click": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15",
          "link": "https://aws.amazon.com/pinpoint/"
        }
      }
    }
  }
}
```

Apertura de correo electrónico

El objeto JSON referente a un evento de apertura de correo electrónico contiene los datos que se muestran en el siguiente ejemplo.

```
{
  "event_type": "_email.open",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618712316,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
```

```
  "client": {
    "client_id": "8dc1f651-b3ec-46fc-9b67-2a050example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "opened"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
```

```
        },
        "open": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko)"
        }
      }
    }
  }
}
```

Atributos de los eventos de correo electrónico

En esta sección se definen los atributos que se han incluido en el ejemplo anterior de los datos de la secuencia de eventos que Amazon Pinpoint genera cuando se envían mensajes de correo electrónico.

Atributo	Descripción
`event_type`	El tipo de evento. Los valores posibles son los siguientes: • _email.send: Amazon Pinpoint aceptó el mensaje e intentó entregarlo al destinatario. • _email.delivered: el mensaje se envió al destinatario. • _email.rejected: Amazon Pinpoint determinó que el mensaje contenía malware y no intentó enviarlo. • _email.hardbounce: un problema permanent e impedía que Amazon Pinpoint entregara el mensaje. Amazon Pinpoint no volverá a realizar un intento de entrega del mensaje. • _email.softbounce: un problema temporal impedía que Amazon Pinpoint entregara el mensaje. Amazon Pinpoint volverá a realizar un intento de entrega del mensaje durante un tiempo determinado. Si el mensaje sigue

Atributo	Descripción
	sin poder entregarse, no se realizarán más reintentos. El estado final del correo electróni co será entonces SOFTBOUNCE. • _email.complaint: el destinatario recibió el mensaje y, a continuación, informó del mensaje al proveedor de correo electróni co como spam (por ejemplo, mediante la característica “Denunciar spam” del cliente de correo electrónico). • _email.open: el destinatario recibió el mensaje y lo abrió. • _email.click: el destinatario recibió el mensaje e hizo clic en su enlace. • _email.unsubscribe: el destinatario recibió el mensaje e hizo clic en su enlace para cancelar la suscripción. • _email.rendering_failure: el correo electrónico no se envió debido a un error de presentac ión. Esto se puede producir cuando faltan datos en la plantilla o cuando los parámetros y los datos de la plantilla no coinciden.
`event_timestamp`	Hora en que se envió el mensaje, que se muestra en formato de tiempo Unix en milisegundos. Este valor suele ser el mismo para todos los eventos que se generan para un mensaje.
`arrival_timestamp`	La hora a la que Amazon Pinpoint recibió el evento, que se muestra en formato de tiempo Unix en milisegundos.

Atributo	Descripción
`event_version`	La versión del esquema JSON del evento. **Tip** Compruebe esta versión en la aplicación de procesamiento de eventos para saber cuándo actualizar la aplicación en respuesta a una actualización del esquema.
`application`	Información sobre el proyecto de Amazon Pinpoint asociado al evento. Consulte la tabla de la sección Aplicación para obtener más información.
`client`	Información sobre el cliente de la aplicación instalado en el dispositivo que notificó el evento. Consulte la tabla de la sección Client (Cliente) para obtener más información.
`device`	Información sobre el dispositivo que notificó el evento. Para obtener más información, consulte la tabla Dispositivos. Para los eventos de correo electrónico, este objeto está vacío.
`session`	Para los eventos de correo electrónico, este objeto está vacío.

Atributo	Descripción
`attributes`	Atributos asociados al evento. Consulte la tabla de la sección Attributes (Atributos) para obtener más información. Para los eventos notificados por una de sus aplicaciones, este objeto puede incluir atributos personalizados definidos por la aplicación. Para los eventos que se crean al enviar un mensaje desde una campaña o recorrido, este objeto contiene atributos que están asociados a la campaña o recorrido. Para los eventos que se generan al enviar mensajes de transacciones, este objeto contiene información relacionada con el propio mensaje.
`client_context`	Para eventos de correo electrónico, este objeto contiene un objeto `custom`, que contiene un atributo `legacy_identifier` . El valor del atributo `legacy_identifier` es el ID del proyecto desde el que se envió el mensaje.
`facets`	Información adicional sobre el mensaje, como los encabezados de correo electrónico. Consulte la tabla de la sección Facetas para obtener más información.
`awsAccountId`	El ID de la cuenta de AWS de que se utilizó para enviar el mensaje.

Aplicación

Incluye información sobre el proyecto de Amazon Pinpoint al que está asociado el evento.

Atributo	Descripción
`app_id`	El ID único del proyecto de Amazon Pinpoint que notificó el evento.
`sdk`	El SDK que se utilizó para notificar el evento. Si envía un mensaje de correo electrónico de transacciones llamando directamente a la API de Amazon Pinpoint o con la consola de Amazon Pinpoint, este objeto estará vacío.

Atributos

Incluye información sobre la campaña o el recorrido que produjo el evento.

Campaña

Incluye información sobre la campaña que produjo el evento.

Atributo	Descripción
`feedback`	Para los eventos `_email.click` , el valor de este atributo es la URL del enlace en el que el destinatario hizo clic en el mensaje para generar el evento. Para otros eventos, el valor representa el tipo de evento, como `received`, `opened` o `clicked`.
`treatment_id`	Si el mensaje se envió mediante una campaña de prueba A/B, este valor representa el número de tratamiento del mensaje. Para campañas estándar y mensajes de correo electrónico de transacciones, este valor es `0`.
`campaign_activity_id`	El ID único que Amazon Pinpoint genera cuando se produce el evento.

Atributo	Descripción
`campaign_id`	El ID único de la campaña que envió el mensaje.

Recorrido

Incluye información sobre el recorrido que produjo el evento.

Atributo	Descripción
`journey_run_id`	El ID único del recorrido ejecutado que envió el mensaje. Amazon Pinpoint genera y asigna este ID automáticamente a cada nueva ejecución de un recorrido.
`feedback`	Para los eventos `_email.click` , el valor de este atributo es la URL del enlace en el que el destinatario hizo clic en el mensaje para generar el evento. Para otros eventos, el valor representa el tipo de evento, como `received`, `delivered` o `opened`.
`journey_id`	El identificador único del recorrido que envió el mensaje.
`journey_activity_id`	El identificador único de la actividad de recorrido que envió el mensaje.

Cliente

El identificador único del cliente al que se dirigió la campaña o el recorrido.

Atributo	Descripción
`client_id`	El ID del cliente. El valor es el ID de punto de conexión para las campañas y los recorridos, para el envío transaccional, es un UUID.

Facetas

Incluye información sobre el mensaje y el tipo de evento.

Atributo	Descripción
`email_channel`	Contiene un objeto `mail_event` , que contiene dos objetos: `mail` y un objeto que se corresponde con el tipo de evento.

Mail

Incluye información sobre el contenido del mensaje de correo electrónico, así como metadatos relacionados con el mensaje.

Atributo	Descripción
`message_id`	El ID único del mensaje. Amazon Pinpoint genera automáticamente este ID cuando acepta el mensaje.
`message_send_timestamp`	La fecha y la hora a la que se envió el mensaje, en el formato especificado en RFC 822.
`from_address`	La dirección de correo electrónico desde la que se envió el mensaje.
`destination`	Una matriz que contiene las direcciones de correo electrónico a las que se envió el mensaje.

Atributo	Descripción
`headers_truncated`	Un valor booleano que indica si los encabezados de correo electrónico se han truncado.
`headers`	Un objeto que contiene varios pares nombre-valor que se corresponden con los encabezados del mensaje. Este objeto normalmente contiene información sobre los siguientes encabezados: • `From`: la dirección de correo electrónico del remitente. • `To`: la dirección de correo electrónico del destinatario. • `Subject`: la línea de asunto del correo electrónico. **Tip** El encabezado del asunto no se incluye en los eventos de _email.send de la campaña. • `MIME-Version` : indica que el mensaje está en formato MIME. Si existe este encabezado, su valor siempre es `1.0`. • `Content-Type` : el tipo de medio MIME del contenido del mensaje.
`common_headers`	Contiene información sobre varios encabezados comunes para mensajes de correo electrónico. La información puede incluir la fecha en que se envió el mensaje y las líneas de De, Para y Asunto del mensaje.

Flujo de datos de eventos de SMS desde Amazon Pinpoint

Si el canal de SMS está habilitado para un proyecto, Amazon Pinpoint puede transmitir datos de eventos acerca de entregas de mensajes SMS para el proyecto. Tras configurar la transmisión de eventos, Amazon Pinpoint recupera los datos de los eventos desde el destino que haya especificado durante la configuración para que pueda verlos. Para obtener información sobre cómo configurar transmisiones de eventos, consulte Configurar Amazon Pinpoint para transmitir datos de eventos de aplicación a través de Amazon Kinesis o Amazon Data Firehose.

> **Note**
>
> Los eventos de SMS generados por los operadores pueden tardar hasta 72 horas en recibirse y no se deben utilizar para determinar si hay un retraso en la entrega de los mensajes salientes. Transcurridas 72 horas, si Amazon Pinpoint no ha recibido un evento final de un operador, el servicio devuelve automáticamente el record_status UNKNOWN, ya que Amazon Pinpoint no sabe qué ha ocurrido con ese mensaje.

Ejemplo de evento de SMS

El objeto JSON relacionado con un evento de SMS contiene los datos que se muestran en el siguiente ejemplo.

```
{
  "event_type": "_SMS.SUCCESS",
  "event_timestamp": 1553104954322,
  "arrival_timestamp": 1553104954064,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "123456789012"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
```

```
    "sender_request_id": "565d4425-4b3a-11e9-b0a5-example",
    "campaign_activity_id": "cbcfc3c5e3bd48a8ae2b9cb41example",
    "origination_phone_number": "+12065550142",
    "destination_phone_number": "+14255550199",
    "record_status": "DELIVERED",
    "iso_country_code": "US",
    "treatment_id": "0",
    "number_of_message_parts": "1",
    "message_id": "1111-2222-3333",
    "message_type": "Transactional",
    "campaign_id": "52dc44b35c4742c98c5935269example"
    "customer_context": "{\"userId\":\"user-id-4\"}"
  },
  "metrics": {
    "price_in_millicents_usd": 645.0
  },
  "awsAccountId": "123456789012"
}
```

Atributos de los eventos de SMS

En esta sección se definen los atributos que se han incluido en el ejemplo anterior de los datos de la secuencia de eventos que Amazon Pinpoint genera cuando se envían mensajes de SMS.

Evento

Atributo	Descripción
`event_type`	El tipo de evento. Los valores posibles son los siguientes: • _SMS.BUFFERED: el mensaje aún está en proceso de entrega al destinatario. • _SMS.SUCCESS: el operador aceptó correctamente el mensaje o lo entregó al destinatario. • _SMS.FAILURE: Amazon Pinpoint no pudo entregar el mensaje al destinatario. Para obtener más información sobre el error que impidió la entrega del mensaje, consulte `attributes.record_status` .

Atributo	Descripción
	• _SMS.OPTOUT: el cliente recibió el mensaje y respondió enviando la palabra clave de desactivación (normalmente “STOP”).
`event_timestamp`	La hora a la que se notificó el evento, que se muestra en formato de tiempo Unix en milisegundos.
`arrival_timestamp`	La hora a la que Amazon Pinpoint recibió el evento, que se muestra en formato de tiempo Unix en milisegundos.
`event_version`	La versión del esquema JSON del evento. **Tip** Compruebe esta versión en la aplicación de procesamiento de eventos para saber cuándo actualizar la aplicación en respuesta a una actualización del esquema.
`application`	Información sobre el proyecto de Amazon Pinpoint asociado al evento. Consulte la tabla de la sección Application (Aplicación) para obtener más información.
`client`	Información sobre el cliente de la aplicación instalado en el dispositivo que notificó el evento. Consulte la tabla de la sección Client (Cliente) para obtener más información.

Atributo	Descripción
`device`	Información sobre el dispositivo que notificó el evento. Para obtener más información, consulte la tabla Dispositivos. Para los eventos de SMS, este objeto está vacío.
`session`	Para los eventos de SMS, este objeto está vacío.
`attributes`	Atributos asociados al evento. Para los eventos notificados por una de sus aplicaciones, este objeto puede incluir atributos personalizados definidos por la aplicación. Para los eventos que se crean al enviar una campaña, este objeto contiene atributos asociados a la campaña. Para los eventos que se generan al enviar mensajes de transacciones, este objeto contiene información relacionada con el propio mensaje. Consulte la tabla de la sección Attributes (Atributos) para obtener más información.
`metrics`	Métricas adicionales asociadas al evento. Para obtener más información, consulte la tabla Métricas.
`awsAccountId`	El ID de la cuenta de AWS de que se utilizó para enviar el mensaje.

Aplicación

Incluye información sobre el proyecto de Amazon Pinpoint al que está asociado el evento y, si procede, el SDK que se utilizó para informar del evento.

Atributo	Descripción
`app_id`	El ID único del proyecto de Amazon Pinpoint que notificó el evento.
`sdk`	El SDK que se utilizó para notificar el evento. Si envía un mensaje SMS transaccional llamando directamente a la API de Amazon Pinpoint o con la consola de Amazon Pinpoint, este objeto estará vacío.

Atributos

Incluye información sobre los atributos asociados al evento.

Atributo	Descripción
`sender_request_id`	Un ID único asociado a la solicitud para enviar el mensaje SMS.
`campaign_activity_id`	El ID único de la actividad dentro de la campaña.
`origination_phone_number`	El número de teléfono desde el que se envió el mensaje.
`destination_phone_number`	El número de teléfono al que se ha intentado enviar el mensaje.
`record_status`	Información adicional sobre el estado del mensaje. Los valores posibles son: • SUCCESSFUL/DELIVERED: el mensaje se entregó de forma satisfactoria. • PENDING: el mensaje aún no se ha entregado al dispositivo del destinatario.

Atributo	Descripción
	• INVALID: el número de teléfono de destino no es válido. • UNREACHABLE: el dispositivo del destinatario no está disponible en este momento. Por ejemplo, el dispositivo podría estar apagado o desconectado de la red. Puede intentar enviar el mensaje de nuevo más tarde. • UNKNOWN: se ha producido un error que ha impedido la entrega del mensaje. Este error suele ser transitorio y es posible intentar enviar el mensaje de nuevo más tarde. • BLOCKED: el dispositivo del destinatario bloquea los mensajes SMS del número de origen. • CARRIER_UNREACHABLE: un problema con la red móvil del destinatario impidió que se entregara el mensaje. Este error suele ser transitorio y es posible intentar enviar el mensaje de nuevo más tarde. • SPAM: el operador de telefonía móvil del destinatario identificó el contenido del mensaje como spam y bloqueó la entrega del mensaje. • INVALID_MESSAGE: el cuerpo del mensaje SMS no es válido y no se puede entregar. • CARRIER_BLOCKED: el operador del destinatario ha bloqueado la entrega de este mensaje. Esto suele ocurrir cuando el operador identifica el contenido del mensaje como no solicitado o malintencionado. • TTL_EXPIRED: el mensaje SMS no se pudo entregar en un periodo de tiempo determinado. Este error suele ser transitorio y es

Atributo	Descripción
	posible intentar enviar el mensaje de nuevo más tarde. • MAX_PRICE_EXCEEED: el envío del mensaje habría supuesto un cargo superior a la cuota mensual de gastos de SMS de la cuenta. Puede solicitar un aumento de esta cuota completando el procedimiento descrito en Solicitud de aumentos de la cuota de gasto mensual de SMS en la Guía del usuario de Amazon Pinpoint. • OPTED_OUT: el mensaje SMS no se envió porque el destinatario desactivó la recepción de mensajes de usted. • NO_QUOTA_LEFT_ON_ACCOUNT: no queda suficiente cuota de gasto en la cuenta para enviar el mensaje. Puede solicitar un aumento de esta cuota completando el procedimiento descrito en Solicitud de aumento de la cuota de gasto mensual de SMS en la Guía del usuario de Mensajes SMS para usuarios finales de AWS. • NO_ORIGINATION_IDENTITY_AVAILABLE_TO_SEND: la cuenta no contiene un número de teléfono que se pueda usar para enviar el mensaje al destino. • DESTINATION_COUNTRY_NOT_SUPPORTED: el país de destino está bloqueado. Para obtener más información sobre los países admitidos, consulte los países y regiones admitidos (canal de SMS) en la Guía del usuario de Mensajes SMS para usuarios finales de AWS. • ACCOUNT_IN_SANDBOX: la cuenta está en un entorno aislado y solo puede enviar

Atributo	Descripción
	a números de destino verificados. Puede verificar el número de destino en la consola de Amazon Pinpoint o iniciar el proceso para sacar la cuenta del entorno de pruebas. Consulte acerca del entorno pruebas de SMS/MMS y voz en la Guía del usuario de Mensajes SMS para usuarios finales de AWS. • RATE_EXCEEDED: ha intentado enviar un mensaje demasiado rápido y se ha producido un error. Necesita reducir la velocidad de las llamadas. Para obtener más informaci ón sobre nuestros límites, consulte Message Parts per Second (MPS) limits en la Guía del usuario de Mensajes SMS para usuarios finales de AWS. • INVALID_ORIGINATION_IDENTITY: la identidad de origen proporcionada no es válida. • ORIGINATION_IDENTITY_DOES_N OT_EXIST: la identidad de origen proporcio nada no existe. • INVALID_DLT_PARAMETERS: se proporcio naron parámetros de DLT no válidos (requerido para destinos en India). • INVALID_PARAMETERS: se proporcionaron parámetros no válidos. • ACCESS_DENIED: la cuenta no puede enviar mensajes. Contacte con el servicio de atención al cliente para averiguar la causa y resolver el problema. • INVALID_KEYWORD: la palabra clave proporcionada no es válida. Es posible que la

Atributo	Descripción
	palabra clave tenga un formato incorrecto o que no esté configurada en la cuenta. • INVALID_SENDER_ID: el ID de remitente proporcionado no es válido. El ID del remitente puede tener un formato o una longitud incorrectos. • INVALID_POOL_ID: el ID de grupo proporcionado no es válido. El ID de grupo puede estar en un formato incorrecto o no pertenecer a la cuenta. • SENDER_ID_NOT_SUPPORTED_FOR_DESTINATION: el país de destino no admite el ID de remitente. Tiene que usar un número de teléfono u otra identidad de origen para enviar. • INVALID_PHONE_NUMBER: el número de teléfono de origen proporcionado no es válido. El número de teléfono puede tener un formato o una longitud incorrectos.
`iso_country_code`	El país que está asociado al número de teléfono del destinatario, mostrado en formato ISO 3166-1 alpha-2.
`treatment_id`	El ID del tratamiento del mensaje, si el mensaje se envió en una campaña A/B.
`treatment_id`	Si el mensaje se envió mediante una campaña de prueba A/B, este valor representa el número de tratamiento del mensaje. Para los mensajes SMS de transacciones, este valor es 0.

Atributo	Descripción
`number_of_message_parts`	El número de partes del mensaje que Amazon Pinpoint ha creado para enviar el mensaje. Por lo general, los mensajes SMS solo pueden contener 160 caracteres GSM-7 o 67 caractere s no GSM, aunque estos límites pueden variar según el país . Si envía un mensaje que supere estos límites, Amazon Pinpoint lo divide automáticamente en partes más pequeñas. La facturación se realiza en función del número de partes de mensajes que envíe.
`message_id`	El ID único que Amazon Pinpoint genera cuando acepta el mensaje.
`message_type`	Tipo de mensaje. Los valores posibles son Promotional (Promocionales) y Transactional (Transaccionales). Este valor se especifica al crear una campaña o al enviar mensajes de transacciones mediante la operación SendMessages de la API de Amazon Pinpoint.
`campaign_id`	El ID único de la campaña DE Amazon Pinpoint que envió el mensaje.
`customer_context`	Una cadena JSON del contenido de la asignación `Context` enviada en una operación SendMessages de Amazon Pinpoint.

Cliente

Incluye información sobre el cliente de la aplicación instalada en el dispositivo que notificó el evento.

Atributo	Descripción
`client_id`	Para los eventos generados por aplicaciones, este valor es el ID único del cliente de la aplicación instalada en el dispositivo. AWS Mobile SDK for iOS y AWS Mobile SDK para Android generan automáticamente este ID. Para los eventos que se generan al enviar campañas y mensajes de transacciones, este valor es igual al ID del punto de enlace al que envía el mensaje.
`cognito_id`	El ID exclusivo asignado al cliente de la aplicación en el grupo de identidades de Amazon Cognito utilizado por la aplicación.

Dispositivo

Incluye información sobre el dispositivo que notificó el evento.

Atributo	Descripción
`locale`	La configuración regional del dispositivo.
`make`	La marca del dispositivo, como, por ejemplo, `Apple` o `Samsung`.
`model`	El modelo del dispositivo, como, por ejemplo, `iPhone`.
`platform`	La plataforma del dispositivo, como, por ejemplo, `ios` o `android`.

Métricas

Incluye información sobre las métricas asociadas al evento.

Atributo	Descripción
`price_in_millicents_usd`	La cantidad que se cobra por enviar el mensaje. Este precio se muestra en milésimas de centavo de dólar estadounidense. Por ejemplo, si el valor de este atributo es 645, se le cobran 0,645 ¢ por enviar el mensaje (645 / 1000 = 0,645 ¢ = 0,00645 USD). **Note** Esta propiedad no aparece en los mensajes con el `event_type` _SMS.BUFFERED.

Eliminar un flujo de eventos de Amazon Pinpoint

Si se asigna una secuencia de Kinesis a una aplicación, puede desactivar la transmisión de eventos para esa aplicación. Amazon Pinpoint deja de transmitir los eventos a Kinesis, pero puede ver los análisis de los eventos mediante la consola de Amazon Pinpoint.

AWS CLI

Utilice el comando `delete-event-stream`:

```
aws pinpoint delete-event-stream --application-id application-id
```

AWS SDK para Java

Utilice el método `deleteEventStream` del cliente de Amazon Pinpoint:

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

Consultar datos de análisis de Amazon Pinpoint

Además de utilizar las páginas de análisis de la consola Amazon Pinpoint, puede utilizar Amazon Pinpoint APIs Analytics para consultar los datos de análisis de un subconjunto de métricas estándar que proporcionan información sobre las tendencias relacionadas con la participación de los usuarios, el alcance de las campañas y mucho más. Estas métricas, también denominadas indicadores clave de rendimiento (KPIs), son valores medibles que pueden ayudarle a supervisar y evaluar el rendimiento de sus proyectos, campañas y procesos.

Si lo utiliza APIs para consultar datos de análisis, puede analizarlos mediante la herramienta de informes que prefiera, sin tener que iniciar sesión en la consola de Amazon Pinpoint ni analizar datos de eventos sin procesar de fuentes como las transmisiones de Amazon Kinesis. Por ejemplo, puede crear un panel personalizado que muestre los resultados semanales de las campañas o que proporcione análisis detallados sobre las tasas de entrega de las campañas.

Puede consultar los datos mediante la API REST de Amazon Pinpoint, el AWS Command Line Interface (AWS CLI) o un AWS SDK. Para consultar los datos, envíe una solicitud a la API de Amazon Pinpoint y utilice los parámetros compatibles para especificar los datos que desee y los filtros que quiere aplicar. Después de enviar la consulta, Amazon Pinpoint devuelve los resultados de la consulta en una respuesta JSON. A continuación, puede transferir los resultados a otro servicio o aplicación para un análisis más exhaustivo, para su almacenamiento o para la generación de informes.

Amazon Pinpoint recopila y acumula automáticamente datos de todas las métricas compatibles para todos los proyectos, campañas y recorridos. Además, los datos se actualizan continuamente, con lo que el periodo de latencia de los datos es de tan solo dos horas aproximadamente. No obstante, tenga en cuenta que puede haber latencia de datos adicional para ciertas métricas. Esto se debe a que los datos de algunas métricas se basan en la información que recibimos de los proveedores de correo electrónico de los destinatarios. Algunos proveedores nos envían esta información inmediatamente, mientras que otros la envían con menos frecuencia.

Amazon Pinpoint almacena los datos durante 90 días. Para almacenar los datos durante más de 90 días o para acceder a los datos de análisis sin procesar en tiempo real, puede configurar un proyecto de Amazon Pinpoint para que transmita datos de eventos a Amazon Kinesis Data Streams o Amazon Data Firehose. Para obtener información sobre la configuración de secuencias de eventos, consulte Transmitir datos de eventos de aplicación a través de Kinesis y Firehose con Amazon Pinpoint.

Consultar componentes y parámetros de las métricas en Amazon Pinpoint

Para consultar los datos de una métrica, envíe una solicitud `get` al recurso de métricas adecuado de la API de Amazon Pinpoint. En la solicitud, para definir la consulta debe utilizar parámetros compatibles con los siguientes componentes de consulta:

- Proyecto: especifique un proyecto proporcionando el ID del proyecto como valor del parámetro `application-id`. Este parámetro es obligatorio para todas las métricas.
- Campaña: especifique una campaña proporcionando el ID de la campaña como valor del parámetro `campaign-id`. Este parámetro solo es obligatorio para las métricas de campaña.
- Recorrido: especifique un recorrido proporcionando el ID del recorrido como valor del parámetro `journey-id`. Este parámetro solo es necesario para métricas de interacción y ejecución de recorrido y para métricas de ejecución de actividades de recorrido.
- Actividad de recorrido: especifique una actividad de recorrido proporcionando el ID de la actividad de recorrido como valor del parámetro `journey-activity-id`. Este parámetro solo es necesario para métricas de ejecución de actividades de recorrido.
- Intervalo de fechas: si lo desea, puede filtrar los datos por intervalo de fechas, indique la primera y la última fecha y la hora del intervalo de fechas mediante los parámetros de hora de inicio y finalización compatibles. Los valores deben estar en formato ISO 8601 extendido y utilizar la hora universal coordinada (UTC), por ejemplo, `2019-07-19T20:00:00Z` para el 19 de julio de 2019 a las 20:00 UTC.

  Los intervalos de fechas son inclusivos y deben tener 31 días naturales o menos. Además, la primera fecha y hora deben ser inferiores a 90 días desde el día actual. Si no especifica un intervalo de fechas, Amazon Pinpoint devuelve los datos de los 31 días naturales anteriores. Todos los parámetros del intervalo de fechas son compatibles con todas las métricas, excepto las métricas de ejecución de recorrido y las métricas de ejecución de actividades de recorrido.
- Métrica: especifique la métrica proporcionando el nombre de la métrica como valor del parámetro `kpi-name`. Este valor describe la métrica asociada y consta de dos o más términos, que se componen de caracteres alfanuméricos en minúsculas, separados por un guion. Algunos ejemplos son `email-open-rate` y `successful-delivery-rate`. Este parámetro es obligatorio para todas las métricas, excepto las métricas de ejecución de recorrido y las métricas de ejecución de actividades de recorrido. Para obtener una lista completa de las métricas admitidas y del valor `kpi-name` que se va a utilizar para cada una, consulte Métricas estándar para proyectos, campañas y recorridos.

Después de enviar la consulta, Amazon Pinpoint devuelve los resultados de la consulta en una respuesta JSON. En la respuesta, la estructura de los resultados varía según la métrica que consultó.

Algunas métricas proporcionan solo un valor, por ejemplo, la cantidad de mensajes que entregó una campaña. Otras métricas proporcionan varios valores y, por lo general, los agrupan por un campo relevante, por ejemplo, el número de mensajes que se entregaron en cada ejecución de una campaña, agrupados por ejecución de campaña. Si una métrica proporciona y agrupa varios valores, la respuesta JSON incluye un campo que indica qué campo se utilizó para agrupar los datos. Para obtener más información sobre la estructura de los resultados de la consulta, lea Usar resultados de consultas JSON.

Políticas de IAM para consultar datos de análisis de Amazon Pinpoint

Con la API de Amazon Pinpoint, puede consultar los datos de análisis de un subconjunto de métricas estándar, también denominadas indicadores clave de rendimiento (KPIs), que se aplican a los proyectos, campañas y recorridos de Amazon Pinpoint. Estas métricas pueden ayudarle a supervisar y evaluar el rendimiento de los proyectos, las campañas y los recorridos.

Para administrar el acceso a estos datos, puede crear políticas AWS Identity and Access Management (de IAM) que definan los permisos para las funciones de IAM o los usuarios que están autorizados a acceder a los datos. Para permitir el control detallado del acceso a estos datos, Amazon Pinpoint proporciona varias acciones distintas que se pueden especificar en las políticas de IAM. Hay una acción determinada para ver los datos de análisis en la consola de Amazon Pinpoint (`mobiletargeting:GetReports`) y hay otras acciones para acceder a los datos de análisis mediante programación con la API de Amazon Pinpoint.

Para crear políticas de IAM que gestionen el acceso a los datos de análisis, puede utilizar la Consola de administración de AWS, la o la API AWS CLI de IAM. Tenga en cuenta que la pestaña Editor visual en la Consola de administración de AWS actualmente no incluye acciones para ver o consultar los datos de análisis de Amazon Pinpoint. Sin embargo, puede agregar las acciones necesarias a las políticas de IAM manualmente desde la pestaña JSON de la consola.

Por ejemplo, la siguiente política permite el acceso programático a todos los datos de análisis de todos sus proyectos, campañas y recorridos en todas las regiones: AWS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryAllAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:*:111122223333:apps/*/kpis/*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/campaigns/*/kpis/
*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/kpis/
*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/
execution-metrics",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/
activities/*/execution-metrics"
            ]
        }
    ]
}
```

¿Dónde *accountId* está el ID de tu AWS cuenta?

Sin embargo, recomendamos que las políticas se creen según el principio de privilegios mínimos. En otras palabras, debe crear políticas que incluyan solo los permisos necesarios para realizar una tarea específica. Para respaldar esta práctica e implementar un control más detallado, puede restringir el acceso programático a los datos de análisis únicamente para un proyecto concreto en una AWS región específica, por ejemplo:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryProjectAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/kpis/
*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/campaigns/*/
kpis/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
kpis/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
execution-metrics",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
activities/*/execution-metrics"
            ]
        }
    ]
}
```

Donde:

- *region*es el nombre de la AWS región que aloja el proyecto.
- *accountId*es el ID AWS de tu cuenta.
- *projectId*es el identificador del proyecto al que quieres proporcionar acceso.

Del mismo modo, la política de ejemplo siguiente permite el acceso mediante programación a los datos de análisis solo para una campaña concreta:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryCampaignAnalytics",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetCampaignDateRangeKpi",
            "Resource": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/projectId/campaigns/campaignId/kpis/*"
        }
    ]
}
```

Donde:

- *region*es el nombre de la AWS región que aloja el proyecto.
- *accountId*es tu Cuenta de AWS ID.
- *projectId*es el identificador del proyecto asociado a la campaña.
- *campaignId*es el identificador de la campaña a la que quieres proporcionar acceso.

Y la siguiente política de ejemplo permite el acceso mediante programación a todos los datos de análisis, tanto datos de interacción como de ejecución, de un recorrido particular y las actividades que lo componen:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryJourneyAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
```

```
            ],
            "Resource": [
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/kpis/*",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/execution-metrics",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/activities/*/execution-metrics"
            ]
        }
    ]
}
```

Donde:

- *region*es el nombre de la AWS región que aloja el proyecto.
- *accountId*es el ID AWS de tu cuenta.
- *projectId*es el identificador del proyecto asociado al viaje.
- *journeyId*es el identificador del viaje al que quieres dar acceso.

Para obtener una lista completa de las acciones de la API de Amazon Pinpoint que puede utilizar en las políticas de IAM, consulte Acciones de Amazon Pinpoint para las políticas de IAM. Para obtener más información sobre cómo crear y administrar políticas de IAM, consulte la Guía del usuario de IAM.

Métricas estándar que se aplican a los proyectos, campañas y recorridos de Amazon Pinpoint

Puede usar Amazon Pinpoint Analytics APIs para consultar datos de análisis de un subconjunto de métricas estándar que se aplican a los proyectos, campañas y recorridos de Amazon Pinpoint. Estas métricas, también denominadas indicadores clave de rendimiento (KPIs), son valores medibles que pueden ayudarle a supervisar y evaluar el rendimiento de los proyectos, las campañas y las trayectorias.

Amazon Pinpoint proporciona acceso mediante programación a los datos de análisis de varios tipos de métricas estándar:

- Métricas de aplicaciones: estas métricas proporcionan información sobre las tendencias de todas las campañas y mensajes transaccionales asociados a un proyecto, también denominado aplicación. Por ejemplo, puede utilizar una métrica de aplicación para ver un desglose del número de mensajes de la campaña que abrieron los destinatarios de cada campaña asociada a un proyecto.
- Métricas de campaña: estas métricas proporcionan información sobre el rendimiento de las campañas individuales. Puede utilizar una métrica de campaña para, por ejemplo, determinar el número de puntos de enlace al que se envió un mensaje de una campaña o sobre el número de mensajes entregados a los puntos de enlace.
- Métricas de participación en recorridos: estas métricas proporcionan información sobre el rendimiento de los recorridos individuales. Por ejemplo, puede utilizar una métrica de interacción del recorrido para obtener un desglose del número de mensajes abiertos por los participantes en cada actividad de un recorrido.
- Métricas de ejecución en recorridos: estas métricas proporcionan información sobre las tendencias de participación en los recorridos individuales. Por ejemplo, puede utilizar una métrica de ejecución de recorrido para determinar cuántos participantes han iniciado un recorrido.
- Métricas de ejecución de actividades de los recorridos: estas métricas proporcionan información sobre las tendencias de participación en los recorridos individuales. Por ejemplo, puede utilizar una métrica de ejecución de actividades de recorrido para determinar cuántos participantes han iniciado una actividad y cuántos participantes han completado cada ruta de una actividad.

Los temas de esta sección enumeran y describen las métricas individuales que puede consultar para cada tipo de métrica.

Temas

- Métricas de aplicación de Amazon Pinpoint para campañas
- Métricas de aplicación de Amazon Pinpoint para mensajes de correo electrónico transaccionales
- Métricas de aplicación de Amazon Pinpoint para mensajes SMS transaccionales
- Métricas de campañas de Amazon Pinpoint
- Métricas de interacción de recorrido de Amazon Pinpoint
- Métricas de ejecución de recorrido de Amazon Pinpoint
- Métricas de ejecución de actividades de recorrido de Amazon Pinpoint
- Métricas de ejecución de recorrido y campaña de Amazon Pinpoint

Métricas de aplicación de Amazon Pinpoint para campañas

En la siguiente tabla se muestran y describen las métricas de aplicación estándar que puede consultar para evaluar todas las campañas asociadas a un proyecto de Amazon Pinpoint. Para consultar los datos de estas métricas, utilice el recurso Métricas de aplicación de la API de Amazon Pinpoint. En la tabla, la columna kpi-name indica el valor que se debe utilizar para el parámetro `kpi-name` en la consulta.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de entregas	`successful-delivery-rate`	Para todas las campañas que están asociadas con un proyecto, el porcentaje de mensajes que se entregaron a los destinatarios. Esta métrica se calcula como el número de mensajes enviados por todas las campañas de un proyecto y entregados a los destinatarios, dividido entre el número de mensajes enviados por todas esas campañas.
Porcentaje de entregas, agrupadas por fecha	`successful-delivery-rate-grouped-by-date`	Para todas las campañas que están asociadas a un proyecto, el porcentaje de mensajes que se proporcio naron a los destinatarios cada día del intervalo de fechas especificado. Esta métrica se calcula como el número de mensajes enviados por todas las campañas de un proyecto y entregados a los destinata

Métrica	Kpi-name	Description (Descripción)
		rios, dividido por el número de mensajes enviados por todas esas campañas, durante cada día del intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Porcentaje de correo electróni cos abiertos	`email-open-rate`	Para todas las campañas que están asociadas con un proyecto, el porcentaje de mensajes de correo electróni co que abrieron los destinata rios. Esta métrica se calcula como el número de mensajes de correo electrónico enviados por todas las campañas de un proyecto y abiertos por los destinatarios, dividido por el número de mensajes de correo electrónico enviados por todas esas campañas y entregados a los destinatarios.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de correos electrónicos abiertos, agrupados por campaña	`email-open-rate-grouped-by-campaign`	Para cada campaña asociada a un proyecto, el porcentaje de mensajes de correo electrónico abiertos por los destinatarios. Esta métrica se calcula como el número de mensajes de correo electrónico enviados por una campaña y abiertos por los destinatarios, dividido por el número de mensajes de correo electrónico enviados por la campaña y entregados a los destinatarios. Los resultados de la consulta de esta métrica se agrupan por ID de campaña (`CampaignId`), que es una cadena que identifica de forma inequívoca una campaña.
Entregas a puntos de enlace	`unique-deliveries`	Para todas las campañas que están asociadas con un proyecto, el número de puntos de enlace distintos a los que se entregaron mensajes.

Métrica	Kpi-name	Description (Descripción)
Entregas a puntos de enlace, agrupadas por campaña	`unique-deliveries-grouped-by-campaign`	Para cada campaña asociada a un proyecto, el número de puntos de enlace distintos a los que se entregaron mensajes. Los resultados de la consulta de esta métrica se agrupan por ID de campaña (`CampaignId`), que es una cadena que identifica de forma inequívoca una campaña.
Entregas a puntos de enlace, agrupadas por fecha	`unique-deliveries-grouped-by-date`	Para todas las campañas que están asociadas con un proyecto, el número de puntos de enlace distintos a los que entregaron mensajes cada día del intervalo de tiempo especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.

Métrica	Kpi-name	Description (Descripción)
Mensajes entregados, agrupados por campaña	`successful-deliveries-grouped-by-campaign`	Para cada campaña asociada a un proyecto, el número de mensajes que se entregaron a los destinatarios. Esta métrica se calcula como el número de mensajes enviados por una campaña, menos el número de mensajes enviados por la campaña que no se pudieron entregar a los destinatarios debido a un rebote permanente. Los resultados de la consulta de esta métrica se agrupan por ID de campaña (`CampaignId`), que es una cadena que identifica de forma inequívoca una campaña.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de notificaciones push abiertas	`push-open-rate`	Para todas las campañas que están asociadas con un proyecto, el porcentaje de notificaciones push que abrieron los destinatarios. Esta métrica se calcula como el número de notificaciones push enviadas por todas las campañas de un proyecto y abiertas por los destinatarios, dividido entre el número de notificaciones push enviadas por todas esas campañas y entregadas a los destinatarios.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de notificaciones push abiertas, agrupadas por campaña	`push-open-rate-grouped-by-campaign`	Para cada campaña asociada a un proyecto, el porcentaje de notificaciones push que abrieron los destinatarios. Esta métrica se calcula como el número de notificaciones push enviadas por una campaña y abiertas por los destinatarios, dividido entre el número de notificaciones push enviadas por la campaña y entregadas a los destinatarios. Los resultados de la consulta de esta métrica se agrupan por ID de campaña (`CampaignId`), que es una cadena que identifica de forma inequívoca una campaña.

Métricas de aplicación de Amazon Pinpoint para mensajes de correo electrónico transaccionales

En la siguiente tabla se muestran y describen las métricas de aplicaciones estándar que puede consultar para monitorear las tendencias de todos los mensajes de correo electrónico transaccionales asociados a un proyecto de Amazon Pinpoint. Para consultar los datos de estas métricas, utilice el recurso Métricas de aplicación de la API de Amazon Pinpoint. En la tabla, la columna kpi-name indica el valor que se debe utilizar para el parámetro `kpi-name` en la consulta.

Tenga en cuenta que estas métricas no proporcionan datos sobre los mensajes de correo electrónico enviados por las campañas. Únicamente proporcionan datos sobre los mensajes de correo electrónico de transacciones. Para consultar los datos de mensajes enviados por una o varias campañas, utilice una métrica de campaña o una una métrica de aplicación para campañas.

Métrica	Kpi-name	Description (Descripción)
Clics	`txn-emails-clicked`	El número de veces que los destinatarios hicieron clic en los enlaces del mensaje. Si un único destinatario hizo clic en varios enlaces del mensaje o hizo clic varias veces en el mismo enlace, cada clic se incluye en el recuento.
Clics, agrupados por fecha	`txn-emails-clicked-grouped-by-date`	El número de veces que los destinatarios han hecho clic en enlaces en los mensajes, para cada día en el intervalo de fechas especificado. Si un único destinatario hizo clic en varios enlaces del mensaje o hizo clic varias veces en el mismo enlace, cada clic se incluye en el recuento. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Complaint rate (Tasa de reclamaciones)	`txn-emails-complaint-rate`	El porcentaje de mensajes notificados por los destinatarios como correo electrónico no solicitado o no deseado. Esta métrica se calcula como el número de mensajes notificados por los destinatarios como correo electrónico no solicitado o no deseado,

Métrica	Kpi-name	Description (Descripción)
		dividido por el número de mensajes enviados.
Tasa de reclamaciones, agrupada por fechas	`txn-emails-complaint-rate-grouped-by-date`	El porcentaje de mensajes notificados por los destinatarios como correo electrónico no solicitado o no deseado, para cada día en el intervalo de fechas especificado. Esta métrica se calcula como el número de mensajes notificados por los destinatarios como correo electrónico no solicitado o no deseado, dividido por el número de mensajes enviados, para cada día en el intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Reclamos	`txn-emails-with-complaints`	El número de mensajes notificados por los destinatarios como correo electrónico no solicitado o no deseado.

Métrica	Kpi-name	Description (Descripción)
Reclamaciones, agrupadas por fechas	`txn-emails-with-complaints-grouped-by-date`	El número de mensajes notificados por los destinatarios como correo electrónico no solicitado o no deseado, para cada día en el intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Entregas	`txn-emails-delivered`	El número de mensajes que se entregaron a los destinatarios. Esta métrica se calcula como el número de mensajes que se enviaron, menos el número de mensajes que no se pudieron entregar debido a un rebote temporal o permanente o porque se rechazaron. Se rechaza un mensaje si Amazon Pinpoint determina que el mensaje contiene malware. Amazon Pinpoint no intenta enviar mensajes rechazados.

Métrica	Kpi-name	Description (Descripción)
Entregas, agrupadas por fechas	`txn-emails-delivered-grouped-by-date`	El número de mensajes que se entregaron a los destinatarios, para cada día en el intervalo de fechas especificado. Esta métrica se calcula como el número de mensajes que se enviaron, menos el número de mensajes que no se pudieron entregar debido a un rebote temporal o permanente o porque se rechazaron, para cada día en el intervalo de fechas especificado. Se rechaza un mensaje si Amazon Pinpoint determina que el mensaje contiene malware. Amazon Pinpoint no intenta enviar mensajes rechazados. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de entregas	`txn-emails-delivery-rate`	El porcentaje de mensajes que se entregaron a los destinatarios. Esta métrica se calcula como el número de mensajes enviados y entregados a los destinatarios, dividido por el número de mensajes enviados.
Porcentaje de entregas, agrupadas por fecha	`txn-emails-delivery-rate-grouped-by-date`	El porcentaje de mensajes que se entregaron a los destinatarios, para cada día en el intervalo de fechas especificado. Esta métrica se calcula como el número de mensajes enviados y entregados a los destinatarios, dividido por el número de mensajes enviados, para cada día en el intervalo de fechas especific ado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.

Métrica	Kpi-name	Description (Descripción)
Rebotes permanentes	`txn-emails-hard-bounced`	El número de mensajes que no se han podido entregar a los destinatarios debido a un rebote permanente. Se produce una devolución permanente si un problema persistente impide la entrega de un mensaje, por ejemplo, si no existe la dirección de correo electrónico del destinatario.
Rebotes permanentes, agrupados por fechas	`txn-emails-hard-bounced-grouped-by-date`	El número de mensajes que no se pudieron entregar a los destinatarios debido a un rebote permanente, para cada día en el intervalo de fechas especificado. Se produce una devolución permanente si un problema persistente impide la entrega de un mensaje, por ejemplo, si no existe la dirección de correo electrónico del destinatario. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Aperturas	`txn-emails-opened`	El número de mensajes que han abierto los destinatarios.

Métrica	Kpi-name	Description (Descripción)
Abiertos, agrupados por fechas	`txn-emails-opened-grouped-by-date`	El número de mensajes que abrieron los destinatarios, para cada día en el intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Envíos	`txn-emails-sent`	El número de mensajes que se han enviado.
Envíos, agrupados por fechas	`txn-emails-sent-grouped-by-date`	El número de mensajes que se enviaron, para cada día en el intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Rebotes temporales	`txn-emails-soft-bounced`	El número de mensajes que no se han podido entregar a los destinatarios debido a un rebote temporal. Se produce una devolución temporal si un problema temporal evita que se entregue un mensaje, por ejemplo, si la bandeja de entrada de un destinatario está llena o cuando el servidor de recepción no está disponible temporalmente.

Métrica	Kpi-name	Description (Descripción)
Rebotes temporales, agrupados por fechas	`txn-emails-soft-bounced-grouped-by-date`	El número de mensajes que no se pudieron entregar a los destinatarios debido a un rebote temporal, para cada día en el intervalo de fechas especificado. Se produce una devolución temporal si un problema temporal evita que se entregue un mensaje, por ejemplo, si la bandeja de entrada de un destinatario está llena o cuando el servidor de recepción no está disponible temporalmente. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.

Métrica	Kpi-name	Description (Descripción)
Eventos de clic de usuario únicos	`txn-emails-unique-clicks`	El número de destinatarios únicos (puntos de enlace) que han hecho clic en enlaces en mensajes. A diferencia de la métrica Clicks (Clics) esta métrica indica el número de destinatarios únicos que han hecho clic en enlaces, no el número de eventos de clic que se han producido. Por ejemplo, si un único destinatario ha hecho clic en varios enlaces del mismo mensaje o ha hecho clic en el mismo enlace más de una vez, esta métrica notifica solo un evento de clic para ese destinatario.

Métrica	Kpi-name	Description (Descripción)
Eventos de clic de usuario únicos, agrupados por fechas	`txn-emails-unique-clicks-grouped-by-date`	El número de destinatarios únicos (puntos de enlace) que han hecho clic en enlaces en mensajes, para cada día en el intervalo de fechas especific ado. A diferencia de la métrica Clicks, grouped by date (Clics, agrupados por fecha), esta métrica indica el número de destinatarios únicos que han hecho clic en enlaces, no el número de eventos de clic que se han producido. Por ejemplo, si un único destinata rio ha hecho clic en varios enlaces del mismo mensaje o ha hecho clic en el mismo enlace más de una vez, esta métrica notifica solo un evento de clic para ese destinatario. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.

Métrica	Kpi-name	Description (Descripción)
Eventos de apertura de usuario únicos	`txn-emails-unique-opens`	El número de destinatarios únicos (puntos de enlace) que han abierto mensajes. A diferencia de la métrica Opens (Aperturas), esta métrica indica el número de destinatarios únicos que han abierto mensajes, no el número de eventos de apertura que se han producido . Por ejemplo, si un único destinatario abre el mismo mensaje varias veces, esta métrica notifica solo un evento de apertura para ese destinata rio.

Métrica	Kpi-name	Description (Descripción)
Eventos de apertura de usuario únicos, agrupados por fechas	`txn-emails-unique-opens-grouped-by-date`	El número de destinatarios únicos (puntos de enlace) que han abierto mensajes, para cada día en el intervalo de fechas especificado. A diferencia de la métrica Opens, grouped by date (Aperturas, agrupadas por fecha), esta métrica indica el número de destinatarios únicos que han abierto mensajes, no el número de eventos de apertura que se han producido. Por ejemplo, si un único destinatario abre el mismo mensaje varias veces, esta métrica notifica solo un evento de apertura para ese destinatario. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.

Métricas de aplicación de Amazon Pinpoint para mensajes SMS transaccionales

En la siguiente tabla se muestran y describen las métricas de aplicaciones estándar que puede consultar para monitorear las tendencias de todos los mensajes SMS transaccionales asociados a un proyecto de Amazon Pinpoint. Para consultar los datos de estas métricas, utilice el recurso Métricas de aplicación de la API de Amazon Pinpoint. En la tabla, la columna kpi-name indica el valor que se debe utilizar para el parámetro `kpi-name` en la consulta.

Tenga en cuenta que estas métricas no proporcionan datos sobre los mensajes SMS enviados por las campañas. Únicamente proporcionan datos sobre mensajes de SMS de transacción. Para consultar los datos de mensajes enviados por una o varias campañas, utilice una métrica de campaña o una una métrica de aplicación para campañas.

Métrica	Kpi-name	Description (Descripción)
Precio medio por mensaje, agrupado por país	`txn-sms-average-price-grouped-by-country`	El precio medio del envío de cada mensaje, para cada país o región adonde se enviaron los mensajes. El precio se muestra en milésimas de centavo de dólar estadouni dense. Por ejemplo, si el valor de este atributo es 645, se le cobran 0,645 ¢ por enviar el mensaje (645 / 1000 = 0,645 ¢ = 0,00645 USD). Esta métrica se calcula como precio total de todos los mensajes que se enviaron a los destinatarios en cada país o región, dividido por el número de mensajes que se enviaron a los destinatarios en cada uno de esos países y regiones. Los resultados de la consulta de esta métrica se agrupan por país o región, en formato ISO 3166-1 alfa-2.
Precio medio por parte del mensaje, agrupado por país	`txn-sms-average-price-by-parts-grouped-by-country`	El precio medio del envío de cada parte de mensaje, para cada país o región adonde se enviaron los mensajes.

Métrica	Kpi-name	Description (Descripción)
		Una parte de un mensaje es una parte de un mensaje SMS. El precio se muestra en milésimas de centavo de dólar estadounidense. Por ejemplo, si el valor de este atributo es 645, se le cobran 0,645 ¢ por enviar el mensaje (645 / 1000 = 0,645 ¢ = 0,00645 USD). Esta métrica se calcula como precio total de todas las partes de mensajes que se enviaron a los destinatarios en cada país o región, dividido por el número de partes de mensajes que se enviaron a los destinatarios en cada uno de esos países y regiones. Los resultados de la consulta de esta métrica se agrupan por país o región, en formato ISO 3166-1 alfa-2.
Entregas	`txn-sms-delivered`	El número de mensajes que se entregaron a los destinatarios.

Métrica	Kpi-name	Description (Descripción)
Entregas, agrupadas por país	`txn-sms-delivered-grouped-by-country`	El número de mensajes que se entregaron a los destinatarios, para cada país o región a donde se enviaron los mensajes. Los resultados de la consulta de esta métrica se agrupan por país o región, en formato ISO 3166-1 alfa-2.
Entregas, agrupadas por fechas	`txn-sms-delivered-grouped-by-date`	El número de mensajes que se entregaron a los destinatarios, para cada día en el intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Errores de entrega	`txn-sms-error-distribution`	El número de veces que se produjo un error al intentar entregar los mensajes, por cada tipo de error que se produjo. Los resultados de la consulta de esta métrica se agrupan por código de error, para cada tipo de error que se ha producido.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de entregas	`txn-sms-delivery-rate`	El porcentaje de mensajes que se entregaron a los destinatarios. Esta métrica se calcula como el número de mensajes enviados y entregados a los destinatarios, dividido por el número de mensajes enviados.
Porcentaje de entregas, agrupadas por fecha	`txn-sms-delivery-rate-grouped-by-date`	El porcentaje de mensajes que se entregaron a los destinatarios, para cada día en el intervalo de fechas especificado. Esta métrica se calcula como el número de mensajes enviados y entregados a los destinatarios, dividido por el número de mensajes enviados, para cada día en el intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.

Métrica	Kpi-name	Description (Descripción)
Partes del mensaje entregadas	`txn-sms-delivered-by-parts`	El número de partes de mensajes que se entregaron. Una parte de un mensaje es una parte de un mensaje SMS. Si un mensaje SMS contiene más caracteres de los que permite el protocolo SMS, Amazon Pinpoint divide el mensaje en tantas partes del mensaje como sea necesario para enviar el mensaje a un destinatario.
Partes del mensaje entregadas, agrupadas por país	`txn-sms-delivered-by-parts-grouped-by-country`	El número de partes de mensajes que se entregaron, para cada país o región a donde se enviaron los mensajes. Una parte de un mensaje es una parte de un mensaje SMS. Los resultados de la consulta de esta métrica se agrupan por país o región, en formato ISO 3166-1 alfa-2.

Métrica	Kpi-name	Description (Descripción)
Partes del mensaje enviadas	`txn-sms-sent-by-parts`	El número de partes de mensajes que se enviaron. Una parte de un mensaje es una parte de un mensaje SMS. Si un mensaje SMS contiene más caracteres de los que permite el protocolo SMS, Amazon Pinpoint divide el mensaje en tantas partes del mensaje como sea necesario para enviar el mensaje a un destinatario.
Partes del mensaje enviadas, agrupadas por país	`txn-sms-sent-by-parts-grouped-by-country`	El número de partes de mensajes que se enviaron, por cada país o región adonde se enviaron los mensajes. Una parte de un mensaje es una parte de un mensaje SMS. Los resultados de la consulta de esta métrica se agrupan por país o región, en formato ISO 3166-1 alfa-2.
Messages sent (Mensajes enviados)	`txn-sms-sent`	El número de mensajes que se han enviado.

Métrica	Kpi-name	Description (Descripción)
Mensajes enviados, agrupados por país	`txn-sms-sent-grouped-by-country`	El número de mensajes que se enviaron, por cada país o región adonde se enviaron los mensajes. Los resultados de la consulta de esta métrica se agrupan por país o región, en formato ISO 3166-1 alfa-2.
Mensajes enviados, agrupados por fecha	`txn-sms-sent-grouped-by-date`	El número de mensajes que se enviaron, para cada día en el intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Precio total, agrupado por país	`txn-sms-total-price-grouped-by-country`	El costo total de enviar los mensajes, para cada país o región adonde se enviaron los mensajes. El precio se muestra en milésimas de centavo de dólar estadounidense. Por ejemplo, si el valor de este atributo es 645, se le cobran 0,645 ¢ por enviar el mensaje (645 / 1000 = 0,645 ¢ = 0,00645 USD). Los resultados de la consulta de esta métrica se agrupan por país o región, en formato ISO 3166-1 alfa-2.

Métricas de campañas de Amazon Pinpoint

En la siguiente tabla se muestran y describen las métricas de campaña estándar que puede consultar para evaluar las tendencias de una campaña individual. Para consultar los datos de estas métricas, utilice el recurso Métricas de campaña de la API de Amazon Pinpoint. En la tabla, la columna kpi-name indica el valor que se debe utilizar para el parámetro `kpi-name` en la consulta.

Métrica	Kpi-name	Description (Descripción)
Bounce rate (Tasa de rebotes)	`hard-bounce-rate`	Para todas las ejecuciones de la campaña, el porcentaje de mensajes de correo electrónico que no se pudieron entregar a los destinatarios. Esta métrica mide solo las devoluciones permanentes, es decir, los mensajes en los que la dirección de correo electrónico del destinatario tenía un problema permanente que impedía la entrega del mensaje. Esta métrica se calcula como el número de mensajes de correo electrónico rebotados enviados por todas las ejecuciones de la campaña, dividido por el número de mensajes de correo electrónico enviados por todas esas ejecuciones de la campaña.
Porcentaje de rebotes, agrupados por ejecuciones de la campaña	`hard-bounce-rate-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el porcentaje de mensajes de correo electrónico que no se pudieron

Métrica	Kpi-name	Description (Descripción)
		entregar a los destinatarios. Esta métrica mide solo las devoluciones permanentes, es decir, los mensajes en los que la dirección de correo electrónico del destinatario tenía un problema permanent e que impedía la entrega del mensaje. Esta métrica se calcula como el número de mensajes de correo electrónico rebotados enviados por una ejecución de la campaña, dividido por el número de mensajes de correo electrónico enviados por la ejecución de la campaña. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignA ctivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de entregas	`successful-delivery-rate`	Para todas las ejecuciones de la campaña, el porcentaje de mensajes que se entregaron a los destinatarios. Esta métrica se calcula como el número de mensajes enviados por todas las ejecuciones de la campaña y entregados a los destinata rios, dividido por el número de mensajes enviados por todas esas ejecuciones de la campaña.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de entregas, agrupadas por ejecuciones de la campaña	`successful-delivery-rate-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el porcentaje de mensajes que se entregaron a los destinatarios. Esta métrica se calcula como el número de mensajes enviados por una ejecución de la campaña y entregados a los destinatarios, dividido por el número de mensajes enviados por la ejecución de la campaña. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignActivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de entregas, agrupadas por fecha	`successful-delivery-rate-grouped-by-date`	Para todas las ejecuciones de la campaña, el porcentaje de mensajes que se entregaron a los destinatarios durante cada día del intervalo de fechas especificado. Esta métrica se calcula como el número de mensajes enviados por todas las ejecuciones de la campaña y entregados a los destinatarios, dividido entre el número de mensajes enviados por todas esas ejecuciones de la campaña, durante cada día del intervalo de fechas especificado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de correo electrónicos abiertos	`email-open-rate`	Para todas las ejecuciones de la campaña, el porcentaje de mensajes de correo electrónico abiertos por los destinatarios. Esta métrica se calcula como el número de mensajes de correo electrónico enviados por todas las ejecuciones de la campaña y abiertos por los destinatarios, dividido entre el número de mensajes de correo electrónico enviados por todas esas ejecuciones de la campaña y entregados a los destinatarios.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de correos electrónicos abiertos, agrupados por ejecuciones de la campaña	`email-open-rate-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el porcentaje de mensajes de correo electrónico abiertos por los destinatarios. Esta métrica se calcula como el número de mensajes de correo electrónico enviados por una ejecución de la campaña y abiertos por los destinatarios, dividido entre el número de mensajes de correo electrónico enviados por la ejecución de la campaña y entregados a los destinatarios. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignActivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.

Métrica	Kpi-name	Description (Descripción)
Correos electrónicos abiertos, agrupados por ejecuciones de la campaña	`direct-email-opens-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el número de mensajes de correo electrónico abiertos por los destinatarios. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignActivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.
Entregas a puntos de enlace	`unique-deliveries`	Para todas las ejecuciones de la campaña, el número de puntos de enlace distintos a los que se entregaron mensajes.
Entregas a puntos de enlace, agrupadas por ejecuciones de la campaña	`unique-deliveries-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el número de puntos de enlace distintos a los que se entregaron mensajes. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignActivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.

Métrica	Kpi-name	Description (Descripción)
Entregas a puntos de enlace, agrupadas por fecha	`unique-deliveries-grouped-by-date`	Para todas las ejecuciones de la campaña, el número de puntos de enlace distintos a los que se entregaro n mensajes cada día del intervalo de fechas especific ado. Los resultados de la consulta de esta métrica se agrupan por día natural, en el formato ISO 8601 extendido.
Enlaces en los que se ha hecho clic, agrupados por ejecuciones de la campaña	`clicks-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el número de veces que los destinatarios hicieron clic en los enlaces del mensaje de correo electróni co. Si un único destinatario hizo clic en varios enlaces del mensaje o hizo clic varias veces en el mismo enlace, cada clic se incluye en el recuento. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignA ctivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.

Métrica	Kpi-name	Description (Descripción)
Mensajes entregados, agrupados por ejecuciones de la campaña	`successful-deliveries-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el número de mensajes que se entregaron a los destinatarios. Esta métrica se calcula como el número de mensajes enviados por una ejecución de la campaña, menos el número de mensajes que no se pudieron entregar a los destinatarios de la ejecución debido a un rebote permanente. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignActivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.

Métrica	Kpi-name	Description (Descripción)
Mensajes enviados, agrupados por ejecuciones de la campaña	`attempted-deliveries-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el número de mensajes que se enviaron. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignActivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.
Porcentaje de notificaciones push abiertas	`push-open-rate`	Para todas las ejecuciones de la campaña, el porcentaje de notificaciones push abiertas por los destinatarios. Esta métrica se calcula como el número de notificaciones push enviadas por todas las ejecuciones de la campaña y abiertas por los destinatarios, dividido entre el número de notificaciones push enviadas por todas esas ejecuciones de la campaña.

Métrica	Kpi-name	Description (Descripción)
Porcentaje de notificaciones push abiertas, agrupadas por ejecuciones de la campaña	`push-open-rate-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el porcentaje de notificaciones push abiertas por los destinatarios. Esta métrica se calcula como el número de notificaciones push enviadas por una ejecución de la campaña y abiertas por los destinatarios, dividido entre el número de notificaciones push enviadas por la ejecución de la campaña y entregadas a los destinatarios. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignActivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.

Métrica	Kpi-name	Description (Descripción)
Total de notificaciones push abiertas, agrupadas por ejecuciones de la campaña	`direct-push-opens-grouped-by-campaign-activity`	Para cada ejecución de la campaña, el número de notificaciones push abiertas por los destinatarios. Los resultados de la consulta de esta métrica se agrupan por ID de actividad de campaña (`CampaignActivityId`), que es una cadena que identifica de forma inequívoca una ejecución de la campaña.
Total SMS spend (Gasto total en SMS)	`sms-spend`	Para todas las campañas, la cantidad total de dinero, en milicents, gastada en el envío de SMS.

Métricas de interacción de recorrido de Amazon Pinpoint

En la siguiente tabla se muestran y describen las métricas de interacción del recorrido estándar que puede consultar para monitorear las tendencias de todos los mensajes de correo electrónico enviados por un recorrido de Amazon Pinpoint. Para consultar datos de estas métricas, utilice el recurso Métricas de interacción del recorrido de la API de Amazon Pinpoint. En la tabla, la columna kpi-name indica el valor que se debe utilizar para el parámetro `kpi-name` en la consulta.

Métrica	Kpi-name	Description (Descripción)
Clics	`journey-emails-clicked`	El número de veces que los participantes hicieron clic en los enlaces de los mensajes. Si un único participante hizo clic en varios enlaces del mensaje o hizo clic varias

Métrica	Kpi-name	Description (Descripción)
		veces en el mismo enlace, cada clic se incluye en el recuento.
Clics, agrupados por actividad	`emails-clicked-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de veces que los participantes hicieron clic en los enlaces de los mensajes. Si un único participante hizo clic en varios enlaces del mensaje o hizo clic varias veces en el mismo enlace, cada clic se incluye en el recuento. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identifica de forma inequívoca una actividad.
Reclamos	`journey-emails-complained`	El número de mensajes notificados por los participantes como correo electrónico no solicitado o no deseado.

Métrica	Kpi-name	Description (Descripción)
Reclamaciones, agrupadas por actividad	`emails-complained-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de mensajes notificados por los participantes como correo electrónico no solicitado o no deseado. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identific a de forma inequívoca una actividad.
Entregas	`journey-emails-del ivered`	El número de mensajes que se entregaron a los participa ntes. Esta métrica se calcula como el número de mensajes que se enviaron, menos el número de mensajes que no se pudieron entregar debido a un rebote temporal o permanente o porque se rechazaron.

Métrica	Kpi-name	Description (Descripción)
Entregas, agrupadas por actividad	`emails-delivered-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de mensajes que se entregaron a los participantes. Esta métrica se calcula como el número de mensajes que se enviaron, menos el número de mensajes que no se pudieron entregar debido a un rebote temporal o permanente o porque se rechazaron, para cada actividad en el recorrido. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identifica de forma inequívoca una actividad.
Rebotes permanentes	`journey-emails-hardbounced`	El número de mensajes que no se han podido entregar a los participantes debido a un rebote permanente. Se produce una devolución permanente si un problema persistente impide la entrega de un mensaje, por ejemplo, si no existe la dirección de correo electrónico del participante.

Métrica	Kpi-name	Description (Descripción)
Rebotes permanentes, agrupados por actividad	`emails-hardbounced-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de mensajes que no se han podido entregar a los participantes debido a un rebote permanente. Se produce una devolución permanente si un problema persistente impide la entrega de un mensaje, por ejemplo, si no existe la dirección de correo electrónico del participante. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identifica de forma inequívoca una actividad.
Aperturas	`journey-emails-opened`	El número de mensajes que han abierto los participantes.

Métrica	Kpi-name	Description (Descripción)
Abiertos, agrupados por actividad	`emails-opened-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de mensajes que abrieron los participantes. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identifica de forma inequívoca una actividad.
Rechazos	`journey-emails-rejected`	El número de mensajes que no se enviaron a los participantes porque fueron rechazados. Se rechaza un mensaje si Amazon Pinpoint determina que el mensaje contiene malware. Amazon Pinpoint no intenta enviar mensajes rechazados.

Métrica	Kpi-name	Description (Descripción)
Rechazos, agrupados por actividad	`emails-rejected-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de mensajes que no se enviaron a los participantes porque fueron rechazados. Se rechaza un mensaje si Amazon Pinpoint determina que el mensaje contiene malware. Amazon Pinpoint no intenta enviar mensajes rechazados. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identifica de forma inequívoca una actividad.
Envíos	`journey-emails-sent`	El número de mensajes que se han enviado.
Envíos, agrupados por actividad	`emails-sent-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de mensajes que se enviaron. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identifica de forma inequívoca una actividad.

Métrica	Kpi-name	Description (Descripción)
Rebotes temporales	`journey-emails-softbounced`	El número de mensajes que no se han podido entregar a los participantes debido a un rebote temporal. Se produce una devolución temporal si un problema temporal evita que se entregue un mensaje, por ejemplo, si la bandeja de entrada de un participante está llena o cuando el servidor de recepción no está disponible temporalmente.
Rebotes temporales, agrupados por actividad	`emails-softbounced-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de mensajes que no se han podido entregar a los participantes debido a un rebote temporal. Se produce una devolución temporal si un problema temporal evita que se entregue un mensaje, por ejemplo, si la bandeja de entrada de un participante está llena o cuando el servidor de recepción no está disponible temporalmente. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identifica de forma inequívoca una actividad.

Métrica	Kpi-name	Description (Descripción)
Cancelaciones de suscripciones	`journey-emails-unsubscribed`	El número de veces que los participantes hicieron clic en los enlaces de cancelación de suscripción en los mensajes. Si un único participante hizo clic en el mismo enlace de cancelación de suscripción varias veces, cada clic se incluye en el recuento.
Cancelaciones de suscripción, agrupadas por actividad	`emails-unsubscribed-grouped-by-journey-activity`	Para cada actividad de recorrido, el número de veces que los participantes hicieron clic en los enlaces de cancelación de suscripción de los mensajes. Si un único participante hizo clic en el mismo enlace de cancelación de suscripción varias veces, cada clic se incluye en el recuento. Los resultados de la consulta de esta métrica se agrupan por ID de actividad (`JourneyActivityId`), que es una cadena que identifica de forma inequívoca una actividad.

Métricas de ejecución de recorrido de Amazon Pinpoint

En la siguiente tabla se muestran y describen las métricas de ejecución estándar que puede consultar para evaluar el estado de los participantes en un recorrido de Amazon Pinpoint. Para consultar los datos de estas métricas, utilice el recurso Métricas de ejecución de recorrido de la API

de Amazon Pinpoint. La columna Campo de la tabla identifica el nombre del campo que aparece en los resultados de la consulta de cada métrica.

Métrica	Campo	Description (Descripción)
Participantes activos	`ENDPOINT_ACTIVE`	El número de participantes que avanzan activamente a través de las actividades del recorrido. Esta métrica se calcula como el número de participantes que han iniciado el recorrido, menos el número de participantes que abandonaron el recorrido y el número de participantes que se retiraron del recorrido.
Cancelaciones de participantes	`CANCELLED`	El número de participantes que no completaron el recorrido porque se canceló.
Salidas de los participantes	`ENDPOINT_LEFT`	El número de participantes que abandonaron el recorrido.
Entradas de participantes	`ENDPOINT_ENTERED`	El número de participantes que iniciaron el recorrido.
Excepciones de participantes, límites de re-entradas	`REENTRY_CAP_EXCEEDED`	El número de participantes que no completaron el recorrido porque habrían superado el número máximo de veces que un solo participante puede volver a entrar en el recorrido.

Métrica	Campo	Description (Descripción)
Excepciones de los participantes, rechazos	`ACTIVE_ENDPOINT_REJECTED`	El número de participantes que no pueden iniciar el recorrido porque ya son participantes activos en el recorrido. Un participante se rechaza si inicia un recorrido y, posteriormente, actualiza la definición de punto de enlace de una manera que afecte a su inclusión en un segmento (según criterios de segmento) o en el recorrido (según condiciones de actividad).

Métricas de ejecución de actividades de recorrido de Amazon Pinpoint

En la siguiente tabla se muestran y describen las métricas de ejecución estándar que puede consultar para evaluar el estado de los participantes en cada tipo de actividad individual de un recorrido de Amazon Pinpoint. Para consultar los datos de estas métricas, utilice el recurso Métricas de ejecución de actividad de recorrido de la API de Amazon Pinpoint. La columna Métricas de la tabla muestra los campos que aparecen en los resultados de la consulta para cada tipo de actividad. También proporciona una breve descripción de cada campo.

Tipo de actividad	Métricas
División sí/no (`CONDITIONAL_SPLIT`)	Las métricas son: • `Branch_FALSE`: el número de participantes que no cumplieron las condiciones de la actividad y procedieron a ella siguiendo la ruta “No”.

Tipo de actividad	Métricas
	• `Branch_TRUE` : el número de participantes que cumplieron las condiciones de la actividad y procedieron a ella siguiendo la ruta "Sí". Hay métricas adicionales disponibles para cada ruta de la actividad. Para obtener información sobre esas métricas, consulte la fila de esta tabla para ese tipo de actividad.
Retención (`HOLDOUT`)	Las métricas son: • `HOLDOUT`: el número de participantes eliminados del recorrido como parte del porcentaje de retención de la actividad. • `PASSED`: el número de participantes que pasaron a la siguiente actividad del recorrido.

Tipo de actividad	Métricas
Correo electrónico (`MESSAGE`)	Las métricas son: • `DAILY_CAP_EXCEEDED` : el número de mensajes que no se enviaron porque habrían superado el número máximo de mensajes que un solo participante puede recibir durante un periodo de 24 horas. • `FAILURE_PERMANENT` : el número de mensajes que no se enviaron debido a un problema permanente. • `QUIET_TIME` : el número de mensajes que no se enviaron porque se habrían entregado en un momento de silencio para la zona horaria del participante. • `SERVICE_FAILURE` : el número de mensajes que no se enviaron debido a un problema con Amazon Pinpoint. • `SUCCESS`: el número de mensajes que se entregaron de forma correcta a los participantes. • `THROTTLED` : el número de mensajes que no se enviaron porque al enviarlos se superarían las cuotas de envío de la cuenta de Amazon Pinpoint. • `TRANSIENT_FAILURE` : el número de mensajes que no se enviaron debido a un problema temporal. • `UNKNOWN`: el número de mensajes que no se enviaron debido a un problema desconocido.

Tipo de actividad	Métricas
División multivariante (`MULTI_CONDITIONAL_SPLIT`)	Para cada ruta de la actividad, el número de participantes que procedieron a la actividad en el camino. Los resultados de la consulta de esta métrica se agrupan por ruta, `Branch_#` donde `#` es el identificador numérico de una ruta, por ejemplo, `Branch_1`, de la primera ruta de la actividad. Hay métricas adicionales disponibles para cada ruta de la actividad. Para obtener información sobre esas métricas, consulte la fila de esta tabla para ese tipo de actividad.
División aleatoria (`RANDOM_SPLIT`)	Para cada ruta de la actividad, el número de participantes que procedieron a la actividad en el camino. Los resultados de la consulta de esta métrica se agrupan por ruta, `Branch_#` donde `#` es el identificador numérico de una ruta, por ejemplo, `Branch_1`, de la primera ruta de la actividad. Hay métricas adicionales disponibles para cada ruta de la actividad. Para obtener información sobre esas métricas, consulte la fila de esta tabla para ese tipo de actividad.

Tipo de actividad	Métricas
Espera (`WAIT`)	Las métricas son: • `WAIT_FINISHED` : el número de participantes que terminaron de esperar durante el tiempo especificado. • `WAIT_SKIPPED` : el número de participantes que no esperaron durante el tiempo especificado, normalmente porque comenzaron la actividad o el recorrido después de la hora de finalización programada para la actividad. • `WAIT_STARTED` : el número de participantes que empezó esperando y que no ha omitido ni terminado la espera durante el tiempo especificado.

Tipo de actividad	Métricas
Centro de contacto (`CONTACT_CENTER`)	Las métricas son: • `CALL_QUEUED` : el número de participantes a los que se ha llamado y puesto en cola para acceder a Amazon Connect. Incluye los intentos de volver a marcar. • `CONTINUE_WAITING` : el número de participantes que siguen esperando a que se realicen intentos de marcado. • `DIAL_FAILURE` : el número de participantes con intentos diarios erróneos. • `DROPPED`: el número de participantes que ya no cumplen las condiciones definidas en las actividades del recorrido anteriores en el momento del envío. • `TIMEOUT`: el número de participantes que no recibieron ningún código de disposición de Amazon Connect tras varios intentos de marcado. • `WAIT_FINISHED` : el número de participa ntes que terminaron de esperar durante el tiempo especificado. • `WAIT_FOR_QUIET_HOURS` : el número de participantes que esperan a que termine el tiempo de silencio para poder realizar la entrega al canal. • `WAIT_STARTED` : el número de participantes que empezó esperando y que no ha omitido ni terminado la espera durante el tiempo especificado.

Métricas de ejecución de recorrido y campaña de Amazon Pinpoint

Puede consultar las métricas de ejecución estándar para evaluar el estado de los participantes en cada tipo de actividad individual de un recorrido o campaña de Amazon Pinpoint. Para consultar los datos para estas métricas, utilice el recurso Métricas de ejecución de actividades de recorrido o Métricas de campaña de la API de Amazon Pinpoint. La siguiente tabla muestra los campos que aparecen en los resultados de la consulta para cada tipo de actividad.

Nombre de métrica	Se aplica a recorridos, campañas o ambos	Description (Descripción)
ENDPOINT_PRODUCED	Ambos	El número de puntos de conexión generados inicialmente a partir del segmento o evento antes de cualquier filtrado.
ENDPOINTS_FROM_USER	Ambos	Si el cliente tiene un segmento solo con el ID de usuario, se agregarán todos los puntos de conexión de esos usuarios. Esta métrica mide la cantidad de puntos de conexión que se agregan de esta manera.
ENDPOINT_OPT_OUT	Ambos	El punto de conexión se desactivó y no ingresó en la campaña ni en el recorrido.
ENDPOINT_INACTIVE	Ambos	El punto de conexión estaba inactivo y no ingresó en la campaña ni en el recorrido.
FILTERED_OUT_BY_SEGMENT	Ambos	El punto de conexión no coincidió con los filtros de segmentos y no ingresó en la campaña ni en el recorrido.

Nombre de métrica	Se aplica a recorridos, campañas o ambos	Description (Descripción)
ENDPOINT_MISSING_ADDRESS	Ambos	Al punto de conexión le faltaba una dirección y no ingresó en la campaña ni en el recorrido.
ENDPOINT_MISSING_CHANNEL	Ambos	Al punto de conexión le faltaba un canal y no ingresó en la campaña ni en el recorrido.
ENDPOINT_MISSING_TIMEZONE	Ambos	Al punto de conexión le faltaba un valor para la zona horaria y se filtró. Esto solo ocurre cuando se requiere un valor de zona horaria.
ENDPOINT_TIMEZONE_MISMATCH	Ambos	El punto de conexión estaba en una zona horaria que no estaba incluida en la ejecución en ese momento.
ENDPOINT_CHANNEL_MISMATCH	Campaigns (Campañas)	La campaña no tiene un mensaje configurado para este tipo de canal de punto de conexión.
DUPLICATE_ENDPOINT	Ambos	Se encontraron puntos de conexión duplicados y se eliminaron los duplicados.
DUPLICATE_USER	Ambos	Se encontraron usuarios duplicados y se eliminaron de un segmento solo con ID de usuario. Si tienen el mismo ID de usuario, se emitirá una métrica igual a 1.

Nombre de métrica	Se aplica a recorridos, campañas o ambos	Description (Descripción)
PAUSED	Journeys	Se retiró de la ejecución porque el recorrido estaba en pausa.
ENDED	Journeys	Se retiró de la ejecución porque el recorrido se terminó.
TREATMENT_HOLDOUT	Campaigns (Campañas)	Esto se emite en A/B las campañas, para los puntos de enlace cuyas cohortes no coinciden con el tratamiento actual. Por ejemplo, si A/B se divide al 50%, el 50% de los puntos finales emitirán esta métrica para cada tratamiento
ENDPOINT_ESTIMATED_TIMEZONE	Journeys	La estimación de la zona horaria permitió estimar una zona horaria para el punto de conexión.

Consultar datos de análisis de Amazon Pinpoint para campañas

Además de utilizar las páginas de análisis de la consola Amazon Pinpoint, puede utilizar Amazon Pinpoint APIs Analytics para consultar los datos de análisis de un subconjunto de métricas estándar que proporcionan información sobre las tendencias de entrega y participación de las campañas.

Cada una de estas métricas es un valor medible, también conocido como un indicador clave de rendimiento (KPI), que puede ayudarle a supervisar y evaluar el rendimiento de una o más campañas. Puede utilizar una métrica para, por ejemplo, averiguar el número de puntos de enlace a los que se envió un mensaje de una campaña o cuántos de esos mensajes se entregaron a los puntos de enlace previstos.

Amazon Pinpoint recopila y agrega automáticamente estos datos de todas las campañas. Almacena los datos durante 90 días. Si integró una aplicación móvil con Amazon Pinpoint mediante un SDK

AWS móvil, Amazon Pinpoint amplía esta compatibilidad para incluir métricas adicionales, como el porcentaje de notificaciones push que abrieron los destinatarios. Para obtener información sobre la integración de una aplicación móvil, consulte Integrar Amazon Pinpoint con la aplicación.

Si utiliza Amazon Pinpoint Analytics APIs para consultar datos, puede elegir varias opciones que definan el alcance, los datos, la agrupación y los filtros de la consulta. Para ello, utilice parámetros que especifiquen el proyecto, la campaña y la métrica que desea consultar, además de los filtros basados en fechas que desee aplicar.

En este tema se explica y se proporcionan ejemplos sobre cómo elegir estas opciones y consultar los datos de una o varias campañas.

Requisitos previos

Antes de consultar los datos de análisis de una o varias campañas, es recomendable recopilar la siguiente información, que después utilizará para definir la consulta:

- Project ID: es el identificador único para el proyecto que está asociado a la campaña o campañas. En la API de Amazon Pinpoint, este valor se almacena en la propiedad `application-id`. En la consola de Amazon Pinpoint, este valor se muestra como ID de proyecto en la página Todos los proyectos.
- ID de campaña: el identificador único de la campaña, si quiere consultar los datos de una sola campaña. En la API de Amazon Pinpoint, este valor se almacena en la propiedad `campaign-id`. Este valor no se muestra en la consola.
- Intervalo de fechas: de forma opcional, la primera y la última fecha y hora del intervalo de fechas para el que se van a consultar los datos. Los intervalos de fechas son inclusivos y deben tener 31 días naturales o menos. Además, deben comenzar en menos de 90 días a partir del día actual. Si no especifica un intervalo de fechas, Amazon Pinpoint consulta automáticamente los datos de los 31 días naturales anteriores.
- Tipo de métrica: el tipo de métrica que se va a consultar. Hay dos tipos de métricas: las métricas de aplicación y las métricas de campaña. Una métrica de aplicación proporciona datos para todas las campañas asociadas a un proyecto, también denominado aplicación. Una métrica de campaña proporciona datos de una sola campaña.
- Métrica: el nombre de la métrica que se va a consultar y, más específicamente, el valor `kpi-name` de la métrica. Para obtener una lista completa de las métricas admitidas y el valor `kpi-name` de cada una de ellas, consulte Métricas estándar para proyectos, campañas y recorridos.

También ayuda determinar si desea agrupar los datos por un campo relevante. Si lo hace, para simplificar el análisis y los informes, elija una métrica diseñada para agrupar datos de forma automática. Por ejemplo, Amazon Pinpoint proporciona varias métricas estándar que informan del porcentaje de mensajes entregados a los destinatarios de una campaña. Una de estas métricas agrupa automáticamente los datos por fecha (`successful-delivery-rate-grouped-by-date`). Otra métrica agrupa automáticamente los datos por ejecución de campaña (`successful-delivery-rate-grouped-by-campaign-activity`). Una tercera métrica simplemente devuelve un único valor: el porcentaje de mensajes que se entregaron a los destinatarios en todas las campañas realizadas (`successful-delivery-rate`).

Si no puede encontrar una métrica estándar que agrupe los datos de la forma que quiere, puede desarrollar una serie de consultas que devuelvan los datos que desee. A continuación, desglose o combine manualmente los resultados de la consulta en grupos personalizados de diseño propio.

Por último, es importante verificar que está autorizado para acceder a los datos que desea consultar. Para obtener más información, consulte Políticas de IAM para consultar datos de análisis de Amazon Pinpoint.

Consultar datos de análisis de Amazon Pinpoint para una campaña

Para consultar los datos de una campaña, utilice la API Métricas de campaña y especifique valores para los siguientes parámetros obligatorios:

- application-id: el ID del proyecto, que es el identificador único del proyecto asociado a la campaña. En Amazon Pinpoint, los términos proyecto y aplicación tienen el mismo significado.
- campaign-id: el identificador único de la campaña.
- kpi-name: el nombre de la métrica que se va a consultar. Este valor describe la métrica asociada y consta de dos o más términos, que se componen de caracteres alfanuméricos en minúsculas, separados por un guion. Para obtener una lista completa de las métricas admitidas y el valor `kpi-name` de cada una de ellas, consulte Métricas estándar para proyectos, campañas y recorridos.

También puede aplicar un filtro que consulte los datos de un intervalo de fechas determinado. Si no especifica un intervalo de fechas, Amazon Pinpoint devuelve los datos de los 31 días naturales anteriores. Para filtrar los datos por fechas diferentes, utilice los parámetros de intervalo de fechas admitidos para especificar la primera y la última fecha y hora del intervalo de fechas. Los valores deben estar en formato ISO 8601 extendido y utilizar la hora universal coordinada (UTC), por ejemplo, `2019-07-19T20:00:00Z` para el 19 de julio de 2019 a las 20:00 UTC. Los intervalos de

fechas son inclusivos y deben tener 31 días naturales o menos. Además, la primera fecha y hora deben ser inferiores a 90 días desde el día actual.

En los siguientes ejemplos se muestra cómo consultar los datos de análisis de una campaña mediante la API REST de Amazon Pinpoint AWS CLI, la y la. AWS SDK para Java Puede usar cualquier AWS SDK compatible para consultar los datos de análisis de una campaña. Los AWS CLI ejemplos están formateados para Microsoft Windows. Para Unix, Linux y macOS, reemplace el carácter de continuación de línea de intercalación (^) por una barra invertida (\).

REST API

Para consultar los datos de análisis de una campaña mediante la API de REST de Amazon Pinpoint, envíe una solicitud HTTP(S) GET al URI Métricas de campaña. En el URI, especifique los valores apropiados para los parámetros de ruta requeridos:

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

Donde:

- *endpoint*es el punto de enlace de Amazon Pinpoint para la AWS región que aloja el proyecto asociado a la campaña.
- *application-id* es el identificador único del proyecto asociado a la campaña.
- *campaign-id* es el identificador único de la campaña.
- *kpi-name*es el `kpi-name` valor de la métrica que se va a consultar.

Todos los parámetros deben tener codificación URL.

Para aplicar un filtro que consulta los datos de un intervalo de fechas específico, agregue los parámetros de consulta `start-time` y `end-time` y los valores al URI. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Utilice un signo ampersand (&) para separar los parámetros.

Por ejemplo, la siguiente solicitud recupera el número de puntos de enlace únicos a los que se entregaron los mensajes, en todas las ejecuciones de una campaña, desde el 19 de julio de 2019 hasta el 26 de julio de 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Donde:

- `pinpoint.us-east-1.amazonaws.com` es el punto de conexión de Amazon Pinpoint de la región de AWS que aloja el proyecto.
- `12345678901234567890123456example` es el identificador único del proyecto asociado a la campaña.
- `80b8efd84042ff8d9c96ce2f8example` es el identificador único de la campaña.
- `unique-deliveries` es el valor `kpi-name` de la métrica de campaña entregas a puntos de enlace que es la métrica que informa sobre el número de puntos de enlace únicos a los que se entregaron los mensajes en todas las ejecuciones de una campaña.
- `2019-07-19T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-07-26T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

AWS CLI

Para consultar los datos de análisis de una campaña mediante el AWS CLI, utilice el get-campaign-date-range-kpi comando y especifique los valores adecuados para los parámetros necesarios:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id application-id ^
    --campaign-id campaign-id ^
    --kpi-name kpi-name
```

Donde:

- *`application-id`* es el identificador único del proyecto asociado a la campaña.
- *`campaign-id`* es el identificador único de la campaña.
- *`kpi-name`*es el `kpi-name` valor de la métrica que se va a consultar.

Para aplicar un filtro que consulte los datos de un intervalo de fechas específico, agregue los parámetros `start-time` y `end-time` a la consulta. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Por ejemplo, la siguiente solicitud recupera el número de puntos de enlace únicos a los que se entregaron los mensajes, en todas las ejecuciones de una campaña, desde el 19 de julio de 2019 hasta el 26 de julio de 2019:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --campaign-id 80b8efd84042ff8d9c96ce2f8example ^
    --kpi-name unique-deliveries ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Donde:

- `12345678901234567890123456example` es el identificador único del proyecto asociado a la campaña.
- `80b8efd84042ff8d9c96ce2f8example` es el identificador único de la campaña.
- `unique-deliveries` es el valor `kpi-name` de la métrica de campaña entregas a puntos de enlace que es la métrica que informa sobre el número de puntos de enlace únicos a los que se entregaron los mensajes en todas las ejecuciones de una campaña.
- `2019-07-19T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-07-26T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

SDK for Java

Para consultar los datos de análisis de una campaña mediante el AWS SDK para Java, utilice el GetCampaignDateRangeKpiRequest método de la API Campaign Metrics. Especifique los valores apropiados para los parámetros requeridos:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withCampaignId("campaignId")
        .withKpiName("kpiName")
```

Donde:

- *applicationId* es el identificador único del proyecto asociado a la campaña.
- *campaignId* es el identificador único de la campaña.
- *kpiName*es el `kpi-name` valor de la métrica que se va a consultar.

Para aplicar un filtro que consulte los datos de un intervalo de fechas específico, incluya los parámetros `startTime` y `endTime` y los valores en la consulta. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Por ejemplo, la siguiente solicitud recupera el número de puntos de enlace únicos a los que se entregaron los mensajes, en todas las ejecuciones de una campaña, desde el 19 de julio de 2019 hasta el 26 de julio de 2019:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456789012345example")
        .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")));
```

Donde:

- `12345678901234567890123456789012345example` es el identificador único del proyecto asociado a la campaña.
- `80b8efd84042ff8d9c96ce2f8example` es el identificador único de la campaña.
- `unique-deliveries` es el valor `kpi-name` de la métrica de campaña entregas a puntos de enlace que es la métrica que informa sobre el número de puntos de enlace únicos a los que se entregaron los mensajes en todas las ejecuciones de una campaña.
- `2019-07-19T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-07-26T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

Después de enviar la consulta, Amazon Pinpoint devuelve los resultados de la consulta en una respuesta JSON. La estructura de los resultados varía en función de la métrica que haya consultado. Algunas métricas solo devuelven un valor. Por ejemplo, la métrica de la campaña entregas a puntos

de conexión (`unique-deliveries`), que se utiliza en los ejemplos anteriores, devuelve un valor: el número de puntos de conexión únicos a los que se entregaron los mensajes en todas las tiradas de una campaña. En este caso, la respuesta JSON es la siguiente:

```
{
    "CampaignDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "CampaignId":"80b8efd84042ff8d9c96ce2f8example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```

Otras métricas devuelven varios valores y agrupan los valores por un campo relevante. Si una métrica devuelve varios valores, la respuesta JSON incluye un campo que indica qué campo se utilizó para agrupar los datos.

Para obtener más información sobre la estructura de los resultados de la consulta, lea Usar resultados de consultas JSON.

Consultar datos de análisis de Amazon Pinpoint para varias campañas

Existen dos formas de consultar los datos de varias campañas. La mejor forma depende de si desea consultar los datos de las campañas asociadas al mismo proyecto. Si es así, también depende de si desea consultar los datos de todas las campañas o solo o subconjunto de ellas.

Para consultar los datos de las campañas asociadas a diferentes proyectos o solo de un subconjunto de campañas asociadas al mismo proyecto, lo mejor es crear y ejecutar una serie de consultas

individuales, una para cada campaña para la que desee consultar los datos. En la sección anterior se explica cómo consultar los datos de una sola campaña.

Para consultar los datos de todas las campañas asociadas al mismo proyecto, puede utilizar la API Métricas de aplicación. Especifique valores para los siguientes parámetros obligatorios:

- application-id: el ID del proyecto, que es el identificador único del proyecto. En Amazon Pinpoint, los términos proyecto y aplicación tienen el mismo significado.
- kpi-name: el nombre de la métrica que se va a consultar. Este valor describe la métrica asociada y consta de dos o más términos, que se componen de caracteres alfanuméricos en minúsculas, separados por un guion. Para obtener una lista completa de las métricas admitidas y el valor `kpi-name` de cada una de ellas, consulte Métricas estándar para proyectos, campañas y recorridos.

También puede filtrar los datos por intervalo de fechas. Si no especifica un intervalo de fechas, Amazon Pinpoint devuelve los datos de los 31 días naturales anteriores. Para filtrar los datos por fechas diferentes, utilice los parámetros de intervalo de fechas admitidos para especificar la primera y la última fecha y hora del intervalo de fechas. Los valores deben estar en formato ISO 8601 extendido y utilizar la hora universal coordinada (UTC), por ejemplo, `2019-07-19T20:00:00Z` para el 19 de julio de 2019 a las 20:00 UTC. Los intervalos de fechas son inclusivos y deben tener 31 días naturales o menos. Además, la primera fecha y hora deben ser inferiores a 90 días desde el día actual.

En los siguientes ejemplos se muestra cómo consultar los datos de análisis de una campaña mediante la API REST de Amazon Pinpoint AWS CLI, la y la. AWS SDK para Java Puede usar cualquier AWS SDK compatible para consultar los datos de análisis de una campaña. Los AWS CLI ejemplos están formateados para Microsoft Windows. Para Unix, Linux y macOS, reemplace el carácter de continuación de línea de intercalación (^) por una barra invertida (\).

REST API

Para consultar los datos de análisis de varias campañas mediante la API de REST de Amazon Pinpoint, envíe una solicitud HTTP(S) GET al URI Métricas de aplicación. En el URI, especifique los valores apropiados para los parámetros de ruta requeridos:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Donde:

- *endpoint*es el punto de enlace de Amazon Pinpoint para la AWS región que aloja el proyecto asociado a las campañas.
- *application-id* es el identificador único del proyecto asociado a las campañas.
- *kpi-name*es el `kpi-name` valor de la métrica que se va a consultar.

Todos los parámetros deben tener codificación URL.

Para aplicar un filtro que recupere los datos de un intervalo de fechas específico, agregue los parámetros de consulta `start-time` y `end-time` y los valores al URI. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Utilice un signo ampersand (&) para separar los parámetros.

Por ejemplo, la siguiente solicitud recupera el número de puntos de enlace únicos a los que se entregaron los mensajes, en todas las campañas de un proyecto, desde el 19 de julio de 2019 hasta el 26 de julio de 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456789012345example/
kpis/daterange/unique-deliveries-grouped-by-campaign?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Donde:

- `pinpoint.us-east-1.amazonaws.com` es el punto de conexión de Amazon Pinpoint de la región de AWS que aloja el proyecto.
- `12345678901234567890123456789012345example` es el identificador único del proyecto asociado a las campañas.
- `unique-deliveries-grouped-by-campaign` es el valor `kpi-name` de la métrica de aplicación entregas a puntos de enlace, agrupadas por campaña que es la métrica que devuelve el número de puntos de enlace únicos a los que se entregaron los mensajes por campaña.
- `2019-07-19T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-07-26T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

AWS CLI

Para consultar los datos de análisis de varias campañas mediante el AWS CLI, utilice el get-application-date-range-kpi comando y especifique los valores adecuados para los parámetros necesarios:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Donde:

- *application-id* es el identificador único del proyecto asociado a las campañas.
- *kpi-name*es el `kpi-name` valor de la métrica que se va a consultar.

Para aplicar un filtro que recupere los datos de un intervalo de fechas específico, incluya los parámetros `start-time` y `end-time` y los valores en la consulta. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Por ejemplo, la siguiente solicitud recupera el número de puntos de enlace únicos a los que se entregaron los mensajes, en todas las campañas de un proyecto, desde el 19 de julio de 2019 hasta el 26 de julio de 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name unique-deliveries-grouped-by-campaign ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Donde:

- `12345678901234567890123456example` es el identificador único del proyecto asociado a la campaña.
- `unique-deliveries-grouped-by-campaign` es el valor `kpi-name` de la métrica de aplicación entregas a puntos de enlace, agrupadas por campaña que es la métrica que devuelve el número de puntos de enlace únicos a los que se entregaron los mensajes por campaña.
- `2019-07-19T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

- `2019-07-26T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

SDK for Java

Para consultar los datos de análisis de varias campañas mediante la AWS SDK para Java, utilice el GetApplicationDateRangeKpiRequest método de la API Application Metrics. Especifique los valores apropiados para los parámetros requeridos:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Donde:

- *`applicationId`* es el identificador único del proyecto asociado a las campañas.
- *`kpiName`*es el `kpi-name` valor de la métrica que se va a consultar.

Para aplicar un filtro que recupere los datos de un intervalo de fechas específico, incluya los parámetros `startTime` y `endTime` y los valores en la consulta. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Por ejemplo, la siguiente solicitud recupera el número de puntos de enlace únicos a los que se entregaron los mensajes, en todas las campañas de un proyecto, desde el 19 de julio de 2019 hasta el 26 de julio de 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123example")
        .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")));
```

Donde:

- `12345678901234567890123example` es el identificador único del proyecto asociado a las campañas.
- `unique-deliveries-grouped-by-campaign` es el valor `kpi-name` de la métrica de aplicación entregas a puntos de enlace, agrupadas por campaña que es la métrica que

devuelve el número de puntos de enlace únicos a los que se entregaron los mensajes por campaña.

- `2019-07-19T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-07-26T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

Después de enviar la consulta, Amazon Pinpoint devuelve los resultados de la consulta en una respuesta JSON. La estructura de los resultados varía en función de la métrica que haya consultado. Algunas métricas solo devuelven un valor. Otras métricas devuelven varios valores y esos valores se agrupan por campo relevante. Si una métrica devuelve varios valores, la respuesta JSON incluye un campo que indica qué campo se utilizó para agrupar los datos.

Por ejemplo, la métrica de la aplicación entregas de puntos de conexión, agrupadas por campaña (`unique-deliveries-grouped-by-campaign`), que se utiliza en los ejemplos anteriores, devuelve valores múltiples: el número de puntos de conexión únicos a los que se entregaron los mensajes, para cada campaña asociada a un proyecto. En este caso, la respuesta JSON es la siguiente:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
```

```
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```

En este caso, el campo `GroupedBys` indica que los valores están agrupados por ID de campaña (`CampaignId`).

Para obtener más información sobre la estructura de los resultados de la consulta, lea Usar resultados de consultas JSON.

Consultar datos de análisis de Amazon Pinpoint para mensajes transaccionales

Además de utilizar las páginas de análisis de la consola Amazon Pinpoint, puede utilizar Amazon Pinpoint APIs Analytics para consultar los datos de análisis de un subconjunto de métricas estándar que proporcionan información sobre las tendencias de entrega e interacción de los mensajes transaccionales que se enviaron para un proyecto.

Cada una de estas métricas es un valor medible, también conocido como un indicador clave de rendimiento (KPI), que puede ayudarle a supervisar y evaluar el rendimiento de los mensajes transaccionales. Por ejemplo, puede usar una métrica para averiguar cuántos mensajes de correo electrónico o SMS transaccionales envió o cuántos de esos mensajes se entregaron a los destinatarios. Amazon Pinpoint recopila y agrega automáticamente estos datos para todos los correos electrónicos y mensajes SMS transaccionales que envíe para un proyecto. Almacena los datos durante 90 días.

Si utiliza Amazon Pinpoint Analytics APIs para consultar datos, puede elegir varias opciones que definan el alcance, los datos, la agrupación y los filtros de la consulta. Para ello, utilice parámetros que especifiquen el proyecto y la métrica que desea consultar, además de los filtros basados en fechas que desee aplicar.

En este tema se explica y se proporcionan ejemplos de cómo elegir estas opciones y consultar los datos de mensajes transaccionales de un proyecto.

Requisitos previos

Antes de consultar los datos de análisis de los mensajes transaccionales, ayuda a recopilar la siguiente información, que utilizará para definir la consulta:

- ID del proyecto: el identificador único del proyecto desde el que se enviaron los mensajes. En la API de Amazon Pinpoint, este valor se almacena en la propiedad `application-id`. En la consola de Amazon Pinpoint, este valor se muestra como ID de proyecto en la página Todos los proyectos.
- Intervalo de fechas: de forma opcional, la primera y la última fecha y hora del intervalo de fechas para el que se van a consultar los datos. Los intervalos de fechas son inclusivos y deben tener

31 días naturales o menos. Además, deben comenzar en menos de 90 días a partir del día actual. Si no especifica un intervalo de fechas, Amazon Pinpoint consulta automáticamente los datos de los 31 días naturales anteriores.

- Métrica: el nombre de la métrica que se va a consultar y, más específicamente, el valor `kpi-name` de la métrica. Para obtener una lista completa de las métricas admitidas y el valor `kpi-name` de cada una de ellas, consulte Métricas estándar para proyectos, campañas y recorridos.

También ayuda determinar si desea agrupar los datos por un campo relevante. Si lo hace, para simplificar el análisis y los informes, elija una métrica diseñada para agrupar datos de forma automática. Por ejemplo, Amazon Pinpoint proporciona varias métricas estándar que informan del número de mensajes SMS transaccionales que se entregaron a los destinatarios. Una de estas métricas agrupa automáticamente los datos por fecha (`txn-sms-delivered-grouped-by-date`). Otra métrica agrupa automáticamente los datos por país o región (`txn-sms-delivered-grouped-by-country`). Una tercera métrica simplemente devuelve un único valor: el número de mensajes que se entregaron a los destinatarios (`txn-sms-delivered`). Si no puede encontrar una métrica estándar que agrupe los datos de la forma que quiere, puede desarrollar una serie de consultas que devuelvan los datos que desee. A continuación, desglose o combine manualmente los resultados de la consulta en grupos personalizados de diseño propio.

Por último, es importante verificar que está autorizado para acceder a los datos que desea consultar. Para obtener más información, consulte Políticas de IAM para consultar datos de análisis de Amazon Pinpoint.

Consultar datos de Amazon Pinpoint para mensajes de correo electrónico transaccionales

Para consultar los datos de los mensajes de correo electrónico transaccionales enviados para un proyecto, utilice la API Métricas de aplicación y especifique valores para los siguientes parámetros obligatorios:

- application-id: el ID del proyecto, que es el identificador único del proyecto. En Amazon Pinpoint, los términos proyecto y aplicación tienen el mismo significado.
- kpi-name: el nombre de la métrica que se va a consultar. Este valor describe la métrica asociada y consta de dos o más términos, que se componen de caracteres alfanuméricos en minúsculas, separados por un guion. Para obtener una lista completa de las métricas admitidas y el valor `kpi-name` de cada una de ellas, consulte Métricas estándar para proyectos, campañas y recorridos.

También puede aplicar un filtro que consulte los datos de un intervalo de fechas determinado. Si no especifica un intervalo de fechas, Amazon Pinpoint devuelve los datos de los 31 días naturales anteriores. Para filtrar los datos por fechas diferentes, utilice los parámetros de intervalo de fechas admitidos para especificar la primera y la última fecha y hora del intervalo de fechas. Los valores deben estar en formato ISO 8601 extendido y utilizar la hora universal coordinada (UTC), por ejemplo, `2019-09-06T20:00:00Z` para el 6 de septiembre de 2019 a las 20:00 UTC. Los intervalos de fechas son inclusivos y deben tener 31 días naturales o menos. Además, la primera fecha y hora deben ser inferiores a 90 días desde el día actual.

En los siguientes ejemplos se muestra cómo consultar los datos de análisis de los mensajes de correo electrónico transaccionales mediante la API REST de Amazon Pinpoint, AWS CLI la y la. AWS SDK para Java Puede utilizar cualquier SDK de AWS compatible para consultar los datos de análisis de mensajes transaccionales. Los AWS CLI ejemplos están formateados para Microsoft Windows. Para Unix, Linux y macOS, reemplace el carácter de continuación de línea de intercalación (^) por una barra invertida (\).

REST API

Para consultar los datos de análisis de mensajes de correo electrónico transaccionales mediante la API de REST de Amazon Pinpoint, envíe una solicitud HTTP(S) GET al URI Métricas de aplicación. En el URI, especifique los valores apropiados para los parámetros de ruta requeridos:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Donde:

- *`endpoint`*es el punto de enlace de Amazon Pinpoint para la AWS región que aloja el proyecto.
- *`application-id`* es el identificador único del producto.
- *`kpi-name`*es el `kpi-name` valor de la métrica que se va a consultar.

Todos los parámetros deben tener codificación URL.

Para aplicar un filtro que consulta los datos de un intervalo de fechas específico, agregue los parámetros de consulta `start-time` y `end-time` y los valores al URI. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Utilice un signo ampersand (&) para separar los parámetros.

Por ejemplo, la siguiente solicitud recupera el número de mensajes de correo electrónico transaccionales enviados para un proyecto desde el 6 de septiembre de 2019 hasta el 13 de septiembre de 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456789012345example/
kpis/daterange/txn-emails-sent?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-13T23:59:59Z
```

Donde:

- `pinpoint.us-east-1.amazonaws.com` es el punto de conexión de Amazon Pinpoint de la región de AWS que aloja el proyecto.
- `12345678901234567890123456789012345example` es el identificador único del producto.
- `txn-emails-sent` es el valor `kpi-name` de la métrica de aplicación envíos que es la métrica que informa sobre el número de mensajes de correo electrónico transaccionales enviados para un proyecto.
- `2019-09-06T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-09-13T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

AWS CLI

Para consultar los datos de análisis de los mensajes de correo electrónico transaccionales mediante el AWS CLI, utilice el get-application-date-range-kpi comando y especifique los valores adecuados para los parámetros necesarios:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Donde:

- *`application-id`* es el identificador único del producto.
- *`kpi-name`*es el `kpi-name` valor de la métrica que se va a consultar.

Para aplicar un filtro que consulte los datos de un intervalo de fechas específico, agregue los parámetros `start-time` y `end-time` a la consulta. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Por ejemplo, la siguiente solicitud recupera el número de mensajes de correo electrónico transaccionales enviados para un proyecto desde el 6 de septiembre de 2019 hasta el 13 de septiembre de 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-emails-sent ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-13T23:59:59Z
```

Donde:

- `12345678901234567890123456example` es el identificador único del producto.
- `txn-emails-sent` es el valor `kpi-name` de la métrica de aplicación envíos que es la métrica que informa sobre el número de mensajes de correo electrónico transaccionales enviados para un proyecto.
- `2019-09-06T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-09-13T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

SDK for Java

Para consultar los datos de análisis de los mensajes de correo electrónico transaccionales mediante el AWS SDK para JavaGetApplicationDateRangeKpiRequest método de la API Application Metrics. Especifique los valores apropiados para los parámetros requeridos:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Donde:

- *`applicationId`* es el identificador único del producto.
- *`kpiName`*es el `kpi-name` valor de la métrica que se va a consultar.

Para aplicar un filtro que consulte los datos de un intervalo de fechas específico, incluya los parámetros `startTime` y `endTime` y los valores en la consulta. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Por ejemplo, la siguiente solicitud recupera el número de mensajes de correo electrónico transaccionales enviados para un proyecto desde el 6 de septiembre de 2019 hasta el 13 de septiembre de 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-emails-sent")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

Donde:

- `12345678901234567890123456example` es el identificador único del producto.
- `txn-emails-sent` es el valor `kpi-name` de la métrica de aplicación envíos que es la métrica que informa sobre el número de mensajes de correo electrónico transaccionales enviados para un proyecto.
- `2019-09-06T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-09-13T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

Después de enviar la consulta, Amazon Pinpoint devuelve los resultados de la consulta en una respuesta JSON. La estructura de los resultados varía en función de la métrica que haya consultado. Algunas métricas solo devuelven un valor. Por ejemplo, la métrica de la aplicación envíos (`txn-emails-sent`), que se utiliza en los ejemplos anteriores, devuelve un valor: el número de mensajes de correo electrónico transaccionales que se enviaron desde un proyecto. En este caso, la respuesta JSON es la siguiente:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-13T23:59:59Z",
        "KpiName":"txn-emails-sent",
        "KpiResult":{
            "Rows":[
```

```
                {
                    "Values":[
                        {
                            "Key":"TxnEmailsSent",
                            "Type":"Double",
                            "Value":"62.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

Otras métricas devuelven varios valores y agrupan los valores por un campo relevante. Si una métrica devuelve varios valores, la respuesta JSON incluye un campo que indica qué campo se utilizó para agrupar los datos.

Para obtener más información sobre la estructura de los resultados de la consulta, lea Usar resultados de consultas JSON.

Consultar datos de Amazon Pinpoint para mensajes SMS transaccionales

Para consultar los datos de los mensajes SMS transaccionales enviados para un proyecto, utilice la API Métricas de aplicación y especifique valores para los siguientes parámetros obligatorios:

- application-id: el ID del proyecto, que es el identificador único del proyecto. En Amazon Pinpoint, los términos proyecto y aplicación tienen el mismo significado.
- kpi-name: el nombre de la métrica que se va a consultar. Este valor describe la métrica asociada y consta de dos o más términos, que se componen de caracteres alfanuméricos en minúsculas, separados por un guion. Para obtener una lista completa de las métricas admitidas y el valor `kpi-name` de cada una de ellas, consulte Métricas estándar para proyectos, campañas y recorridos.

También puede aplicar un filtro que consulte los datos de un intervalo de fechas determinado. Si no especifica un intervalo de fechas, Amazon Pinpoint devuelve los datos de los 31 días naturales anteriores. Para filtrar los datos por fechas diferentes, utilice los parámetros de intervalo de fechas admitidos para especificar la primera fecha y hora y la última fecha y hora del intervalo de fechas. Los valores deben estar en formato ISO 8601 extendido y utilizar la hora universal coordinada (UTC), por ejemplo, `2019-09-06T20:00:00Z` para el 6 de septiembre de 2019 a las 20:00 UTC. Los

intervalos de fechas son inclusivos y deben tener 31 días naturales o menos. Además, la primera fecha y hora deben ser inferiores a 90 días desde el día actual.

Los siguientes ejemplos muestran cómo consultar datos analíticos para mensajes SMS transaccionales mediante la API REST de Amazon Pinpoint, AWS CLI la y la. AWS SDK para Java Puede usar cualquier AWS SDK compatible para consultar los datos de análisis de los mensajes transaccionales. Los AWS CLI ejemplos están formateados para Microsoft Windows. Para Unix, Linux y macOS, reemplace el carácter de continuación de línea de intercalación (^) por una barra invertida (\).

REST API

Para consultar los datos de análisis de mensajes SMS transaccionales mediante la API de REST de Amazon Pinpoint, envíe una solicitud HTTP(S) GET al URI Métricas de aplicación. En el URI, especifique los valores apropiados para los parámetros de ruta requeridos:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Donde:

- *endpoint*es el punto de enlace de Amazon Pinpoint para la AWS región que aloja el proyecto.
- *application-id* es el identificador único del producto.
- *kpi-name*es el `kpi-name` valor de la métrica que se va a consultar.

Todos los parámetros deben tener codificación URL.

Para aplicar un filtro que recupere los datos de un intervalo de fechas específico, agregue los parámetros de consulta `start-time` y `end-time` y los valores al URI. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Utilice un signo ampersand (&) para separar los parámetros.

Por ejemplo, la siguiente solicitud recupera el número de mensajes SMS transaccionales enviados cada día desde el 6 de septiembre de 2019 hasta el 8 de septiembre de 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-08T23:59:59Z
```

Donde:

- `pinpoint.us-east-1.amazonaws.com` es el punto de conexión de Amazon Pinpoint de la región de AWS que aloja el proyecto.
- `12345678901234567890123example` es el identificador único del producto.
- `txn-sms-sent-grouped-by-date` es el valor `kpi-name` de los envíos, agrupados por fecha que es la métrica que devuelve el número de mensajes SMS transaccionales enviados cada día durante el intervalo de fechas.
- `2019-09-06T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-09-08T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

AWS CLI

Para consultar los datos analíticos de los mensajes SMS transaccionales mediante el AWS CLI, utilice el get-application-date-range-kpi comando y especifique los valores adecuados para los parámetros necesarios:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Donde:

- *`application-id`* es el identificador único del producto.
- *`kpi-name`*es el `kpi-name` valor de la métrica que se va a consultar.

Para aplicar un filtro que recupere los datos de un intervalo de fechas específico, incluya los parámetros `start-time` y `end-time` y los valores en la consulta. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Por ejemplo, la siguiente solicitud recupera el número de mensajes SMS transaccionales enviados cada día desde el 6 de septiembre de 2019 hasta el 8 de septiembre de 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
```

```
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-sms-sent-grouped-by-date ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-08T23:59:59Z
```

Donde:

- `12345678901234567890123456example` es el identificador único del producto.
- `txn-sms-sent-grouped-by-date` es el valor `kpi-name` de los envíos, agrupados por fecha que es la métrica que devuelve el número de mensajes SMS transaccionales enviados cada día durante el intervalo de fechas.
- `2019-09-06T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-09-08T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

SDK for Java

Para consultar los datos analíticos de los mensajes SMS transaccionales mediante el AWS SDK para Java, utilice el GetApplicationDateRangeKpiRequest método de la API Application Metrics y especifique los valores adecuados para los parámetros necesarios:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Donde:

- *`applicationId`* es el identificador único del producto.
- *`kpiName`*es el `kpi-name` valor de la métrica que se va a consultar.

Para aplicar un filtro que recupere los datos de un intervalo de fechas específico, incluya los parámetros `startTime` y `endTime` y los valores en la consulta. Estos parámetros permiten especificar la primera y última fecha y hora, en formato ISO 8601 ampliado, de un intervalo de fechas inclusivo para recuperar los datos. Por ejemplo, la siguiente solicitud recupera el número de mensajes SMS transaccionales enviados cada día desde el 6 de septiembre de 2019 hasta el 8 de septiembre de 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456789012345example")
        .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")));
```

Donde:

- `12345678901234567890123456789012345example` es el identificador único del producto.
- `txn-sms-sent-grouped-by-date` es el valor `kpi-name` de los envíos, agrupados por fecha que es la métrica que devuelve el número de mensajes SMS transaccionales enviados cada día durante el intervalo de fechas.
- `2019-09-06T00:00:00Z` es la primera fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.
- `2019-09-08T23:59:59Z` es la última fecha y hora para recuperar datos dentro de un intervalo de fechas inclusivo.

Después de enviar la consulta, Amazon Pinpoint devuelve los resultados de la consulta en una respuesta JSON. La estructura de los resultados varía en función de la métrica que haya consultado. Algunas métricas solo devuelven un valor. Otras métricas devuelven varios valores y agrupan estos valores por un campo relevante. Si una métrica devuelve varios valores, la respuesta JSON incluye un campo que indica qué campo se utilizó para agrupar los datos.

Por ejemplo, la métrica de la aplicación envíos, agrupados por fecha (`txn-sms-sent-grouped-by-date`), que se utiliza en los ejemplos anteriores, devuelve varios valores: el número de mensajes SMS transaccionales que se enviaron durante cada día del intervalo de fechas especificado. En este caso, la respuesta JSON es la siguiente:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456789012345example",
        "EndTime":"2019-09-08T23:59:59Z",
        "KpiName":"txn-sms-sent-grouped-by-date",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
```

```
                        "Type":"String",
                        "Value":"2019-09-06"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"29.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-07"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"35.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-08"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"10.0"
                    }
                ]
            }
```

```
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

En este caso, el campo `GroupedBys` indica que los valores se agrupan por día natural (`Date`). Esto significa que:

- El 6 de septiembre de 2019 se enviaron 29 mensajes.
- El 7 de septiembre de 2019 se enviaron 35 mensajes.
- El 8 de septiembre de 2019 se enviaron 10 mensajes.

Para obtener más información sobre la estructura de los resultados de la consulta, lea Usar resultados de consultas JSON.

Usar resultados de consultas JSON de análisis de Amazon Pinpoint

Cuando utiliza Amazon Pinpoint Analytics APIs para consultar datos de análisis, Amazon Pinpoint devuelve los resultados en una respuesta JSON. Para las métricas de aplicación, las métricas de campaña y las métricas de interacción del recorrido, los datos de la respuesta siguen un esquema JSON estándar a la hora de generar informes de datos de análisis de Amazon Pinpoint.

Esto significa que puede utilizar el lenguaje de programación o la herramienta de su elección para implementar una solución personalizada que consulte los datos de una o más de estas métricas, capture los resultados de cada consulta y, a continuación, escriba los resultados en una tabla, en un objeto o en otra ubicación. Luego podrá trabajar con los resultados de la consulta en esa ubicación con otro servicio o aplicación.

Por ejemplo, puede hacer lo siguiente:

- Crear un panel personalizado que consulte regularmente un conjunto de métricas y muestre los resultados en el marco de visualización de datos preferido.
- Crear un informe que realice un seguimiento de las tasas de interacción a partir de las métricas adecuadas y mostrar los resultados en un gráfico u otro tipo de informe que haya diseñado.

- Analizar y escribir datos de análisis en un formato de almacenamiento determinado y, a continuación, trasladar los resultados a una solución de almacenamiento a largo plazo.

Tenga en cuenta que Amazon Pinpoint Analytics APIs no está diseñado para crear ni almacenar ningún objeto persistente que pueda leer o utilizar posteriormente en un proyecto de Amazon Pinpoint o en su cuenta de Amazon Pinpoint. Por el contrario, APIs están diseñados para ayudarle a recuperar datos analíticos y transferirlos a otros servicios y aplicaciones para su posterior análisis, almacenamiento o elaboración de informes. Utilizan en parte la misma estructura de respuesta y esquema JSON para todos los datos de análisis que se puede consultar mediante programación para métricas de aplicación, métricas de campaña y métricas de interacción del recorrido.

En este tema se explica la estructura, los objetos y los campos de una respuesta JSON a una consulta para métricas de aplicación, métricas de campaña y métricas de interacción del recorrido. Para obtener información sobre los campos de una respuesta JSON a una consulta para una métrica de ejecución de recorrido o una métrica de ejecución de actividad de recorrido, consulte Métricas estándar que se aplican a los proyectos, campañas y recorridos de Amazon Pinpoint.

Estructura JSON

Para ayudarle a analizar y utilizar los resultados de las consultas, Amazon Pinpoint Analytics APIs utiliza la misma estructura de respuesta JSON para todos los datos de Amazon Pinpoint Analytics, que puede consultar mediante programación para obtener métricas de aplicaciones, métricas de campañas y métricas de interacción con el recorrido. Cada respuesta JSON especifica los valores que definieron la consulta, como el ID del proyecto (`ApplicationId`). La respuesta también incluye un (solo) objeto `KpiResult`. El objeto `KpiResult` contiene el conjunto de resultados generales de una consulta.

Cada objeto `KpiResult` contiene un objeto `Rows`. Se trata de una matriz de objetos que contiene resultados de consulta y los metadatos relevantes sobre los valores de esos resultados. La estructura y el contenido de un objeto `Rows` tienen las siguientes características generales:

- Cada fila de resultados de la consulta es un objeto JSON separado, denominado `Values`, en el objeto `Rows`. Por ejemplo, si una consulta devuelve tres valores, el objeto `Rows` contiene tres objetos `Values`. Cada objeto `Values` contiene un resultado individual para la consulta.
- Cada columna de los resultados de consulta es una propiedad del objeto `Values` al que se aplica. El nombre de la columna se almacena en el campo `Key` del objeto `Values`.

- Para los resultados de consultas agrupadas, cada objeto `Values` tiene un objeto `GroupedBys` asociado. El objeto `GroupedBys` indica qué campo se utilizó para agrupar los resultados. También proporciona el valor de agrupación para el objeto `Values` asociado.
- Si los resultados de la consulta de una métrica son nulos, el objeto `Rows` estará vacío.

Además de estas características generales, la estructura y el contenido del objeto `Rows` varía según la métrica. Esto se debe a que Amazon Pinpoint admite dos tipos de métricas: las métricas de valor único y las métricas de varios valores.

Una métrica de valor único solo proporciona un valor acumulativo. Un ejemplo es el porcentaje de mensajes entregados a los destinatarios por todas las ejecuciones de una campaña. Una métrica de varios valores proporciona más de un valor y los agrupa por campo relevante. Un ejemplo es el porcentaje de mensajes entregados a los destinatarios en cada ejecución de una campaña, agrupados por ejecución de campaña.

Para determinar rápidamente si una métrica es de valor único o de varios valores observe el nombre de la métrica. Si el nombre no contiene `grouped-by`, se tratará de una métrica de valor único. Si lo contiene, será una métrica de varios valores. Para obtener una lista completa de las métricas que puede consultar mediante programación, visite Métricas estándar que se aplican a los proyectos, campañas y recorridos de Amazon Pinpoint.

Métricas de valor único

En una métrica de valor único, el objeto `Rows` contiene un objeto `Values` que:

- Especifica el nombre descriptivo de la métrica que se consultó.
- Proporciona el valor de la métrica que se consultó.
- Identifica el tipo de datos del valor devuelto.

Por ejemplo, la siguiente respuesta JSON contiene los resultados de la consulta para una métrica de valor único. Esta métrica comunica el número de puntos de enlace únicos a los que se entregaron mensajes por todas las campañas asociadas a un proyecto, desde el 1 de agosto de 2019 hasta el 31 de agosto de 2019:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456789012345example",
        "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"
    }
}
```

En este ejemplo, la respuesta indica que todas las campañas del proyecto enviaron mensajes a 1368 puntos de enlace únicos desde el 1 de agosto de 2019 hasta el 31 de agosto de 2019, donde:

- `Key` es el nombre descriptivo de la métrica cuyo valor se especifica en el campo `Value` (`UniqueDeliveries`).
- `Type` es el tipo de datos del valor especificado en el campo `Value` (`Double`).
- `Value` es el valor real de la métrica que se consultó, incluidos los filtros que se aplicaron (`1368.0`).

Si los resultados de la consulta para una métrica de un solo valor son nulos (no mayor o igual a cero), el objeto de `Rows` está vacío. Amazon Pinpoint devuelve un valor nulo para una métrica si no hay ningún dato que devolver para la métrica. Por ejemplo:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"234567890123456789012345 6example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[

            ]
        },
```

```
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Métricas de varios valores

La estructura y el contenido del objeto `Rows` para una métrica de varios valores suelen ser las mismos que para una métrica de valor único. El objeto `Rows` de una métrica de valores múltiples también contiene un objeto `Values`. El objeto `Values` especifica el nombre descriptivo de la métrica consultada, proporciona el valor de esa métrica e identifica el tipo de datos de ese valor.

Sin embargo, el objeto `Rows` de una métrica de valores múltiples también contiene uno o más objetos `GroupedBy`. Hay un objeto `GroupedBy` para cada objeto `Values` en los resultados de la consulta. El objeto `GroupedBy` indica qué campo se utilizó para agrupar los datos en los resultados y el tipo de datos de ese campo. También indica el valor de agrupación para ese campo (para el objeto `Values` asociado).

Por ejemplo, la siguiente respuesta JSON contiene los resultados de la consulta de una métrica de varios valores que informa sobre el número de puntos de enlace únicos a los que se entregaron mensajes en cada campaña asociada a un proyecto desde el 1 de agosto de 2019 hasta el 31 de agosto de 2019:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
```

```
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

En este ejemplo, la respuesta indica que tres de las campañas del proyecto entregaron mensajes a puntos de enlace únicos desde el 1 de agosto de 2019 hasta el 31 de agosto de 2019. Para cada una de estas campañas, el desglose de los recuentos de entregas es el siguiente:

- La campaña `80b8efd84042ff8d9c96ce2f8example` entregó mensajes a 123 puntos de enlace únicos.
- La campaña `810c7aab86d42fb2b56c8c966example` entregó mensajes a 456 puntos de enlace únicos.
- La campaña `42d8c7eb0990a57ba1d5476a3example` entregó mensajes a 789 puntos de enlace únicos.

Donde la estructura general de los objetos y campos es la siguiente:

- `GroupedBys.Key`: el nombre de la propiedad o el campo que almacena el valor de agrupamiento especificado en el campo `GroupedBys.Value` (`CampaignId`).
- `GroupedBys.Type`: el tipo de datos del valor especificado en el campo `GroupedBys.Value` (`String`).
- `GroupedBys.Value`: el valor real del campo que se utilizó para agrupar los datos, tal como se especifica en el campo `GroupedBys.Key` (ID de campaña).
- `Values.Key`: es el nombre descriptivo de la métrica cuyo valor se especifica en el campo `Values.Value` (`UniqueDeliveries`).
- `Values.Type`: el tipo de datos del valor especificado en el campo `Values.Value` (`Double`).
- `Values.Value`: el valor real de la métrica que se consultó, incluidos los filtros que se aplicaron.

Si los resultados de la consulta de una métrica de varios valores son nulos (ni mayor ni igual a cero) para un proyecto, campaña u otro recurso específico, Amazon Pinpoint no devuelve ningún objeto ni campo para el recurso. Si los resultados de la consulta de una métrica de varios valores son nulos para todos los recursos, Amazon Pinpoint devuelve un objeto `Rows` vacío.

Objetos y campos JSON

Además de especificar los valores que definieron una consulta, como el ID del proyecto (`ApplicationId`), cada respuesta JSON a una consulta para métricas de aplicación, métricas de campaña o métricas de interacción del recorrido incluye un objeto `KpiResult`. Este objeto contiene el conjunto de resultados generales de una consulta, que puede analizar para enviar los datos de

análisis a otro servicio o aplicación. Cada objeto `KpiResult` contiene algunos o todos los objetos y campos estándar siguientes, dependiendo de la métrica.

Objeto o campo	Description (Descripción)
`Rows`	Matriz de objetos que contiene el conjunto de resultados de una consulta.
`Rows.GroupedBys`	En una métrica de varios valores, es una matriz de campos que define el campo y los valores utilizados para agrupar datos en los resultados de la consulta.
`Rows.GroupedBys.Key`	En una métrica de varios valores, es el nombre de la propiedad o del campo que almacena el valor especificado en el campo `GroupedBys.Value` .
`Rows.GroupedBys.Type`	En una métrica de varios valores, es el tipo de dato del valor especificado en el campo `GroupedBys.Value` .
`Rows.GroupedBys.Value`	En una métrica de varios valores, es el valor real del campo que se utilizó para agrupar datos en los resultados de la consulta. Este valor se correlaciona con un objeto `Values` asociado.
`Rows.Values`	Matriz de campos que contiene los resultados de la consulta.
`Rows.Values.Key`	El nombre descriptivo de la métrica que se consultó. El valor de la métrica se especifica en el campo `Values.Value` .
`Rows.Values.Type`	Tipo de datos del valor especificado en el campo `Values.Value` .

Objeto o campo	Description (Descripción)
`Rows.Values.Value`	Valor real de la métrica que se consultó, incluidos los filtros que se aplicaron.

Para obtener información sobre los campos de una respuesta JSON a una consulta para una métrica de ejecución de recorrido o una métrica de ejecución de actividad de recorrido, consulte Métricas estándar que se aplican a los proyectos, campañas y recorridos de Amazon Pinpoint.

Registre las llamadas a la API Amazon Pinpoint con AWS CloudTrail

Amazon Pinpoint está integrado con AWS CloudTrail, que es un servicio que proporciona un registro de las acciones realizadas por un usuario, función o AWS servicio en Amazon Pinpoint. CloudTrail captura las llamadas a la API de Amazon Pinpoint como eventos. Las llamadas capturadas incluyen las llamadas realizadas desde la consola de Amazon Pinpoint y las llamadas de código a las operaciones de la API de Amazon Pinpoint.

Si crea un registro, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon Simple Storage Service (Amazon S3), incluidos los eventos de Amazon Pinpoint. Si no configura una ruta, podrá ver los eventos más recientes mediante el historial de eventos de la CloudTrail consola. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon Pinpoint, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, incluido cómo configurarlo y habilitarlo, consulte la Guía del AWS CloudTrail usuario.

Información de Amazon Pinpoint en CloudTrail

CloudTrail está habilitada en su AWS cuenta al crear la cuenta. Cuando se produce una actividad de eventos admitida en Amazon Pinpoint, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los eventos recientes en su AWS cuenta. Para obtener más información, consulta Cómo ver eventos con el historial de CloudTrail eventos.

Para obtener un registro continuo de los eventos de su AWS cuenta, incluidos los eventos de Amazon Pinpoint, cree una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando crea una ruta en la consola, la ruta se aplica a todas AWS las regiones. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento

- CloudTrail servicios e integraciones compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad lo ayuda a determinar:

- Si la solicitud se realizó con credenciales de AWS Identity and Access Management usuario o raíz.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte Elemento userIdentity de CloudTrail .

Puede crear un registro de seguimiento y almacenar los archivos de registro en el bucket de Amazon S3 durante el tiempo que desee. Además, puede definir reglas de ciclo de vida de Amazon S3 para archivar o eliminar archivos de registros de forma automática. De forma predeterminada, los archivos de registro se cifran con el cifrado del servidor (SSE) de Amazon S3.

Para recibir notificaciones de la entrega de archivos de registro, CloudTrail configúrelo para publicar notificaciones de Amazon SNS cuando se entreguen nuevos archivos de registro. Para obtener más información, consulte Configuración de notificaciones de Amazon SNS para CloudTrail.

También puede agrupar archivos de registro de Amazon Pinpoint de varias AWS regiones y AWS cuentas en un único bucket de Amazon S3. Para obtener más información, consulte Recibir archivos de registro de CloudTrail de varias regiones y Recibir archivos de registro de CloudTrail de varias cuentas.

Puede utilizarlo CloudTrail para registrar las acciones de los siguientes Amazon Pinpoint APIs:

- API de Amazon Pinpoint
- API de SMS y voz de Amazon Pinpoint

Acciones de la API Amazon Pinpoint compatibles en CloudTrail los archivos de registro

La API Amazon Pinpoint admite el registro de las siguientes acciones como eventos en los archivos de CloudTrail registro:

- CreateApp
- CreateCampaign
- CreateEmailTemplate
- CreateExportJob
- CreateImportJob
- CreateJourney
- CreatePushTemplate
- CreateRecommenderConfiguration
- CreateSegment
- CreateSmsTemplate
- CreateVoiceTemplate
- DeleteAdmChannel
- DeleteApnsChannel
- DeleteApnsSandboxChannel
- DeleteApnsVoipChannel
- DeleteApnsVoipSandboxChannel
- DeleteApp
- DeleteBaiduChannel
- DeleteCampaign
- DeleteEmailChannel
- DeleteEmailTemplate
- DeleteEndpoint
- DeleteEventStream
- DeleteGcmChannel

- DeleteJourney
- DeletePushTemplate
- DeleteRecommenderConfiguration
- DeleteSegment
- DeleteSmsChannel
- DeleteSmsTemplate
- DeleteUserEndpoints
- DeleteVoiceChannel
- DeleteVoiceTemplate
- GetAdmChannel
- GetApnsChannel
- GetApnsSandboxChannel
- GetApnsVoipChannel
- GetApnsVoipSandboxChannel
- GetApp
- GetApplicationDateRangeKpi
- GetApplicationSettings
- GetApps
- GetBaiduChannel
- GetCampaign
- GetCampaignActivities
- GetCampaignDateRangeKpi
- GetCampaignVersion
- GetCampaignVersions
- GetCampaigns
- GetChannels
- GetEmailChannel
- GetEmailTemplate
- GetEndpoint
- GetEventStream

- GetExportJob
- GetExportJobs
- GetGcmChannel
- GetImportJob
- GetImportJobs
- GetJourney
- GetJourneyDateRangeKpi
- GetJourneyExecutionActivityMetrics
- GetJourneyExecutionMetrics
- GetPushTemplate
- GetRecommenderConfiguration
- GetRecommenderConfigurations
- GetSegment
- GetSegmentExportJobs
- GetSegmentImportJobs
- GetSegmentVersion
- GetSegmentVersions
- GetSegments
- GetSmsChannel
- GetSmsTemplate
- GetUserEndpoints
- GetVoiceChannel
- GetVoiceTemplate
- ListJourneys
- ListTagsForResource
- ListTemplates
- ListTemplateVersions
- PhoneNumberValidate
- PutEvents

- PutEventStream
- RemoveAttributes
- SendMessages
- Enviar OTPMessage
- SendUsersMessages
- TagResource
- UntagResource
- UpdateAdmChannel
- UpdateApnsChannel
- UpdateApnsSandboxChannel
- UpdateApnsVoipChannel
- UpdateApnsVoipSandboxChannel
- UpdateApplicationSettings
- UpdateBaiduChannel
- UpdateCampaign
- UpdateEmailChannel
- UpdateEmailTemplate
- UpdateEndpoint
- UpdateEndpointsBatch
- UpdateGcmChannel
- UpdateJourney
- UpdateJourneyState
- UpdatePushTemplate
- UpdateRecommenderConfiguration
- UpdateSegment
- UpdateSmsChannel
- UpdateSmsTemplate
- UpdateTemplateActiveVersion
- UpdateVoiceChannel
- UpdateVoiceTemplate

Acciones de la API de correo electrónico Amazon Pinpoint compatibles en CloudTrail los archivos de registro

La API de correo electrónico de Amazon Pinpoint admite el registro de las siguientes acciones como eventos en los archivos de CloudTrail registro:

- 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

La siguiente acción de la API de Amazon Pinpoint Email no ha iniciado sesión: CloudTrail

- SendEmail

Acciones de la versión 1 de la API de voz y SMS de Amazon Pinpoint compatibles en CloudTrail los archivos de registro

La versión 1 de la API Amazon Pinpoint SMS and Voice admite el registro de las siguientes acciones como eventos en los archivos de CloudTrail registro:

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- GetConfigurationSetEventDestinations
- UpdateConfigurationSetEventDestination

La siguiente acción de la API Amazon Pinpoint SMS and Voice versión 1 no ha iniciado sesión: CloudTrail

- SendVoiceMessage

CloudTrail ejemplos de entradas de registro que muestran las acciones de la API Amazon Pinpoint

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o

más entradas de registro. Un evento representa una única solicitud desde cualquier origen. Incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a las API públicas, por lo que no aparecen en ningún orden específico.

El siguiente ejemplo muestra una entrada de CloudTrail registro que muestra las `CreateCampaign` acciones `GetCampaigns` y las acciones de la API Amazon Pinpoint.


```
{
  "Records": [
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "GetCampaigns",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T00:56:48Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": true,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "application-id": "example71dfa4c1aab66332a5839798f",
        "page-size": "1000"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Jersey/${project.version} (HttpUrlConnection 1.8.0_144)",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "principalId": "123456789012",
        "sessionContext": {
          "attributes": {
            "creationDate": "2018-02-02T16:55:29Z",
            "mfaAuthenticated": "false"
          }
        },
        "type": "Root"
      }
    },
    {
```

```
    "awsRegion": "us-east-1",
    "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
    "eventName": "CreateCampaign",
    "eventSource": "pinpoint.amazonaws.com",
    "eventTime": "2018-02-03T01:05:16Z",
    "eventType": "AwsApiCall",
    "eventVersion": "1.05",
    "readOnly": false,
    "recipientAccountId": "123456789012",
    "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
    "requestParameters": {
      "Description": "***",
      "HoldoutPercent": 0,
      "IsPaused": false,
      "MessageConfiguration": "***",
      "Name": "***",
      "Schedule": {
        "Frequency": "ONCE",
        "IsLocalTime": true,
        "StartTime": "2018-02-03T00:00:00-08:00",
        "Timezone": "utc-08"
      },
      "SegmentId": "exampleda204adf991a80281aa0e591",
      "SegmentVersion": 1,
      "application-id": "example71dfa4c1aab66332a5839798f"
    },
    "responseElements": {
      "ApplicationId": "example71dfa4c1aab66332a5839798f",
      "CreationDate": "2018-02-03T01:05:16.425Z",
      "Description": "***",
      "HoldoutPercent": 0,
      "Id": "example54a654f80948680cbba240ede",
      "IsPaused": false,
      "LastModifiedDate": "2018-02-03T01:05:16.425Z",
      "MessageConfiguration": "***",
      "Name": "***",
      "Schedule": {
        "Frequency": "ONCE",
        "IsLocalTime": true,
        "StartTime": "2018-02-03T00:00:00-08:00",
        "Timezone": "utc-08"
      },
      "SegmentId": "example4da204adf991a80281example",
      "SegmentVersion": 1,
```

```
        "State": {
          "CampaignStatus": "SCHEDULED"
        },
        "Version": 1
      },
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.9 Python/3.4.3 Linux/3.4.0+ botocore/1.8.34",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/userName",
        "principalId": "AIDAIHTHRCDA62EXAMPLE",
        "type": "IAMUser",
        "userName": "userName"
      }
    }
  ]
}
```

El siguiente ejemplo muestra una entrada de CloudTrail registro que muestra `CreateConfigurationSetEventDestination` las acciones `CreateConfigurationSet` y las acciones en la API de SMS y voz de Amazon Pinpoint.

```
{
  "Records": [
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:45:55Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSet",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "ConfigurationSetName":"MyConfigurationSet"
```

```
      },
      "responseElements":null,
      "requestID":"56dcc091-e20d-11e8-87d2-9994aexample",
      "eventID":"725843fc-8846-41f4-871a-7c52dexample",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "recipientAccountId":"123456789012"
    },
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:47:08Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSetEventDestination",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "EventDestinationName":"CloudWatchEventDestination",
        "ConfigurationSetName":"MyConfigurationSet",
        "EventDestination":{
          "Enabled":true,
          "MatchingEventTypes":[
            "INITIATED_CALL",
            "INITIATED_CALL"
          ],
          "CloudWatchLogsDestination":{
            "IamRoleArn":"arn:aws:iam::111122223333:role/iamrole-01",
            "LogGroupArn":"arn:aws:logs:us-east-1:111122223333:log-
group:clientloggroup-01"
          }
        }
      },
      "responseElements":null,
      "requestID":"81de1e73-e20d-11e8-b158-d5536example",
      "eventID":"fcafc21f-7c93-4a3f-9e72-fca2dexample",
      "readOnly":false,
```

```
      "eventType":"AwsApiCall",
      "recipientAccountId":"111122223333"
    }
  ]
}
```

Utilice un modelo de recomendación en Amazon Pinpoint con AWS Lambda

En Amazon Pinpoint, puede recuperar recomendaciones personalizadas de un modelo de recomendación y agregarlas a los mensajes que envíe desde campañas y recorridos. Un modelo de recomendación es un tipo de modelo de aprendizaje automático (ML) que encuentra patrones en los datos y genera predicciones y recomendaciones basadas en los patrones que encuentra. Predice lo que un usuario en particular preferirá de un determinado conjunto de productos o elementos, y proporciona esa información como un conjunto de recomendaciones para el usuario.

Mediante el uso de modelos de recomendación con Amazon Pinpoint, puede enviar recomendaciones personalizadas a los destinatarios de mensajes en función de los atributos y el comportamiento de cada destinatario. Con AWS Lambda ella, también puedes personalizar y mejorar estas recomendaciones. Por ejemplo, puede transformar dinámicamente una recomendación de un solo valor de texto (como un nombre de producto o un ID) a contenido más sofisticado (como un nombre de producto, una descripción y una imagen). Y puede hacerlo en tiempo real, cuando Amazon Pinpoint envía el mensaje.

Esta función está disponible en las siguientes AWS regiones: EE.UU. Este (Norte de Virginia); EE.UU. Oeste (Oregón); Asia Pacífico (Bombay); Asia Pacífico (Sídney); y Europa (Irlanda).

Añadir recomendaciones de modelos de recomendación a los mensajes en Amazon Pinpoint

Para utilizar un modelo de recomendación con Amazon Pinpoint, comienza creando una solución de Amazon Personalize e implementando esa solución como una campaña de Amazon Personalize. A continuación, se crea una configuración para el modelo de recomendación en Amazon Pinpoint. En la configuración, especifique la configuración que determina cómo recuperar y procesar los datos de recomendación de la campaña de Amazon Personalize. Esto incluye la posibilidad de invocar una AWS Lambda función para realizar un procesamiento adicional de los datos recuperados.

Amazon Personalize es un AWS servicio diseñado para ayudarle a crear modelos de aprendizaje automático que ofrezcan recomendaciones personalizadas en tiempo real a los clientes que utilizan sus aplicaciones. Amazon Personalize lo guía a través del proceso de creación y entrenamiento de un modelo de ML y, posteriormente, de preparación e implementación del modelo como

una campaña de Amazon Personalize. A continuación, puede recuperar recomendaciones personalizadas en tiempo real de la campaña. Para obtener más información sobre Amazon Personalize, consulte la Guía para desarrolladores de Amazon Personalize.

AWS Lambda es un servicio informático que puede utilizar para ejecutar código sin aprovisionar ni administrar servidores. Empaqueta el código y lo carga AWS Lambda como una función Lambda. AWS Lambda luego ejecuta la función cuando se invoca la función. Una función se puede invocar manualmente, automáticamente en respuesta a eventos o en respuesta a solicitudes de aplicaciones o servicios, como Amazon Pinpoint. Para obtener más información sobre cómo crear e invocar funciones de Lambda, consulte la Guía para desarrolladores de AWS Lambda.

Después de crear una configuración de Amazon Pinpoint para un modelo de recomendación, puede agregar recomendaciones del modelo a los mensajes que envíe desde campañas y recorridos. Para ello, utilice plantillas de mensaje que contienen variables de mensaje para los atributos recomendados. Un atributo recomendado es un punto de enlace dinámico o atributo de usuario diseñado para almacenar datos de recomendación. Estos atributos se definen al crear la configuración para un modelo de recomendación.

Puede utilizar variables para los atributos recomendados en los siguientes tipos de plantillas de mensaje:

- Plantillas de correo electrónico, para los mensajes de correo electrónico que envía desde campañas o recorridos.
- Plantillas de notificaciones push, para notificaciones push que envía desde campañas.
- Plantillas de SMS, para mensajes de texto SMS que envía desde campañas.

Para obtener más información acerca del uso de modelos de recomendación con Amazon Pinpoint, consulte Modelos de machine learning en la Guía para usuarios de Amazon Pinpoint.

Si configura Amazon Pinpoint para invocar una función de Lambda que procesa datos de recomendación, Amazon Pinpoint realiza las siguientes tareas generales cada vez que envía recomendaciones personalizadas en un mensaje para una campaña o recorrido:

1. Evalúa y procesa la configuración y el contenido del mensaje de la plantilla de mensaje.
2. Determina que la plantilla de mensaje está conectada a un modelo de recomendación.
3. Evalúa los parámetros de configuración para conectarse al modelo y utilizarlo. Estos se definen mediante el recurso Modelo de recomendación para el modelo.

4. Detecta una o más variables de mensaje para los atributos recomendados definidos por los parámetros de configuración del modelo.
5. Recupera los datos de recomendación de la campaña de Amazon Personalize que se especifican en la configuración del modelo. Utiliza la GetRecommendationsoperación de la API Amazon Personalize Runtime para realizar esta tarea.
6. Agrega los datos de recomendación adecuados a un atributo dinámico recomendado (`RecommendationItems`) para cada destinatario del mensaje.
7. Invoca la función de Lambda y envía los datos de recomendación de cada destinatario a esa función para su procesamiento.

   Los datos se envían como un objeto JSON que contiene la definición de punto de enlace para cada destinatario. Cada definición de punto de conexión incluye un campo `RecommendationItems` que contiene una matriz ordenada de 1 a 5 valores. El número de valores de la matriz depende de los parámetros de configuración del modelo.
8. Espera a que la función de Lambda procese los datos y devuelva los resultados.

   Los resultados son un objeto JSON que contiene una definición de punto de enlace actualizada para cada destinatario. Cada definición de punto de enlace actualizada contiene un nuevo objeto `Recommendations`. Este objeto contiene 1-10 campos, uno para cada atributo recomendado personalizado definido en los parámetros de configuración del modelo. Cada uno de estos campos almacena datos de recomendación mejorados para el punto de enlace.
9. Utiliza la definición de punto de enlace actualizada para que cada destinatario sustituya cada variable de mensaje por el valor adecuado para ese destinatario.
10. Envía una versión del mensaje que contiene las recomendaciones personalizadas para cada destinatario del mensaje.

Para personalizar y mejorar las recomendaciones de esta manera, comience por crear una función de Lambda que procese las definiciones de punto de conexión enviadas por Amazon Pinpoint y devuelva las definiciones de punto de conexión actualizadas. A continuación, asigne una política de función de Lambda a la función y autorice a Amazon Pinpoint a invocar la función. A continuación, configure el modelo de recomendación en Amazon Pinpoint. Al configurar el modelo, especifique la función que se va a invocar y defina los atributos recomendados que se van a utilizar.

Crear una función de Lambda para que Amazon Pinpoint la invoque para un modelo de recomendaciones

Para obtener información sobre cómo crear una función de Lambda, consulte Introducción en la Guía para desarrolladores de AWS Lambda . Cuando diseñe y desarrolle la función, tenga en cuenta los siguientes requisitos y directrices.

Datos de eventos de entrada

Cuando Amazon Pinpoint invoca una función de Lambda para un modelo de recomendación, envía una carga que contiene la configuración y otros ajustes para la campaña o recorrido que envía el mensaje. La carga útil incluye un `Endpoints` objeto, que es un mapa que asocia el punto final IDs con las definiciones de punto final para los destinatarios de los mensajes.

Las definiciones de punto de conexión utilizan la estructura definida por el recurso de punto de conexión de la API de Amazon Pinpoint. Sin embargo, también incluyen un campo para un atributo dinámico recomendado denominado `RecommendationItems`. El campo `RecommendationItems` contiene uno o varios elementos recomendados para el punto de conexión, tal como se devuelve de la campaña de Amazon Personalize. El valor de este campo es una matriz ordenada de 1 a 5 elementos recomendados (como cadenas). El número de elementos de la matriz depende del número de elementos recomendados que haya configurado Amazon Pinpoint para recuperar para cada punto de conexión o usuario.

Por ejemplo:

```
"Endpoints": {
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
```

```
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z"
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
```

```
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z"
    }
}
```

En el ejemplo anterior, la configuración de Amazon Pinpoint relevante es:

- El modelo de recomendación está configurado para recuperar tres elementos recomendados para cada punto de enlace o usuario. (El valor de la propiedad `RecommendationsPerMessage` se establece en 3). Con esta configuración, Amazon Pinpoint recupera y agrega solo los elementos recomendados primero, segundo y tercero para cada punto de conexión o usuario.
- El proyecto está configurado para utilizar atributos de usuario personalizados que almacenan el nombre, el apellido y el vecindario de cada usuario. (El objeto `UserAttributes` contiene los valores de estos atributos).
- El proyecto está configurado para utilizar un atributo de punto de enlace personalizado (`AddressType`) que indica si el punto de enlace es la dirección preferida del usuario (canal) para recibir mensajes del proyecto. (El objeto `Attributes` contiene el valor de este atributo).

Cuando Amazon Pinpoint invoca la función de Lambda y envía esta carga como datos de evento, AWS Lambda transfiere los datos a la función de Lambda para su procesamiento.

Cada carga útil puede contener datos de hasta 50 puntos de enlace. Si un segmento contiene más de 50 puntos de conexión, Amazon Pinpoint invoca la función repetidamente, para un máximo de 50 puntos de conexión a la vez, hasta que la función procesa todos los datos.

Datos y requisitos de respuesta

A medida que diseñe y desarrolle la función de Lambda, tenga en cuenta las cuotas para los modelos de machine learning. Si la función no cumple las condiciones definidas por estas cuotas, Amazon Pinpoint no podrá procesar y enviar el mensaje.

Tenga en cuenta también los siguientes requisitos:

- La función debe devolver definiciones actualizadas de punto de enlace en el mismo formato que proporcionaron los datos de evento de entrada.

- Cada definición de punto de conexión actualizada puede contener 1-10 atributos recomendados personalizados para el punto de conexión o el usuario. Los nombres de estos atributos deben coincidir con los nombres de atributo que especifique al configurar el modelo de recomendación en Amazon Pinpoint.
- Todos los atributos personalizados recomendados deben devolverse en un solo objeto `Recommendations` para cada punto de enlace o usuario. Este requisito ayuda a garantizar que no se produzcan conflictos de nombres. Puede agregar el objeto `Recommendations` a cualquier ubicación en una definición de punto de enlace.
- El valor de cada atributo recomendado personalizado tiene que ser una cadena (valor único) o una matriz de cadenas (valores múltiples). Si el valor es una matriz de cadenas, le recomendamos que mantenga el orden de los elementos recomendados que devolvió Amazon Personalize, como se indica en el campo `RecommendationItems`. De lo contrario, es posible que el contenido no refleje las predicciones del modelo para un punto de enlace o usuario.
- La función no debe modificar otros elementos en los datos del evento, incluidos otros valores de atributo para un punto de enlace o usuario. Solo debe agregar y devolver valores para los atributos personalizados recomendados. Amazon Pinpoint no aceptará actualizaciones de ningún otro valor de la respuesta de la función.
- La función debe estar alojada en la misma AWS región que el proyecto Amazon Pinpoint que invoca la función. Si la función y el proyecto no están en la misma región, Amazon Pinpoint no puede enviar datos de eventos a la función.

Si no se cumple ninguno de los requisitos anteriores, Amazon Pinpoint no será capaz de procesar y enviar el mensaje a uno o más puntos de conexión. Esto puede provocar un error en la actividad de la campaña o recorrido.

Por último, se recomienda reservar 256 ejecuciones simultáneas para la función.

En general, la función de Lambda debe procesar los datos de evento que envía Amazon Pinpoint y devolver definiciones de punto de conexión modificadas. Puede hacerlo iterando a través de cada punto de enlace en el objeto `Endpoints` y, para cada punto de enlace, creando y estableciendo valores para los atributos recomendados personalizados que desee usar. El controlador de ejemplo siguiente, escrito en Python y siguiendo con el ejemplo anterior de datos de eventos de entrada, muestra esto:

```
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
```

En el ejemplo anterior, AWS Lambda pasa los datos del evento al controlador como parámetro. `event` El controlador itera a través de cada punto de enlace del objeto `Endpoints` y establece valores para los atributos recomendados personalizados denominados `Recommendations.Title` y `Recommendations.Genre`. La instrucción `return` devuelve cada definición de punto de conexión actualizada a Amazon Pinpoint.

Siguiendo con el ejemplo anterior de datos de eventos de entrada, las definiciones de punto de enlace actualizadas son:

```
"Endpoints":{
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
```

```
        "CreationDate":"2020-02-26T18:56:24.875Z",
        "Recommendations":{
            "Title":[
                "Fleabag",
                "Late Night",
                "Catastrophe"
            ],
            "Genre":[
                "Comedy",
                "Comedy",
                "Comedy"
            ]
        }
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1210",
            "6542",
            "4582"
        ],
```

```
        "CreationDate":"2020-02-26T18:56:24.897Z",
        "Recommendations":{
            "Title":[
                "Hanna",
                "Hunters",
                "Agatha Christie\'s The ABC Murders"
            ],
            "Genre":[
                "Action",
                "Drama",
                "Crime"
            ]
        }
    }
}
```

En el ejemplo anterior, la función modificó el objeto `Endpoints` que recibió y devolvió los resultados. El objeto `Endpoint` de cada punto de enlace contiene ahora un nuevo objeto `Recommendations`, que contiene los campos `Title` y `Genre`. Cada uno de estos campos almacena una matriz ordenada de tres valores (como cadenas), donde cada valor proporciona contenido mejorado para un elemento recomendado correspondiente en el campo `RecommendationItems`.

Asignar una política de función de Lambda para autorizar a Amazon Pinpoint a procesar datos de recomendaciones

Antes de poder utilizar la función de Lambda para procesar los datos de recomendación, debe autorizar a que Amazon Pinpoint invoque la función. Para conceder permiso de invocación, asigne una política de función de Lambda a la función. Una política de función de Lambda consiste en una política de permisos basada en el recurso que designa qué entidades pueden usar una función y qué acciones pueden realizar dichas entidades. Para obtener más información, consulte Uso de políticas basadas en recursos para AWS Lambda en la Guía para desarrolladores de AWS Lambda .

El siguiente ejemplo de política permite al director del servicio Amazon Pinpoint utilizar la `lambda:InvokeFunction` acción para una campaña concreta de Amazon Pinpoint *`campaignId`* () en un proyecto concreto de Amazon Pinpoint (): *`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 política de función requiere un bloque `Condition` que incluye una clave `AWS:SourceArn`. Esta clave especifica qué recurso puede invocar la función. En el ejemplo anterior, la directiva permite que una campaña concreta invoque la función.

También puede redactar una política que permita al director del servicio Amazon Pinpoint utilizar la `lambda:InvokeFunction` acción en todas las campañas y recorridos de un proyecto específico de Amazon Pinpoint (). *`projectId`* La siguiente política de ejemplo se muestra esto:

```
{
  "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/*"
    }
  }
}
```

A diferencia del primer ejemplo, la clave `AWS:SourceArn` del bloque `Condition` de este ejemplo permite que un proyecto en particular invoque la función. Este permiso se aplica a todas las campañas y recorridos del proyecto.

Para escribir una política más genérica, utilice caracteres comodín (*) que sustituyan a varios caracteres. Por ejemplo, puede utilizar el siguiente bloque de `Condition` para permitir que cualquier proyecto de Amazon Pinpoint invoque la función:

```
"Condition": {
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}
```

Si desea utilizar la función de Lambda con todos los proyectos de la cuenta de Amazon Pinpoint, le recomendamos que configure el bloque `Condition` de la política de la forma anterior. Sin embargo, como práctica recomendada, debe crear políticas que incluyan solo los permisos necesarios para realizar una acción específica en un recurso específico.

Autorizar a Amazon Pinpoint a invocar una función de Lambda mediante el comando and AWS CLI the Lambda add-permission

Después de asignar una política de función de Lambda a una función, puede agregar permisos que permitan a Amazon Pinpoint invocar la función para un proyecto, campaña o recorrido específico. Puede hacerlo mediante el comando AWS Command Line Interface (AWS CLI) y el comando Lambda `add-permission`. El siguiente ejemplo muestra cómo hacerlo para un proyecto específico (*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/*
```

El ejemplo anterior está preparado para Unix, Linux y macOS. En Microsoft Windows, reemplace el carácter de continuación de línea de barra diagonal inversa (\) por un símbolo de intercalación (^).

Si el comando se ejecuta correctamente, aparecerá un resultado similar al siguiente:

```
{
  "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/*\"}}}"
}
```

El valor de `Statement` es una versión de cadena JSON de la instrucción que se agregó a la política de la función de Lambda.

Configurar Amazon Pinpoint para invocar la función de Lambda para un modelo de recomendaciones

Para configurar Amazon Pinpoint para invocar la función de Lambda de un modelo de recomendación, especifique los siguientes ajustes de configuración específicos de Lambda para el modelo:

- `RecommendationTransformerUri`: esta propiedad especifica el nombre o el nombre de recurso de Amazon (ARN) de la función de Lambda.
- `Attributes`: este objeto es un mapa que define los atributos personalizados recomendados que la función agrega a cada definición de punto de conexión. Cada uno de estos atributos se puede utilizar como variable de mensaje en una plantilla de mensaje.

Puede especificar estos valores mediante el recurso Modelos de recomendación de la API de Amazon Pinpoint (al crear la configuración de un modelo) o el recurso Modelo de recomendación de la API de Amazon Pinpoint (si actualiza la configuración de un modelo). También puede definir estos ajustes mediante la consola de Amazon Pinpoint.

Para obtener más información acerca del uso de modelos de recomendación con Amazon Pinpoint, consulte Modelos de machine learning en la Guía para usuarios de Amazon Pinpoint.

Eliminar un proyecto de Amazon Pinpoint y los datos personales confidenciales

En función de cómo lo utilice, Amazon Pinpoint puede almacenar determinados datos que podrían considerarse personales. Por ejemplo, un punto de conexión de Amazon Pinpoint contiene información de contacto de un usuario final, como la dirección de correo electrónico o el número de teléfono móvil.

Puede utilizar la consola o la API de Amazon Pinpoint para eliminar de forma permanente los datos personales. Este tema incluye procedimientos para eliminar diversos tipos de datos que podrían considerarse personales.

También puede cerrar su cuenta de AWS por completo. Para obtener más información, consulte Close an AWS account en la Guía de referencia de AWS Account Management.

Eliminar todos los datos de un proyecto de Amazon Pinpoint

Puede eliminar de forma permanente todos los datos almacenados para un proyecto de Amazon Pinpoint. Puede hacerlo eliminando el proyecto.

> **Warning**
>
> Si elimina un proyecto, Amazon Pinpoint elimina todos los ajustes y datos específicos del proyecto. La información no se puede recuperar.

Cuando se elimina un proyecto, Amazon Pinpoint elimina toda la configuración específica del proyecto para los canales de notificación de inserción y mensajería SMS bidireccional, así como todos los segmentos, campañas, recorridos y datos de análisis específicos del proyecto almacenados en Amazon Pinpoint, como los siguientes:

- Segmentos: todos los datos y la configuración de los segmentos. Para los segmentos dinámicos, esto incluye los grupos de segmento y filtros que ha definido. Para los segmentos importados, esto incluye los puntos de enlace, ID de usuario y otros datos que ha importado. así como los filtros que haya aplicado.
- Campañas: todos los mensajes, los tratamientos y las variables de los mensajes, los datos de análisis, las programaciones y otros ajustes.

- Recorridos: todas las actividades, datos analíticos, programaciones y otros ajustes.
- Análisis: datos de todas las métricas de participación, como la cantidad de mensajes enviados y entregados para las campañas y los recorridos y todas las métricas de ejecución de los recorridos. Para aplicaciones web y móviles, esto incluye todos los datos de eventos que no se transmitieron a otro servicio de AWS como, por ejemplo, Amazon Kinesis, todos los embudos y los datos de métricas de uso de aplicación, ingresos y datos demográficos. Antes de eliminar un proyecto, le recomendamos que exporte estos datos a otra ubicación.

Puede eliminar un proyecto con la consola de Amazon Pinpoint. Para obtener más información, consulte Eliminar un proyecto en la Guía del usuario de Amazon Pinpoint. También puede eliminar un proyecto mediante programación con el recurso App de la API de Amazon Pinpoint.

Ejemplos de código para Amazon Pinpoint con AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar Amazon Pinpoint con un kit de desarrollo de software (SDK) de AWS.

Para obtener una lista completa de las guías para desarrolladores de AWS SDK y ejemplos de código, consulte Uso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Ejemplos de código para Amazon Pinpoint mediante AWS SDKs
  - Ejemplos básicos del uso de Amazon Pinpoint AWS SDKs
    - Acciones para Amazon Pinpoint mediante AWS SDKs
      - Úselo CreateApp con un AWS SDK o CLI
      - CreateCampaignÚselo con un AWS SDK
      - CreateExportJobÚselo con un AWS SDK
      - CreateImportJobÚselo con un AWS SDK
      - CreateSegmentÚselo con un AWS SDK
      - Úselo DeleteApp con un AWS SDK o CLI
      - DeleteEndpointÚselo con un AWS SDK
      - Úselo GetEndpoint con un AWS SDK o CLI
      - GetSegmentsÚselo con un AWS SDK
      - Úselo GetSmsChannel con un AWS SDK o CLI
      - GetUserEndpointsÚselo con un AWS SDK
      - Úselo SendMessages con un AWS SDK o CLI
      - UpdateEndpointÚselo con un AWS SDK
- Ejemplos de código para la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs
  - Ejemplos básicos de la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs
    - Acciones para la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs
      - CreateConfigurationSetÚselo con un AWS SDK
      - CreateConfigurationSetEventDestinationÚselo con un AWS SDK
      - DeleteConfigurationSetÚselo con un AWS SDK

- DeleteConfigurationSetEventDestinationÚselo con un AWS SDK
- GetConfigurationSetEventDestinationsÚselo con un AWS SDK
- ListConfigurationSetsÚselo con un AWS SDK
- SendVoiceMessageÚselo con un AWS SDK
- UpdateConfigurationSetEventDestinationÚselo con un AWS SDK

Ejemplos de código para Amazon Pinpoint mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar Amazon Pinpoint con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Amazon Pinpoint

- Ejemplos básicos del uso de Amazon Pinpoint AWS SDKs
  - Acciones para Amazon Pinpoint mediante AWS SDKs
    - Úselo CreateApp con un AWS SDK o CLI
    - CreateCampaignÚselo con un AWS SDK
    - CreateExportJobÚselo con un AWS SDK
    - CreateImportJobÚselo con un AWS SDK
    - CreateSegmentÚselo con un AWS SDK
    - Úselo DeleteApp con un AWS SDK o CLI
    - DeleteEndpointÚselo con un AWS SDK
    - Úselo GetEndpoint con un AWS SDK o CLI
    - GetSegmentsÚselo con un AWS SDK
    - Úselo GetSmsChannel con un AWS SDK o CLI
    - GetUserEndpointsÚselo con un AWS SDK
    - Úselo SendMessages con un AWS SDK o CLI

- UpdateEndpointÚselo con un AWS SDK

Ejemplos básicos del uso de Amazon Pinpoint AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de Amazon Pinpoint con. AWS SDKs

Ejemplos

- Acciones para Amazon Pinpoint mediante AWS SDKs
  - Úselo CreateApp con un AWS SDK o CLI
  - CreateCampaignÚselo con un AWS SDK
  - CreateExportJobÚselo con un AWS SDK
  - CreateImportJobÚselo con un AWS SDK
  - CreateSegmentÚselo con un AWS SDK
  - Úselo DeleteApp con un AWS SDK o CLI
  - DeleteEndpointÚselo con un AWS SDK
  - Úselo GetEndpoint con un AWS SDK o CLI
  - GetSegmentsÚselo con un AWS SDK
  - Úselo GetSmsChannel con un AWS SDK o CLI
  - GetUserEndpointsÚselo con un AWS SDK
  - Úselo SendMessages con un AWS SDK o CLI
  - UpdateEndpointÚselo con un AWS SDK

Acciones para Amazon Pinpoint mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Pinpoint con. AWS SDKs Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la Referencia de la API de Amazon Pinpoint.

Ejemplos

- Úselo CreateApp con un AWS SDK o CLI

- CreateCampaignÚselo con un AWS SDK
- CreateExportJobÚselo con un AWS SDK
- CreateImportJobÚselo con un AWS SDK
- CreateSegmentÚselo con un AWS SDK
- Úselo DeleteApp con un AWS SDK o CLI
- DeleteEndpointÚselo con un AWS SDK
- Úselo GetEndpoint con un AWS SDK o CLI
- GetSegmentsÚselo con un AWS SDK
- Úselo GetSmsChannel con un AWS SDK o CLI
- GetUserEndpointsÚselo con un AWS SDK
- Úselo SendMessages con un AWS SDK o CLI
- UpdateEndpointÚselo con un AWS SDK

Úselo **`CreateApp`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `CreateApp`.

CLI

AWS CLI

Ejemplo 1: Para crear una aplicación

En el siguiente ejemplo de `create-app` se crea una nueva aplicación (proyecto).

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

Salida:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
```

```
}
```

Ejemplo 2: Creación de una aplicación etiquetada

En el siguiente ejemplo de `create-app` se crea una nueva aplicación (proyecto) y asocia una etiqueta (clave y valor) a la aplicación.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

Salida:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
            "Stack": "Test"
        }
    }
}
```

- Para obtener más información sobre la API, consulte CreateAppla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
}
```

- Para obtener más información sobre la API, consulta CreateAppla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun createApplication(applicationName: String?): String? {
    val createApplicationRequestOb =
        CreateApplicationRequest {
            name = applicationName
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
```

```
        val result =
            pinpoint.createApp(
                CreateAppRequest {
                    createApplicationRequest = createApplicationRequestOb
                },
            )
        return result.applicationResponse?.id
    }
}
```

- Para obtener más información sobre la API, consulta CreateAppla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateCampaign`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateCampaign`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear una campaña.

```
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;
```

```
    }
}
```

- Para obtener más información sobre la API, consulta CreateCampaignla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun createPinCampaign(
    appId: String,
    segmentIdVal: String,
) {
    val scheduleOb =
        Schedule {
            startTime = "IMMEDIATE"
        }

    val defaultMessageOb =
        Message {
            action = Action.OpenApp
            body = "My message body"
            title = "My message title"
        }

    val messageConfigurationOb =
        MessageConfiguration {
            defaultMessage = defaultMessageOb
        }

    val writeCampaign =
        WriteCampaignRequest {
            description = "My description"
```

```
            schedule = scheduleOb
            name = "MyCampaign"
            segmentId = segmentIdVal
            messageConfiguration = messageConfigurationOb
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result: CreateCampaignResponse =
            pinpoint.createCampaign(
                CreateCampaignRequest {
                    applicationId = appId
                    writeCampaignRequest = writeCampaign
                },
            )
        println("Campaign ID is ${result.campaignResponse?.id}")
    }
}
```

- Para obtener más información sobre la API, consulta CreateCampaignla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateExportJob`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateExportJob`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Exportar un punto de conexión.

```
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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta CreateExportJobla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateImportJob`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateImportJob`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar un segmento.

```
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;
    }
}
```

- Para obtener más información sobre la API, consulta CreateImportJobla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateSegment`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `CreateSegment`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
}
```

- Para obtener más información sobre la API, consulta CreateSegmentla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun createPinpointSegment(applicationIdVal: String?): String? {
    val segmentAttributes = mutableMapOf<String, AttributeDimension>()
    val myList = mutableListOf<String>()
    myList.add("Lakers")

    val atts =
        AttributeDimension {
            attributeType = AttributeType.Inclusive
            values = myList
        }

    segmentAttributes["Team"] = atts
    val recencyDimension =
        RecencyDimension {
            duration = Duration.fromValue("DAY_30")
            recencyType = RecencyType.fromValue("ACTIVE")
        }

    val segmentBehaviors =
        SegmentBehaviors {
            recency = recencyDimension
        }

    val segmentLocation = SegmentLocation {}
    val dimensionsOb =
        SegmentDimensions {
            attributes = segmentAttributes
            behavior = segmentBehaviors
            demographic = SegmentDemographics {}
            location = segmentLocation
        }

    val writeSegmentRequestOb =
        WriteSegmentRequest {
            name = "MySegment101"
            dimensions = dimensionsOb
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val createSegmentResult: CreateSegmentResponse =
            pinpoint.createSegment(
                CreateSegmentRequest {
                    applicationId = applicationIdVal
```

```
                    writeSegmentRequest = writeSegmentRequestOb
                },
            )
        println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
        return createSegmentResult.segmentResponse?.id
    }
}
```

- Para obtener más información sobre la API, consulta CreateSegmentla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DeleteApp`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteApp`.

CLI

AWS CLI

Para eliminar una aplicación

En el siguiente ejemplo `delete-app` se elimina una aplicación (proyecto).

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

Salida:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
```

```
}
```

- Para obtener más información sobre la API, consulte DeleteAppla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine una aplicación.

```
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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta DeleteAppla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun deletePinApp(appId: String?) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.deleteApp(
                DeleteAppRequest {
                    applicationId = appId
                },
            )
        val appName = result.applicationResponse?.name
        println("Application $appName has been deleted.")
    }
}
```

- Para obtener más información sobre la API, consulta DeleteAppla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DeleteEndpoint`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `DeleteEndpoint`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Eliminar un punto de conexión.

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

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

                Usage:   <appName> <endpointId >

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

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

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

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

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

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

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

- Para obtener más información sobre la API, consulta DeleteEndpointla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun deletePinEncpoint(
    appIdVal: String?,
    endpointIdVal: String?,
) {
    val deleteEndpointRequest =
        DeleteEndpointRequest {
            applicationId = appIdVal
            endpointId = endpointIdVal
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.deleteEndpoint(deleteEndpointRequest)
        val id = result.endpointResponse?.id
        println("The deleted endpoint is  $id")
    }
}
```

- Para obtener más información sobre la API, consulta DeleteEndpointla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`GetEndpoint`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetEndpoint`.

CLI

AWS CLI

Para recuperar información sobre la configuración y los atributos de un punto de conexión específico de una aplicación

En el siguiente ejemplo `get-endpoint` se recupera información sobre la configuración y los atributos de un punto de conexión específico de una aplicación.

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

Salida:

```
{
    "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"
```

```
        }
    }
}
```

- Para obtener más información sobre la API, consulte GetEndpointla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
    }
}
```

- Para obtener más información sobre la API, consulta GetEndpointla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun lookupPinpointEndpoint(
    appId: String?,
    endpoint: String?,
) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.getEndpoint(
                GetEndpointRequest {
                    applicationId = appId
                    endpointId = endpoint
                },
            )
        val endResponse = result.endpointResponse

        // Uses the Google Gson library to pretty print the endpoint JSON.
        val gson: com.google.gson.Gson =
            GsonBuilder()
                .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                .setPrettyPrinting()
                .create()
```

```
        val endpointJson: String = gson.toJson(endResponse)
        println(endpointJson)
    }
}
```

- Para obtener más información sobre la API, consulta GetEndpointla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`GetSegments`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `GetSegments`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enumerar segmentos.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
```

```
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListSegments {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId>

                Where:
                  appId - The ID of the application that contains a segment.

                """;

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

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

        listSegs(pinpoint, appId);
        pinpoint.close();
    }

    public static void listSegs(PinpointClient pinpoint, String appId) {
        try {
            GetSegmentsRequest request = GetSegmentsRequest.builder()
                    .applicationId(appId)
                    .build();

            GetSegmentsResponse response = pinpoint.getSegments(request);
            List<SegmentResponse> segments = response.segmentsResponse().item();
            for (SegmentResponse segment : segments) {
                System.out
                        .println("Segement " + segment.id() + " " +
 segment.name() + " " + segment.lastModifiedDate());
            }

        } catch (PinpointException e) {
```

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

- Para obtener más información sobre la API, consulta GetSegmentsla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun listSegs(appId: String?) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val response =
            pinpoint.getSegments(
                GetSegmentsRequest {
                    applicationId = appId
                },
            )
        response.segmentsResponse?.item?.forEach { segment ->
            println("Segement id is ${segment.id}")
        }
    }
}
```

- Para obtener más información sobre la API, consulta GetSegmentsla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`GetSmsChannel`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetSmsChannel`.

CLI

AWS CLI

Para recuperar información sobre el estado y la configuración del canal de SMS para una aplicación

En el siguiente ejemplo de `get-sms-channel` se recupera el estado y la configuración del canal de SMS de una aplicación.

```
aws pinpoint get-sms-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Salida:

```
{
    "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
    }
}
```

- Para obtener más información sobre la API, consulte GetSmsChannella Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta GetSmsChannella Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`GetUserEndpoints`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `GetUserEndpoints`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta GetUserEndpointsla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`SendMessages`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `SendMessages`.

.NET

SDK para .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enviar un mensaje de correo electrónico.

```
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;
    }
}
```

Enviar un mensaje SMS.

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

namespace SendSmsMessage;

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

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
```

```
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

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

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

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

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

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

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

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
 {response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
```

```
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
{messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";


        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
```

```
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await
 client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

- Para obtener más información sobre la API, consulta SendMessagesla Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Envío de un mensaje SMS mediante el punto de conexión de una aplicación

En el siguiente ejemplo de `send-messages` se envía un mensaje directo a una aplicación con un punto de conexión.

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

Contenido de `myfile.json`:

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
```

```
    "Endpoints": {
        "testendpoint": {}
    }
}
```

Salida:

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

Para obtener más información, consulte Canal de SMS de Amazon Pinpoint en la Guía del usuario de Amazon Pinpoint.

- Para obtener más información sobre la API, consulta SendMessagesla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enviar un mensaje de correo electrónico.

```
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);
        }
    }
}
```

Envíe un mensaje de correo electrónico con valores de 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();
        }
    }
}
```

Enviar un mensaje SMS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

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

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

```
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

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

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

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application
 ID to use when you send this message.
                                  originationNumber - The phone number or
 short code that you specify has to be associated with your Amazon Pinpoint
 account. For best results, specify long codes in E.164 format (for example,
 +1-555-555-5654).
                                  destinationNumber - The recipient's phone
 number.  For best results, you should specify the phone number in E.164 format
 (for example, +1-555-555-5654).\s
                                  """;

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

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();
```

```
                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request =
SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();
```

```
                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" +
 v)));

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

Enviar mensajes SMS por lotes.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

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

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

```
 */
public class SendMessageBatch {

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

    // The registered keyword associated with the originating short code.
    public static String registeredKeyword = "myKeyword";

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

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

                Usage:   <message> <appId> <originationNumber>
 <destinationNumber> <destinationNumber1>\s

                Where:
                  message - The body of the message to send.
                  appId - The Amazon Pinpoint project/application ID to use when
 you send this message.
                  originationNumber - The phone number or short code that
 you specify has to be associated with your Amazon Pinpoint account. For best
 results, specify long codes in E.164 format (for example, +1-555-555-5654).
                  destinationNumber - The recipient's phone number.  For best
 results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).
                  destinationNumber1 - The second recipient's phone number.  For
 best results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).\s
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String message = args[0];
```

```
        String appId = args[1];
        String originationNumber = args[2];
        String destinationNumber = args[3];
        String destinationNumber1 = args[4];
        System.out.println("Sending a message");
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber, destinationNumber1);
        pinpoint.close();
    }

    public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                                      String originationNumber,
                                      String destinationNumber, String
destinationNumber1) {
        try {
            Map<String, AddressConfiguration> addressMap = new HashMap<String,
AddressConfiguration>();
            AddressConfiguration addConfig = AddressConfiguration.builder()
                    .channelType(ChannelType.SMS)
                    .build();

            // Add an entry to the Map object for each number to whom you want to
send a
            // message.
            addressMap.put(destinationNumber, addConfig);
            addressMap.put(destinationNumber1, addConfig);
            SMSMessage smsMessage = SMSMessage.builder()
                    .body(message)
                    .messageType(messageType)
                    .originationNumber(originationNumber)
                    .senderId(senderId)
                    .keyword(registeredKeyword)
                    .build();

            // Create a DirectMessageConfiguration object.
            DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                    .smsMessage(smsMessage)
                    .build();
```

```
            MessageRequest msgReq = MessageRequest.builder()
                    .addresses(addressMap)
                    .messageConfiguration(direct)
                    .build();

            // Create a SendMessagesRequest object.
            SendMessagesRequest request = SendMessagesRequest.builder()
                    .applicationId(appId)
                    .messageRequest(msgReq)
                    .build();

            SendMessagesResponse response = pinpoint.sendMessages(request);
            MessageResponse msg1 = response.messageResponse();
            Map map1 = msg1.result();

            // Write out the result of sendMessage.
            map1.forEach((k, v) -> System.out.println((k + ":" + v)));

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

- Para obtener más información sobre la API, consulta SendMessagesla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurarlo y ejecutarlo en el Repositorio de ejemplos de código de AWS.

Cree el cliente en un módulo separado y expórtelo.

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
// Set the AWS Region.
const REGION = "us-east-1";
export const pinClient = new PinpointClient({ region: REGION });
```

Enviar un mensaje de correo electrónico.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

// The subject line of the email.
const subject = "Amazon Pinpoint Test (AWS SDK for JavaScript in Node.js)";

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

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

// The character encoding for the subject line and message body of the email.
const charset = "UTF-8";
```

```
const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

const run = async () => {
  try {
    const { MessageResponse } = await pinClient.send(
      new SendMessagesCommand(params),
    );

    if (!MessageResponse) {
      throw new Error("No message response.");
    }

    if (!MessageResponse.Result) {
      throw new Error("No message result.");
    }
```

```
    const recipientResult = MessageResponse.Result[toAddress];

    if (recipientResult.StatusCode !== 200) {
      throw new Error(recipientResult.StatusMessage);
    }
    console.log(recipientResult.MessageId);
  } catch (err) {
    console.log(err.message);
  }
};

run();
```

Enviar un mensaje SMS.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

/* The phone number or short code to send the message from. The phone number
 or short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format. */
const originationNumber = "SENDER_NUMBER"; //e.g., +1XXXXXXXXXX

// The recipient's phone number.  For best results, you should specify the phone
 number in E.164 format.
const destinationNumber = "RECEIVER_NUMBER"; //e.g., +1XXXXXXXXXX

// The content of the SMS message.
const message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

/*The Amazon Pinpoint project/application ID to use when you send this message.
Make sure that the SMS channel is enabled for the project or application
that you choose.*/
const projectId = "PINPOINT_PROJECT_ID"; //e.g., XXXXXXXX66e4e9986478cXXXXXXXXX

/* The type of SMS message that you want to send. If you plan to send
time-sensitive content, specify TRANSACTIONAL. If you plan to send
```

```
marketing-related content, specify PROMOTIONAL.*/
const messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
const registeredKeyword = "myKeyword";

/* The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html.*/

const senderId = "MySenderID";

// Specify the parameters to pass to the API.
const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

const run = async () => {
  try {
    const data = await pinClient.send(new SendMessagesCommand(params));
    console.log(
      `Message sent!
 ${data.MessageResponse.Result[destinationNumber].StatusMessage}`,
    );
  } catch (err) {
    console.log(err);
  }
```

```
};
run();
```

- Para obtener más información sobre la API, consulta SendMessagesla Referencia AWS SDK para JavaScript de la API.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enviar un mensaje de correo electrónico.

```
"use strict";

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

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

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

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

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

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";
```

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

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

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

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

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

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

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

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

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

Enviar un mensaje 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"]
    );
  }
});
```

- Para obtener más información sobre la API, consulta SendMessagesla Referencia AWS SDK para JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */

val body: String =
    """
    Amazon Pinpoint test (AWS SDK for Kotlin)

    This email was sent through the Amazon Pinpoint Email API using the AWS SDK
 for Kotlin.

    """.trimIndent()

suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <subject> <appId> <senderAddress> <toAddress>

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

```
        toAddress - The to address. This address has to be verified in Amazon
 Pinpoint in the region you're using to send email
    """

    if (args.size != 3) {
        println(usage)
        exitProcess(0)
    }

    val subject = args[0]
    val senderAddress = args[1]
    val toAddress = args[2]
    sendEmail(subject, senderAddress, toAddress)
}

suspend fun sendEmail(
    subjectVal: String?,
    senderAddress: String,
    toAddressVal: String,
) {
    var content =
        Content {
            data = body
        }

    val messageBody =
        Body {
            text = content
        }

    val subContent =
        Content {
            data = subjectVal
        }

    val message =
        Message {
            body = messageBody
            subject = subContent
        }

    val destinationOb =
        Destination {
            toAddresses = listOf(toAddressVal)
```

```
        }

    val emailContent =
        EmailContent {
            simple = message
        }

    val sendEmailRequest =
        SendEmailRequest {
            fromEmailAddress = senderAddress
            destination = destinationOb
            this.content = emailContent
        }

    PinpointEmailClient.fromEnvironment { region = "us-east-1" }.use
 { pinpointemail ->
        pinpointemail.sendEmail(sendEmailRequest)
        println("Message Sent")
    }
}
```

- Para obtener más información sobre la API, consulta SendMessagesla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enviar un mensaje de correo electrónico.

```
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()
```

Enviar un mensaje 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()
```

Enviar un mensaje de correo electrónico con una plantilla de correo electrónico existente.

```
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()
```

Enviar un mensaje de texto con una plantilla de SMS existente.

```
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()
```

- Para obtener más información sobre la API, consulta SendMessagesla AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enviar un mensaje de correo electrónico.

```
    " Build the addresses map from the list of to_addresses
    DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
    LOOP AT it_to_addresses INTO DATA(lo_address).
      INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
        key = lo_address->get_value( )
        value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
      ) INTO TABLE lt_addresses.
    ENDLOOP.

    " Send the email message
    DATA(lo_result) = lo_ppt->sendmessages(
```

```
      iv_applicationid = iv_app_id
      io_messagerequest = NEW /aws1/cl_pptmessagerequest(
        it_addresses = lt_addresses
        io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
          io_emailmessage = NEW /aws1/cl_pptemailmessage(
            iv_fromaddress = iv_sender
            io_simpleemail = NEW /aws1/cl_pptsimpleemail(
              io_subject = NEW /aws1/cl_pptsimpleemailpart(
                iv_charset = iv_char_set
                iv_data = iv_subject
              )
              io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
                iv_charset = iv_char_set
                iv_data = iv_html_message
              )
              io_textpart = NEW /aws1/cl_pptsimpleemailpart(
                iv_charset = iv_char_set
                iv_data = iv_text_message
              )
            )
          )
        )
      ).

    " Extract message IDs from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    ot_message_ids = lo_message_response->get_result( ).

    MESSAGE 'Email message sent successfully.' TYPE 'I'.
```

Enviar un mensaje SMS.

```
    " Build the addresses map for the destination number
    DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
    INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
      key = iv_destination_number
      value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
    ) INTO TABLE lt_addresses.

    " Send the SMS message
    DATA(lo_result) = lo_ppt->sendmessages(
```

```
      iv_applicationid = iv_app_id
      io_messagerequest = NEW /aws1/cl_pptmessagerequest(
        it_addresses = lt_addresses
        io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
          io_smsmessage = NEW /aws1/cl_pptsmsmessage(
            iv_body = iv_message
            iv_messagetype = iv_message_type
            iv_originationnumber = iv_origination_number
          )
        )
      )
    ).

    " Extract message ID from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    DATA(lt_results) = lo_message_response->get_result( ).
    LOOP AT lt_results INTO DATA(ls_result).
      IF ls_result-key = iv_destination_number.
        ov_message_id = ls_result-value->get_messageid( ).
        EXIT.
      ENDIF.
    ENDLOOP.

    MESSAGE 'SMS message sent successfully.' TYPE 'I'.
```

Enviar un mensaje de correo electrónico con una plantilla de correo electrónico existente.

```
    " Build the addresses map from the list of to_addresses
    DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
    LOOP AT it_to_addresses INTO DATA(lo_address).
      INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
        key = lo_address->get_value( )
        value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
      ) INTO TABLE lt_addresses.
    ENDLOOP.

    " Send the email message using a template
    DATA(lo_result) = lo_ppt->sendmessages(
      iv_applicationid = iv_app_id
      io_messagerequest = NEW /aws1/cl_pptmessagerequest(
        it_addresses = lt_addresses
        io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
```

```
        io_emailmessage = NEW /aws1/cl_pptemailmessage(
          iv_fromaddress = iv_sender
        )
      )
      io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
        io_emailtemplate = NEW /aws1/cl_ppttemplate(
          iv_name = iv_template_name
          iv_version = iv_template_version
        )
      )
    )
  ).

  " Extract message IDs from response
  DATA(lo_message_response) = lo_result->get_messageresponse( ).
  ot_message_ids = lo_message_response->get_result( ).

  MESSAGE 'Templated email message sent successfully.' TYPE 'I'.
```

Enviar un mensaje de texto con una plantilla de SMS existente.

```
  " Build the addresses map for the destination number
  DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = iv_destination_number
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
  ) INTO TABLE lt_addresses.

  " Send the SMS message using a template
  DATA(lo_result) = lo_ppt->sendmessages(
    iv_applicationid = iv_app_id
    io_messagerequest = NEW /aws1/cl_pptmessagerequest(
      it_addresses = lt_addresses
      io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
        io_smsmessage = NEW /aws1/cl_pptsmsmessage(
          iv_messagetype = iv_message_type
          iv_originationnumber = iv_origination_number
        )
      )
      io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
        io_smstemplate = NEW /aws1/cl_ppttemplate(
          iv_name = iv_template_name
```

```
            iv_version = iv_template_version
          )
        )
      )
    ).

    " Extract message ID from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    DATA(lt_results) = lo_message_response->get_result( ).
    LOOP AT lt_results INTO DATA(ls_result).
      IF ls_result-key = iv_destination_number.
        ov_message_id = ls_result-value->get_messageid( ).
        EXIT.
      ENDIF.
    ENDLOOP.

    MESSAGE 'Templated SMS message sent successfully.' TYPE 'I'.
```

- Para obtener más información sobre la API, consulte SendMessagesla referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`UpdateEndpoint`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateEndpoint`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
    }
}
```

- Para obtener más información sobre la API, consulta UpdateEndpointla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código para la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar la API de SMS y voz de Amazon Pinpoint con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

API de SMS y voz de Amazon Pinpoint

- Ejemplos básicos de la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs
  - Acciones para la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs
    - CreateConfigurationSetÚselo con un AWS SDK
    - CreateConfigurationSetEventDestinationÚselo con un AWS SDK
    - DeleteConfigurationSetÚselo con un AWS SDK
    - DeleteConfigurationSetEventDestinationÚselo con un AWS SDK
    - GetConfigurationSetEventDestinationsÚselo con un AWS SDK
    - ListConfigurationSetsÚselo con un AWS SDK
    - SendVoiceMessageÚselo con un AWS SDK
    - UpdateConfigurationSetEventDestinationÚselo con un AWS SDK

Ejemplos básicos de la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los aspectos básicos de la API de voz y SMS de Amazon Pinpoint con. AWS SDKs

Ejemplos

- Acciones para la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs

- CreateConfigurationSetÚselo con un AWS SDK
- CreateConfigurationSetEventDestinationÚselo con un AWS SDK
- DeleteConfigurationSetÚselo con un AWS SDK
- DeleteConfigurationSetEventDestinationÚselo con un AWS SDK
- GetConfigurationSetEventDestinationsÚselo con un AWS SDK
- ListConfigurationSetsÚselo con un AWS SDK
- SendVoiceMessageÚselo con un AWS SDK
- UpdateConfigurationSetEventDestinationÚselo con un AWS SDK

Acciones para la API de SMS y voz de Amazon Pinpoint mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de la API de voz y SMS de Amazon Pinpoint con. AWS SDKs Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la Referencia de la API de SMS y voz de Amazon Pinpoint.

Ejemplos

- CreateConfigurationSetÚselo con un AWS SDK
- CreateConfigurationSetEventDestinationÚselo con un AWS SDK
- DeleteConfigurationSetÚselo con un AWS SDK
- DeleteConfigurationSetEventDestinationÚselo con un AWS SDK
- GetConfigurationSetEventDestinationsÚselo con un AWS SDK
- ListConfigurationSetsÚselo con un AWS SDK
- SendVoiceMessageÚselo con un AWS SDK
- UpdateConfigurationSetEventDestinationÚselo con un AWS SDK

`CreateConfigurationSet`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateConfigurationSet`.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Create a new configuration set
        lo_pps->createconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        MESSAGE 'Configuration set created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateConfigurationSetla referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateConfigurationSetEventDestination`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateConfigurationSetEventDestination`.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Create event destination for the configuration set
        lo_pps->createconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
```

```
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateConfigurationSetEventDestinationla referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DeleteConfigurationSet`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteConfigurationSet`.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Delete the configuration set
        lo_pps->deleteconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        MESSAGE 'Configuration set deleted successfully.' TYPE 'I'.
```

```
    CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
      MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_not_found_ex.
    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteConfigurationSetla referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DeleteConfigurationSetEventDestination`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteConfigurationSetEventDestination`.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  TRY.
      " Delete the event destination
      lo_pps->deleteconfseteventdst(
```

```
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
        ).

        MESSAGE 'Event destination deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteConfigurationSetEventDestinationla referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`GetConfigurationSetEventDestinations`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `GetConfigurationSetEventDestinations`.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Get event destinations for the configuration set
        oo_result = lo_pps->getconfseteventdestinations(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        " Process the event destinations
        LOOP AT oo_result->get_eventdestinations( ) INTO DATA(lo_event_dest).
          DATA(lv_dest_name) = lo_event_dest->get_name( ).
          DATA(lv_enabled) = lo_event_dest->get_enabled( ).

          MESSAGE |Event destination: { lv_dest_name }, Enabled: { lv_enabled }|
 TYPE 'I'.

          " Check for CloudWatch Logs destination
          DATA(lo_cloudwatch_dest) = lo_event_dest->get_cloudwatchlogsdst( ).
          IF lo_cloudwatch_dest IS NOT INITIAL.
            DATA(lv_log_group_arn) = lo_cloudwatch_dest->get_loggrouparn( ).
            MESSAGE |  CloudWatch Logs destination: { lv_log_group_arn }| TYPE
 'I'.
          ENDIF.

          " Check for Kinesis Firehose destination
          DATA(lo_firehose_dest) = lo_event_dest->get_kinesisfirehosedst( ).
          IF lo_firehose_dest IS NOT INITIAL.
            DATA(lv_delivery_stream) = lo_firehose_dest-
>get_deliverystreamarn( ).
            MESSAGE |  Kinesis Firehose destination: { lv_delivery_stream }| TYPE
 'I'.
          ENDIF.
```

```
          " Check for SNS destination
          DATA(lo_sns_dest) = lo_event_dest->get_snsdestination( ).
          IF lo_sns_dest IS NOT INITIAL.
            DATA(lv_topic_arn) = lo_sns_dest->get_topicarn( ).
            MESSAGE |  SNS destination: { lv_topic_arn }| TYPE 'I'.
          ENDIF.
        ENDLOOP.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte GetConfigurationSetEventDestinationsla referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ListConfigurationSets`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `ListConfigurationSets`.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   TRY.
       " List all configuration sets
       oo_result = lo_pps->listconfigurationsets(
         iv_nexttoken = iv_next_token    " Optional: Token for pagination
         iv_pagesize = iv_page_size      " Optional: Number of results per page,
e.g., '10'
       ).

       " Process the configuration sets
       LOOP AT oo_result->get_configurationsets( ) INTO DATA(lo_config_set).
         DATA(lv_config_set_name) = lo_config_set->get_value( ).
         MESSAGE |Configuration set: { lv_config_set_name }| TYPE 'I'.
       ENDLOOP.

       " Check if there are more results
       DATA(lv_next_token) = oo_result->get_nexttoken( ).
       IF lv_next_token IS NOT INITIAL.
         MESSAGE |More results available. Next token: { lv_next_token }| TYPE
'I'.
       ENDIF.

     CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
       MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
       RAISE EXCEPTION lo_bad_request_ex.
     CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
       MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
       RAISE EXCEPTION lo_internal_error_ex.
     CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
       MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
       RAISE EXCEPTION lo_too_many_requests_ex.
   ENDTRY.
```

- Para obtener más información sobre la API, consulte ListConfigurationSetsla referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`SendVoiceMessage`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `SendVoiceMessage`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

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

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

```
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a
 list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";

    // The language to use when sending the message. For a list of supported
    // languages, see
    // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
    static final String languageCode = "en-US";

    // The content of the message. This example uses SSML to customize and
 control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
    static final String ssmlMessage = "<speak>This is a test message sent from "
            + "<emphasis>Amazon Pinpoint</emphasis> "
            + "using the <break strength='weak'/>AWS "
            + "SDK for Java. "
            + "<amazon:effect phonation='soft'>Thank "
            + "you for listening.</amazon:effect></speak>";

    public static void main(String[] args) {

        final String usage = """
                Usage:   <originationNumber> <destinationNumber>\s

                Where:
                  originationNumber - The phone number or short code that
 you specify has to be associated with your Amazon Pinpoint account. For best
 results, specify long codes in E.164 format (for example, +1-555-555-5654).
                  destinationNumber - The recipient's phone number.  For best
 results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).\s
                """;

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

```
            System.out.println(usage);
            System.exit(1);
        }
        String originationNumber = args[0];
        String destinationNumber = args[1];
        System.out.println("Sending a voice message");

        // Set the content type to application/json.
        List<String> listVal = new ArrayList<>();
        listVal.add("application/json");
        Map<String, List<String>> values = new HashMap<>();
        values.put("Content-Type", listVal);

        ClientOverrideConfiguration config2 =
ClientOverrideConfiguration.builder()
                .headers(values)
                .build();

        PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
                .overrideConfiguration(config2)
                .region(Region.US_EAST_1)
                .build();

        sendVoiceMsg(client, originationNumber, destinationNumber);
        client.close();
    }

    public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                                    String destinationNumber) {
        try {
            SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                    .languageCode(languageCode)
                    .text(ssmlMessage)
                    .voiceId(voiceName)
                    .build();

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

            SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()
                    .destinationPhoneNumber(destinationNumber)
```

```
                    .originationPhoneNumber(originationNumber)
                    .content(content)
                    .build();

            client.sendVoiceMessage(voiceMessageRequest);
            System.out.println("The message was sent successfully.");

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

- Para obtener más información sobre la API, consulta SendVoiceMessagela Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v2)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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"]);
  }
});
```

- Para obtener más información sobre la API, consulta SendVoiceMessagela Referencia AWS SDK para JavaScript de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your
 Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your
 Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
 message.
    :param ssml_message: The content of the message. This example uses SSML to
 control
                         certain aspects of the message, such as the volume and
 the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
```

```
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
        )
        raise
    else:
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
```

```
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


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

- Para obtener más información sobre la API, consulta SendVoiceMessagela AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Create SSML message type object with voice parameters
        DATA(lo_ssml_message) = NEW /aws1/cl_ppsssmlmessagetype(
          iv_languagecode = iv_language_code    " e.g., 'en-US'
          iv_voiceid = iv_voice_id              " e.g., 'Matthew'
          iv_text = iv_ssml_message             " SSML formatted message text
        ).

        " Create voice message content with the SSML message
        DATA(lo_content) = NEW /aws1/cl_ppsvoicemessagecont(
          io_ssmlmessage = lo_ssml_message
        ).

        " Send the voice message
        DATA(lo_result) = lo_pps->sendvoicemessage(
          iv_originationphonenumber = iv_origination_number  " e.g.,
'+12065550110'
```

```
          iv_callerid = iv_caller_id                          " e.g.,
'+12065550199'
          iv_destinationphonenumber = iv_destination_number  " e.g.,
'+12065550142'
          io_content = lo_content
        ).

        " Retrieve the message ID from the response
        ov_message_id = lo_result->get_messageid( ).

        MESSAGE 'Voice message sent successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte SendVoiceMessagela referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`UpdateConfigurationSetEventDestination`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateConfigurationSetEventDestination`.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Update the event destination
        lo_pps->updateconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination updated successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte UpdateConfigurationSetEventDestinationla referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Pinpoint con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Seguridad en Amazon Pinpoint

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de conformidad que se aplican a Amazon Pinpoint, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad AWS Servicios incluidos en el ámbito de aplicación por programa .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación lo ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon Pinpoint. En los siguientes temas, se mostrará cómo configurar Amazon Pinpoint para satisfacer los objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de Amazon Pinpoint.

Para obtener más información, consulte sobre las arquitecturas de referencia, consulte la Guía de arquitectura resiliente de Amazon Pinpoint.

Temas

- Protección de datos en Amazon Pinpoint
- Administración de identidades y accesos para Amazon Pinpoint
- Registro y monitoreo en Amazon Pinpoint
- Validación de la conformidad para Amazon Pinpoint
- Resiliencia de Amazon Pinpoint
- Seguridad de la infraestructura en Amazon Pinpoint

- Configuración y análisis de vulnerabilidades en Amazon Pinpoint
- Prácticas recomendadas de seguridad para Amazon Pinpoint

Protección de datos en Amazon Pinpoint

El modelo de se aplica a protección de datos en Amazon Pinpoint. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con Amazon Pinpoint u otro dispositivo Servicios de AWS mediante la consola, la API o. AWS CLI AWS SDKs Cualquier dato que introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

En función de cómo configure y utilice el servicio, Amazon Pinpoint puede almacenar los siguientes tipos de datos personales sobre usted o sobre los clientes:

Datos de configuración

Esto incluye datos de configuración del proyecto, como credenciales y parámetros que definen cómo y cuándo Amazon Pinpoint envía mensajes a través de los canales admitidos y los segmentos de usuario a los que envía mensajes. Para enviar mensajes, estos datos pueden incluir direcciones IP exclusivas para los mensajes de correo electrónico, códigos abreviados y el remitente IDs de los mensajes de texto SMS y credenciales para comunicarse con los servicios de notificaciones push, como el servicio de notificaciones push de Apple (APNs) y Firebase Cloud Messaging (FCM).

Datos de usuario y de punto de enlace

Entre estos se incluyen atributos estándar y personalizados que se utilizan para almacenar y administrar datos sobre usuarios y puntos de conexión de un proyecto de Amazon Pinpoint. Un atributo puede almacenar información sobre un usuario específico (como el nombre de un usuario) o un punto de enlace específico de un usuario (como la dirección de correo electrónico de un usuario, el número de teléfono móvil o el token de dispositivo móvil). Estos datos también pueden incluir usuarios externos IDs que correlacionan a los usuarios de un proyecto de Amazon Pinpoint con los usuarios de un sistema externo, como un sistema de gestión de relaciones con los clientes. Para obtener más información acerca de lo que pueden incluir estos datos, consulte los esquemas de Usuario y Punto de conexión en la Referencia de la API de Amazon Pinpoint.

Datos de análisis

Esto incluye datos de métricas, también denominados indicadores clave de rendimiento (KPIs), que proporcionan información sobre el rendimiento de un proyecto de Amazon Pinpoint en áreas como la participación de los usuarios y la actividad de compra. Entre estos se incluyen también los datos de métricas que proporcionan información sobre la demografía de los usuarios de

un proyecto. Los datos pueden proceder de atributos estándar y personalizados para usuarios y puntos de enlace, como la ciudad en la que vive un usuario. También pueden proceder de eventos, como eventos de apertura y clic de los mensajes de correo electrónico que envía para un proyecto.

Datos importados

Incluyen todos los datos de usuario, segmentación y análisis que agregue o importe desde orígenes externos y utilice en Amazon Pinpoint. Un ejemplo es un archivo JSON que importe en Amazon Pinpoint (directamente a través de la consola o desde un bucket de Amazon S3) para crear un segmento estático. Otros ejemplos son los datos de punto de conexión que se agregan mediante programación para crear un segmento dinámico, las direcciones de punto de conexión a las que se envían mensajes directos y los eventos que se configuran en una aplicación para informar a Amazon Pinpoint.

Temas

- Cifrado de datos
- Privacidad del tráfico entre redes
- Creación de un punto de conexión de VPC de interfaz para Amazon Pinpoint

Cifrado de datos

Los datos de Amazon Pinpoint se cifran en tránsito y en reposo. Cuando envía datos a Amazon Pinpoint, el servicio cifra los datos a medida que los recibe y los almacena. Cuando se recuperan datos de Amazon Pinpoint, el servicio los transmite mediante los protocolos de seguridad actuales.

Cifrado en reposo

Amazon Pinpoint cifra todos los datos que almacena para usted. Entre estos se incluyen los datos de configuración, los datos de usuario y punto de conexión, los datos de análisis y los datos que agrega o importa a Amazon Pinpoint. Para cifrar sus datos, Amazon Pinpoint utiliza claves AWS Key Management Service internas AWS KMS() que el servicio posee y mantiene en su nombre. Rotamos estas claves periódicamente. Para obtener información al respecto AWS KMS, consulte la Guía para AWS Key Management Service desarrolladores.

Cifrado en tránsito

Amazon Pinpoint utiliza HTTPS y la seguridad de la capa de transporte (TLS) 1.2 o posterior para comunicarse con los clientes y aplicaciones. Para comunicarse con otros AWS servicios, Amazon Pinpoint usa HTTPS y TLS 1.2. Además, al crear y administrar los recursos de Amazon Pinpoint mediante la consola, un AWS SDK o el AWS Command Line Interface, todas las comunicaciones se protegen mediante HTTPS y TLS 1.2.

Administración de claves

Para cifrar sus datos de Amazon Pinpoint, Amazon Pinpoint utiliza claves AWS KMS internas que el servicio posee y mantiene en su nombre. Rotamos estas claves periódicamente. No puede aprovisionar ni utilizar claves propias AWS KMS ni de otro tipo para cifrar los datos que almacene en Amazon Pinpoint.

Privacidad del tráfico entre redes

La privacidad del tráfico entre redes se refiere a la protección de las conexiones y el tráfico entre Amazon Pinpoint y sus clientes y aplicaciones locales, y entre Amazon Pinpoint y AWS otros recursos de la misma región. AWS Las siguientes características y prácticas pueden ayudarle a garantizar la privacidad del tráfico entre redes en Amazon Pinpoint.

Tráfico entre Amazon Pinpoint y aplicaciones y clientes en las instalaciones

Para establecer una conexión privada entre Amazon Pinpoint y clientes y aplicaciones en la red en las instalaciones, puede utilizar Direct Connect. Esto le permite vincular su red a una ubicación de AWS Direct Connect mediante un cable de Ethernet de fibra óptica estándar. Un extremo del cable se conecta al enrutador. El otro extremo está conectado a un router. Direct Connect Para obtener más información, consulte ¿Qué es Direct Connect? en la Guía del usuario de Direct Connect .

Para ayudar a proteger el acceso a Amazon Pinpoint a través de las publicaciones APIs, le recomendamos que cumpla con los requisitos de Amazon Pinpoint para las llamadas a la API. Amazon Pinpoint requiere que los clientes utilicen la seguridad de la capa de transporte (TLS) 1.2 o una versión posterior. Los clientes también deben admitir conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben firmarse con un identificador de clave de acceso y una clave de acceso secreta que estén asociadas al principal AWS Identity and Access Management (IAM) de

su cuenta. AWS También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Tráfico entre Amazon Pinpoint y otros recursos AWS

Para proteger las comunicaciones entre Amazon Pinpoint y otros AWS recursos de la misma AWS región, Amazon Pinpoint utiliza HTTPS y TLS 1.2 de forma predeterminada.

Creación de un punto de conexión de VPC de interfaz para Amazon Pinpoint

Puede establecer una conexión privada entre la nube privada virtual (VPC) y un punto de conexión en Amazon Pinpoint creando un punto de conexión de VPC de interfaz.

Los puntos de enlace de la interfaz funcionan con una tecnología que le permite acceder de forma privada a Amazon APIs Pinpoint sin una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o. AWS PrivateLink Direct Connect Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con el Amazon APIs Pinpoint con el que se integran. AWS PrivateLink

Para obtener más información, consulte la Guía de AWS PrivateLink.

Creación de puntos de conexión de VPC de una interfaz

Puede crear un punto final de interfaz mediante la consola de Amazon VPC o con AWS Command Line Interface ()AWS CLI. Para obtener más información, consulte Crear un punto final de interfaz en la AWS PrivateLink guía.

Amazon Pinpoint admite los siguientes nombres de servicio:

- `com.amazonaws.region.pinpoint`
- `com.amazonaws.region.pinpoint-sms-voice-v2`

Si activa el DNS privado para un punto final de la interfaz, puede realizar solicitudes de API a Amazon Pinpoint utilizando el nombre de DNS predeterminado para, por ejemplo Región de AWS,. `com.amazonaws.us-east-1.pinpoint` Para obtener más información, consulte nombres de host DNS en la Guía del usuario de AWS PrivateLink .

Para obtener una lista de todas las regiones y puntos de conexión en los que Amazon Pinpoint se encuentra actualmente disponible, consulte puntos de conexión de los servicios de AWS en Referencia general de Amazon Web Services.

Creación de una política de punto de conexión de VPC

Puede asociar una política de punto de conexión con el punto de conexión de VPC que controla el acceso. La política especifica la siguiente información:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para obtener más información, consulte Control del acceso a los servicios con políticas de punto de conexión en la Guía del usuario de AWS PrivateLink .

Ejemplo: Política de punto de conexión de VPC

La siguiente política de punto de conexión de VPC concede acceso a las acciones de Amazon Pinpoint mostradas para todas las entidades principales en todos los recursos.

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

Administración de identidades y accesos para Amazon Pinpoint

AWS Identity and Access Management (IAM) es un sistema Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de

IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon Pinpoint. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración del acceso con políticas
- Cómo funciona Amazon Pinpoint con IAM
- Acciones de Amazon Pinpoint para las políticas de IAM
- Ejemplos de políticas basadas en identidades de Amazon Pinpoint
- Roles de IAM para tareas comunes de Amazon Pinpoint
- Solución de problemas de la administración de identidades y accesos de Amazon Pinpoint

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte Solución de problemas de la administración de identidades y accesos de Amazon Pinpoint).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte Cómo funciona Amazon Pinpoint con IAM).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte Ejemplos de políticas basadas en identidades de Amazon Pinpoint).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las

credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte Cómo iniciar sesión en la Cuenta de AWS en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte AWS Signature Version 4 para solicitudes de API en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear un Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz que tiene acceso completo a todos Servicios de AWS los recursos. Se recomiendaencarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver la lista completa de las tareas que requieren credenciales de usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Usuarios y grupos de IAM

Un usuario de IAM es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales en la Guía del usuario de IAM.

Un grupo de IAM especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un Rol de IAM es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol cambiando de un rol de usuario a uno de IAM (consola) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte Información general de políticas de JSON en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de política de permisos JSON que asocia a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Las políticas basadas en identidad pueden ser políticas insertadas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte Selección entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Amazon Pinpoint admite el uso de políticas basadas en la identidad para controlar el acceso a los recursos de Amazon Pinpoint.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe especificar una entidad principal en una política basada en recursos.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Amazon Pinpoint admite el uso de políticas basadas en recursos para controlar el acceso a los recursos de Amazon Pinpoint.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Amazon Pinpoint no admite el uso de ACLs para controlar el acceso a los recursos de Amazon Pinpoint.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte Políticas de control de servicios en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCPs): establece los permisos máximos disponibles para los recursos de tus cuentas. Para obtener más información, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Amazon Pinpoint admite el uso de estos tipos de políticas para controlar el acceso a los recursos de Amazon Pinpoint.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon Pinpoint con IAM

Para usar Amazon Pinpoint, los usuarios de su AWS cuenta necesitan permisos que les permitan ver datos de análisis, crear proyectos, definir segmentos de usuarios, implementar campañas y mucho más. Si integra una aplicación móvil o web con Amazon Pinpoint, los usuarios de la aplicación también requieren acceso a Amazon Pinpoint. Este acceso permite que la aplicación registre los datos de uso y punto de conexión en Amazon Pinpoint. Para conceder acceso a las funciones de Amazon Pinpoint, cree políticas AWS Identity and Access Management (IAM) que permitan a Amazon Pinpoint realizar acciones para las identidades de IAM o los recursos de Amazon Pinpoint.

La IAM es un servicio que ayuda a los administradores a controlar de forma segura el acceso a los recursos. AWS Las políticas de IAM incluyen instrucciones que permiten o deniegan acciones específicas de usuarios específicos o para recursos específicos. Amazon Pinpoint proporciona un conjunto de acciones que puede usar en políticas de IAM para especificar permisos detallados para los usuarios y recursos de Amazon Pinpoint. Esto significa que puede conceder el nivel adecuado de acceso a Amazon Pinpoint sin necesidad de crear políticas demasiado permisivas, algo que podría dejar expuestos datos importantes o poner en peligro los recursos. Por ejemplo, puede conceder acceso sin restricciones a un administrador de Amazon Pinpoint y conceder acceso de solo lectura a las personas que solo necesiten acceso a un proyecto específico.

Antes de utilizar IAM para administrar el acceso a Amazon Pinpoint, debe conocer qué características de IAM se encuentran disponibles con Amazon Pinpoint. Para obtener una visión general de cómo Amazon Pinpoint y otros AWS servicios funcionan con IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Temas

- Políticas basadas en identidades de Amazon Pinpoint
- Políticas de permisos basadas en recursos de Amazon Pinpoint
- Autorización basada en etiquetas de Amazon Pinpoint
- Roles de IAM de Amazon Pinpoint

Políticas basadas en identidades de Amazon Pinpoint

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Amazon Pinpoint admite acciones, claves de condiciones y recursos específicos. Para obtener más información acerca de los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Acciones

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Esto significa que las acciones de la política controlan lo que los usuarios pueden hacer en la consola de Amazon Pinpoint. También controlan lo que los usuarios pueden hacer mediante programación mediante el uso directo de AWS SDKs, the AWS Command Line Interface (AWS CLI) o Amazon APIs Pinpoint.

Las acciones de política de Amazon Pinpoint utilizan los prefijos siguientes:

- **`mobiletargeting`**: para acciones derivadas de la API de Amazon Pinpoint, que es la API principal de Amazon Pinpoint.
- **`sms-voice`**: para acciones derivadas de la API de SMS y voz de Amazon Pinpoint, que es una API complementaria que ofrece opciones avanzadas para usar y administrar los canales de SMS y voz en Amazon Pinpoint.

Por ejemplo, para conceder a alguien permiso para ver información sobre todos los segmentos de un proyecto, que es una acción que corresponde a la operación `GetSegments` de la API de Amazon Pinpoint, incluya la acción `mobiletargeting:GetSegments` en la política. Las instrucciones de la política deben incluir un elemento `Action` o un elemento `NotAction`. Amazon Pinpoint define su propio conjunto de acciones que describen las tareas que se pueden realizar con él.

Para especificar varias acciones en una única instrucción, sepárelas con comas:

```
"Action": [
```

```
      "mobiletargeting:action1",
      "mobiletargeting:action2"
```

También puede utilizar caracteres comodín (*) para especificar varias acciones. Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Get`, incluya la siguiente acción:

```
"Action": "mobiletargeting:Get*"
```

Sin embargo, recomendamos que las políticas se creen según el principio de privilegios mínimos. En otras palabras, debe crear políticas que incluyan solo los permisos necesarios para realizar una acción específica.

Para obtener una lista de las acciones de Amazon Pinpoint que puede utilizar en las políticas de IAM, consulte Acciones de Amazon Pinpoint para las políticas de IAM.

Recursos

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). En el caso de las acciones que no admiten permisos por recurso, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Por ejemplo, la acción `mobiletargeting:GetSegments` recupera información sobre todos los segmentos asociados a un proyecto de Amazon Pinpoint específico. Identifica un proyecto con un ARN con el siguiente formato:

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

Para obtener más información sobre el formato de ARNs, consulte Amazon Resource Names (ARNs) en Referencia general de AWS.

En las políticas de IAM, puede especificar ARNs los siguientes tipos de recursos de Amazon Pinpoint:

- Campaigns (Campañas)
- Journeys
- Plantillas de mensajes (denominadas plantillas en algunos contextos)
- Proyectos (denominados apps o aplicaciones en algunos contextos)
- Modelos de recomendación (denominados recomendadores en algunos contextos)
- Segmentos

Por ejemplo, para crear una instrucción de política para el proyecto con el ID de proyecto `810c7aab86d42fb2b56c8c966example`, utilice el siguiente ARN:

```
"Resource": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

Para especificar todos los proyectos que pertenecen a una cuenta específica, utilice el carácter comodín (*):

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

Algunas acciones de Amazon Pinpoint, como determinadas acciones para crear recursos, no se pueden llevar a cabo en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*):

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

En las políticas de IAM, también puede especificar ARNs los siguientes tipos de recursos de SMS y voz de Amazon Pinpoint:

- Conjunto de configuraciones
- Lista de exclusión
- Número de teléfono
- Grupo
- ID de remitente

Por ejemplo, para crear una instrucción de política para un número de teléfono que tenga el ID de número de teléfono `phone-12345678901234567890123456789012` utilice el siguiente ARN:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-1234567890123456789012345678901 2"
```

Para especificar todos los números de teléfono que pertenezcan a una cuenta específica, utilice un comodín (*) en lugar del ID del número de teléfono:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

Algunas acciones de SMS y voz de Amazon Pinpoint no se realizan en un recurso específico, como las que se utilizan para administrar ajustes a nivel de cuenta, como los límites de gasto. En dichos casos, debe utilizar el carácter comodín (*):

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

Algunas acciones de la API de Amazon Pinpoint implican varios recursos. Por ejemplo, la acción `TagResource` puede agregar una etiqueta a varios proyectos. Para especificar varios recursos en una sola declaración, sepárelos ARNs con comas:

```
"Resource": [
      "resource1",
      "resource2"
```

Para ver una lista de los tipos de recursos de Amazon Pinpoint y sus tipos ARNs, consulte Recursos definidos por Amazon Pinpoint en la Guía del usuario de IAM. Para obtener información acerca de las acciones con las que puede especificar el ARN de cada tipo de recurso, consulte Acciones definidas por Amazon Pinpoint en la Guía del usuario de IAM.

Claves de condición

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Amazon Pinpoint define su propio conjunto de claves de condición y también admite algunas claves de condición globales. Para ver una lista de todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM. Para consultar una lista de claves de condición de Amazon Pinpoint, consulte Claves de condición para Amazon Pinpoint en la Guía del usuario de IAM. Para obtener información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon Pinpoint en la Guía del usuario de IAM.

Ejemplos

Para ver ejemplos de políticas basadas en identidad de Amazon Pinpoint, consulte Ejemplos de políticas basadas en identidades de Amazon Pinpoint.

Políticas de permisos basadas en recursos de Amazon Pinpoint

Las políticas de permisos basadas en recursos son documentos de políticas de JSON que especifican qué acciones puede realizar una entidad principal de servicio especificada en un recurso de Amazon Pinpoint y en qué condiciones. Amazon Pinpoint admite políticas de permisos basadas en recursos para campañas, recorridos, plantillas de mensajes (plantillas), modelos de recomendaciones (generadores de recomendaciones), proyectos (aplicaciones) y segmentos.

Ejemplos

Para ver ejemplos de políticas basadas en recursos de Amazon Pinpoint, consulte the section called “Ejemplos de políticas basadas en identidades”.

Autorización basada en etiquetas de Amazon Pinpoint

Puede asociar etiquetas a determinados tipos de recursos de Amazon Pinpoint o pasar etiquetas en una solicitud a Amazon Pinpoint. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/${TagKey}`, `aws:RequestTag/${TagKey}` o `aws:TagKeys`.

Para obtener información sobre cómo etiquetar los recursos de Amazon Pinpoint, incluida una política de IAM de ejemplo, consulte Administración de etiquetas de recurso de Amazon Pinpoint.

Roles de IAM de Amazon Pinpoint

Un rol de IAM es una entidad de la cuenta de AWS que dispone de permisos específicos.

Uso de credenciales temporales con Amazon Pinpoint

Puede utilizar credenciales temporales para iniciar sesión con identidad federada, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se obtienen llamando a AWS Security Token Service (AWS STS) operaciones de API como AssumeRoleo GetFederationToken.

Amazon Pinpoint admite el uso de credenciales temporales.

Roles vinculados a servicios

Los roles vinculados a un servicio permiten a AWS los servicios acceder a los recursos de otros servicios para completar una acción en tu nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Amazon Pinpoint no usa roles vinculados a servicios.

Roles de servicio

Esta característica permite que un servicio asuma un rol de servicio en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su cuenta de IAM y son propiedad de la cuenta. Esto significa que un administrador de IAM puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

Amazon Pinpoint admite el uso de roles de servicio.

Acciones de Amazon Pinpoint para las políticas de IAM

Para gestionar el acceso a los recursos de Amazon Pinpoint de su AWS cuenta, puede añadir acciones de Amazon Pinpoint AWS Identity and Access Management a las políticas (IAM). Mediante las acciones de las políticas, puede controlar lo que los usuarios pueden hacer en la consola de Amazon Pinpoint. También puede controlar lo que los usuarios pueden hacer mediante programación utilizando directamente el AWS SDKs, el AWS Command Line Interface (AWS CLI) o el Amazon APIs Pinpoint.

En una política, cada acción se especifica con el espacio de nombres de Amazon Pinpoint correspondiente seguido de dos puntos y el nombre de la acción, como `GetSegments`. La mayoría de las acciones se corresponden con una solicitud a la API de Amazon Pinpoint mediante un método

URI y HTTP específico. Por ejemplo, si permite la acción `mobiletargeting:GetSegments` en la política de un usuario, el usuario puede recuperar información sobre todos los segmentos de un proyecto enviando una solicitud HTTP GET al URI `/apps/projectId/segments`. Esta política también permite al usuario ver esa información en la consola y recuperarla mediante un AWS SDK o el. AWS CLI

Cada acción se realiza en un recurso de Amazon Pinpoint específico, que se identifica en una declaración de una política por su nombre de recurso de Amazon (ARN). Por ejemplo, la acción `mobiletargeting:GetSegments` se realiza en un proyecto específico, que se identifica con el ARN, `arn:aws:mobiletargeting:region:accountId:apps/projectId`.

En este tema se identifican las acciones de Amazon Pinpoint que puede agregar a políticas de IAM para la cuenta de AWS . Para ver ejemplos que muestren cómo utilizar las acciones de las políticas para administrar el acceso a los recursos de Amazon Pinpoint, consulte Ejemplos de políticas basadas en identidades de Amazon Pinpoint.

Temas

- Acciones de la API de Amazon Pinpoint
- Acciones de la API de SMS y voz de Amazon Pinpoint, versión 1

Acciones de la API de Amazon Pinpoint

En esta sección se identifican las acciones para las características que están disponibles en la API de Amazon Pinpoint, que es la API principal de Amazon Pinpoint. Para obtener más información sobre esta API, consulte la referencia de la API de Amazon Pinpoint.

Categorías:

- Análisis y métricas
- Campaigns (Campañas)
- Canales
- Puntos de conexión
- Secuencias de eventos
- Events (Eventos)
- Trabajos de exportación
- Trabajos de importación
- Journeys

- Plantillas de mensaje
- Mensajes
- Contraseñas únicas
- Validación de números de teléfono
- Proyectos
- Modelos de recomendación
- Segmentos
- Tags
- Users

Análisis y métricas

Los siguientes permisos están relacionados con la visualización de datos de análisis en la consola de Amazon Pinpoint. También están relacionadas con la recuperación (consulta) de datos agregados para obtener métricas estándar, también conocidas como indicadores clave de rendimiento (KPIs), que se aplican a proyectos, campañas y procesos.

`mobiletargeting:GetReports`

Consulte datos de análisis en la consola de Amazon Pinpoint. Este permiso también es necesario para crear segmentos que contengan atributos personalizados mediante la consola de Amazon Pinpoint. También es necesario obtener una estimación del tamaño de un segmento en la consola de Amazon Pinpoint.

- URI: No aplicable.
- Método: No aplicable.
- ARN del recurso: `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:*`

`mobiletargeting:GetApplicationDateRangeKpi`

Permite recuperar (consultar) datos acumulados de una métrica de aplicación estándar. Es una métrica que se aplica a todas las campañas o mensajes transaccionales asociados con un proyecto.

- URI: `/apps/`*`projectId`*`/kpis/daterange/`*`kpi-name`*
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:apps/`*`projectId`*`/kpis/daterange/`*`kpi-name`*

mobiletargeting:GetCampaignDateRangeKpi

Permite recuperar (consultar) datos acumulados de una métrica de campaña estándar. Es una métrica que se aplica a una campaña individual.

- URI: /apps/*projectId*/campaigns/*campaignId*/kpis/daterange/*kpi-name*
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/campaigns/`*campaignId*`/kpis/daterange/`*kpi-name*

mobiletargeting:GetJourneyDateRangeKpi

Permite recuperar (consultar) datos acumulados de una métrica de interacción del recorrido estándar. Se trata de una métrica de participación que se aplica a un recorrido individual, por ejemplo, la cantidad de mensajes que abrieron los participantes para todas las actividades de un recorrido.

- URI: /apps/*projectId*/journeys/*journeyId*/kpis/daterange/*kpi-name*
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/journeys/`*journeyId*`/kpis/daterange/`*kpi-name*

mobiletargeting:GetJourneyExecutionMetrics

Recupere (consulta) datos agregados para obtener métricas de ejecución estándar que se aplican a un recorrido individual, por ejemplo, el número de participantes que realizan activamente todas las actividades de un recorrido.

- URI: /apps/*projectId*/journeys/*journeyId*/execution-metrics
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/journeys/`*journeyId*`/execution-metrics`

mobiletargeting:GetJourneyExecutionActivityMetrics

Recupere (consulta) datos agregados para obtener métricas de ejecución estándar que se aplican a una actividad individual en un recorrido, por ejemplo, el número de participantes que empezaron o completaron una actividad.

- URI: /apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics
- Método: GET

- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/activities/journey-activity-id/execution-metrics`

Campaigns (Campañas)

Los siguientes permisos están relacionados con la administración de campañas en la cuenta de Amazon Pinpoint.

`mobiletargeting:CreateCampaign`

Permite crear una campaña para un proyecto.

- URI: `/apps/projectId/campaigns`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns`

`mobiletargeting:DeleteCampaign`

Elimine una campaña específica.

- URI: `/apps/projectId/campaigns/campaignId`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:GetCampaign`

Recupere información acerca de una campaña específica.

- URI: `/apps/projectId/campaigns/campaignId`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:GetCampaignActivities`

Recupere información acerca de las actividades realizadas por una campaña.

- URI: `/apps/projectId/campaigns/campaignId/activities`
- Método: GET

- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:GetCampaigns`

Permite recuperar información acerca de todas las campañas de un proyecto.

- URI: `/apps/projectId/campaigns`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetCampaignVersion`

Recupere información acerca de la versión de una campaña específica.

- URI: `/apps/projectId/campaigns/campaignId/versions/versionId`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:GetCampaignVersions`

Recupere información acerca de las versiones actual y anteriores de una campaña.

- URI: `/apps/projectId/campaigns/campaignId/versions`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:UpdateCampaign`

Actualice una campaña específica.

- URI: `/apps/projectId/campaigns/campaignId`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

Canales

Los siguientes permisos están relacionados con la administración de canales en la cuenta de Amazon Pinpoint. En Amazon Pinpoint, al hablar de canales, se hace referencia a los métodos que

se utilizan para contactar con los clientes, como el envío de correo electrónico, mensajes SMS o notificaciones de inserción.

`mobiletargeting:DeleteAdmChannel`

Permite deshabilitar el canal de Amazon Device Messaging (ADM) de un proyecto.

- URI: `/apps/projectId/channels/adm`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

`mobiletargeting:GetAdmChannel`

Permite recuperar información sobre el canal de ADM de un proyecto.

- URI: `/apps/projectId/channels/adm`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

`mobiletargeting:UpdateAdmChannel`

Permite habilitar o actualizar el canal de ADM de un proyecto.

- URI: `/apps/projectId/channels/adm`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

`mobiletargeting:DeleteApnsChannel`

Inhabilita el canal del servicio de notificaciones push de Apple (APNs) para un proyecto.

- URI: `/apps/projectId/channels/apns`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

`mobiletargeting:GetApnsChannel`

Recupera información sobre el APNs canal de un proyecto.

- URI: `/apps/projectId/channels/apns`

- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

`mobiletargeting:UpdateApnsChannel`

Habilita o actualiza el APNs canal de un proyecto.

- URI: `/apps/projectId/channels/apns`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

`mobiletargeting:DeleteApnsSandboxChannel`

Deshabilite el canal APNs sandbox para un proyecto.

- URI: `/apps/projectId/channels/apns_sandbox`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:GetApnsSandboxChannel`

Recupera información sobre el canal APNs sandbox de un proyecto.

- URI: `/apps/projectId/channels/apns_sandbox`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:UpdateApnsSandboxChannel`

Habilite o actualice el canal APNs sandbox de un proyecto.

- URI: `/apps/projectId/channels/apns_sandbox`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:DeleteApnsVoipChannel`

Deshabilita el canal APNs VoIP para un proyecto.

- URI: /apps/*projectId*/channels/apns_voip
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip`

`mobiletargeting:GetApnsVoipChannel`

Recupera información sobre el canal APNs VoIP de un proyecto.

- URI: /apps/*projectId*/channels/apns_voip
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip`

`mobiletargeting:UpdateApnsVoipChannel`

Habilita o actualiza el canal APNs VoIP de un proyecto.

- URI: /apps/*projectId*/channels/apns_voip
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip`

`mobiletargeting:DeleteApnsVoipSandboxChannel`

Deshabilite el canal sandbox de APNs VoIP para un proyecto.

- URI: /apps/*projectId*/channels/apns_voip_sandbox
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip_sandbox`

`mobiletargeting:GetApnsVoipSandboxChannel`

Recupera información sobre el canal sandbox de APNs VoIP de un proyecto.

- URI: /apps/*projectId*/channels/apns_voip_sandbox
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip_sandbox`

`mobiletargeting:UpdateApnsVoipSandboxChannel`

Habilita o actualiza el canal sandbox de APNs VoIP para un proyecto.

- URI: /apps/*projectId*/channels/apns_voip_sandbox
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip_sandbox`

`mobiletargeting:DeleteBaiduChannel`

Permite deshabilitar el canal de Baidu Cloud Push de un proyecto.

- URI: /apps/*projectId*/channels/baidu
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/baidu`

`mobiletargeting:GetBaiduChannel`

Permite recuperar información sobre el canal de Baidu Cloud Push de un proyecto.

- URI: /apps/*projectId*/channels/baidu
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/baidu`

`mobiletargeting:UpdateBaiduChannel`

Permite habilitar o actualizar el canal de Baidu Cloud Push para un proyecto.

- URI: /apps/*projectId*/channels/baidu
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/baidu`

`mobiletargeting:DeleteEmailChannel`

Permite deshabilitar el canal de correo electrónico de un proyecto

- URI: /apps/*projectId*/channels/email
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/email`

`mobiletargeting:GetEmailChannel`

Permite recuperar información sobre el canal de correo electrónico de un proyecto.

- URI: /apps/*projectId*/channels/email
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/email`

`mobiletargeting:UpdateEmailChannel`

Permite habilitar o actualizar el canal de correo electrónico de un proyecto.

- URI: /apps/*projectId*/channels/email
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/email`

`mobiletargeting:DeleteGcmChannel`

Desactive el canal de Firebase Cloud Messaging (FCM) para un proyecto. Este canal permite a Amazon Pinpoint enviar notificaciones de inserción a una aplicación de Android a través del servicio FCM, que sustituye al servicio Google Cloud Messaging (GCM).

- URI: /apps/*projectId*/channels/gcm
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/gcm`

`mobiletargeting:GetGcmChannel`

Permite recuperar información acerca del canal de FCM de un proyecto. Este canal permite a Amazon Pinpoint enviar notificaciones de inserción a una aplicación de Android a través del servicio FCM, que sustituye al servicio Google Cloud Messaging (GCM).

- URI: /apps/*projectId*/channels/gcm
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/gcm`

`mobiletargeting:UpdateGcmChannel`

Habilite o actualice el canal de FCM de un proyecto. Este canal permite a Amazon Pinpoint enviar notificaciones de inserción a una aplicación de Android a través del servicio FCM, que sustituye al servicio Google Cloud Messaging (GCM).

- URI: /apps/*projectId*/channels/gcm

- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

`mobiletargeting:DeleteSmsChannel`

Permite deshabilitar el canal de SMS de un proyecto

- URI: `/apps/projectId/channels/sms`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

`mobiletargeting:GetSmsChannel`

Permite recuperar información acerca del canal de SMS de un proyecto.

- URI: `/apps/projectId/channels/sms`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

`mobiletargeting:UpdateSmsChannel`

Permite habilitar o actualizar el canal de SMS de un proyecto.

- URI: `/apps/projectId/channels/sms`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

`mobiletargeting:GetChannels`

Recupera información sobre el historial y el estado de cada canal de una aplicación.

- URI: `/apps/application-id/channels`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels`

`mobiletargeting:DeleteVoiceChannel`

Desactiva el canal de voz de una aplicación y elimina cualquier configuración existente para el canal.

- URI: /apps/*application-id*/channels/voice
- Método: DELETE
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:GetVoiceChannel

Recupera información sobre el estado y la configuración del canal de voz para una aplicación.

- URI: /apps/*application-id*/channels/voice
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:UpdateVoiceChannel

Habilita el canal de voz de una aplicación o actualiza el estado y la configuración del canal de voz de una aplicación.

- URI: /apps/*application-id*/channels/voice
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

Puntos de conexión

Los siguientes permisos están relacionados con la administración de puntos de conexión en la cuenta de Amazon Pinpoint. En Amazon Pinpoint, un punto de conexión constituye un destino único para los mensajes. Por ejemplo, un punto de enlace podría ser una dirección de correo electrónico del cliente, un número de teléfono o un token de dispositivo.

mobiletargeting:DeleteEndpoint

Eliminar un punto de conexión.

- URI: /apps/*projectId*/endpoints/*endpointId*
- Método: DELETE
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:GetEndpoint

Recupere información sobre un punto de enlace específico.

- URI: /apps/*projectId*/endpoints/*endpointId*
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:RemoveAttributes

Elimina uno o más atributos, del mismo tipo de atributo, de todos los puntos de conexión asociados a una aplicación.

- URI: apps/*application-id*/attributes/*attribute-type*
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/attributes/*attribute-type*

mobiletargeting:UpdateEndpoint

Cree un punto de enlace o actualice la información sobre uno.

- URI: /apps/*projectId*/endpoints/*endpointId*
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:UpdateEndpointsBatch

Cree o actualice puntos de enlace como una operación por lotes.

- URI: /apps/*projectId*/endpoints
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

Secuencias de eventos

Los siguientes permisos están relacionados con la administración de secuencias de eventos para la cuenta de Amazon Pinpoint.

mobiletargeting:DeleteEventStream

Permite eliminar la secuencia de eventos de un proyecto.

- URI: /apps/*projectId*/eventstream/
- Método: DELETE
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:GetEventStream

Permite recuperar información sobre la secuencia de eventos de un proyecto.

- URI: /apps/*projectId*/eventstream/
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:PutEventStream

Permite crear o actualizar una secuencia de eventos de un proyecto.

- URI: /apps/*projectId*/eventstream/
- Método: POST
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

Events (Eventos)

Los siguientes permisos están relacionados con la administración de trabajos de eventos en la cuenta de Amazon Pinpoint. En Amazon Pinpoint, se crean trabajos de importación para crear segmentos en función de las definiciones de punto de conexión almacenadas en un bucket de Amazon S3.

mobiletargeting:PutEvents

Crea un nuevo evento para registrarlo para los puntos de conexión o crea o actualiza los datos de los puntos de conexión a los que están asociados los eventos existentes.

- URI: /apps/*application-id*/events
- Método: POST

- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/events`

Trabajos de exportación

Los siguientes permisos están relacionados con la administración de trabajos de exportación en la cuenta de Amazon Pinpoint. En Amazon Pinpoint, se crean trabajos de exportación para enviar información sobre los puntos de conexión a un bucket de Amazon S3 para almacenamiento o análisis.

`mobiletargeting:CreateExportJob`

Cree un trabajo de exportación para exportar definiciones de puntos de conexión a Amazon S3.

- URI: `/apps/projectId/jobs/export`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

`mobiletargeting:GetExportJob`

Permite recuperar información sobre un trabajo de exportación específico de un proyecto.

- URI: `/apps/projectId/jobs/export/jobId`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export/jobId`

`mobiletargeting:GetExportJobs`

Permite recuperar una lista de todos los trabajos de exportación de un proyecto.

- URI: `/apps/projectId/jobs/export`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

Trabajos de importación

Los siguientes permisos están relacionados con la administración de trabajos de importación en la cuenta de Amazon Pinpoint. En Amazon Pinpoint, se crean trabajos de importación para crear

segmentos en función de las definiciones de punto de conexión almacenadas en un bucket de Amazon S3.

`mobiletargeting:CreateImportJob`

Importe definiciones de punto de conexión desde Amazon S3 para crear un segmento.

- URI: `/apps/projectId/jobs/import`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetImportJob`

Permite recuperar información sobre un trabajo de importación específico de un proyecto.

- URI: `/apps/projectId/jobs/import/jobId`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/import/jobId`

`mobiletargeting:GetImportJobs`

Permite recuperar información sobre todos los trabajos de importación de un proyecto.

- URI: `/apps/projectId/jobs/import`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Journeys

Los siguientes permisos están relacionados con la administración de recorridos en la cuenta de Amazon Pinpoint.

`mobiletargeting:CreateJourney`

Permite crear un recorrido para un proyecto.

- URI: `/apps/projectId/journeys`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

mobiletargeting:GetJourney

Recupera información sobre un segmento específico.

- URI: /apps/*projectId*/journeys/*journeyId*
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

mobiletargeting:ListJourneys

Permite recuperar información sobre todos los recorridos de un proyecto.

- URI: /apps/*projectId*/journeys
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys

mobiletargeting:UpdateJourney

Permite actualizar la configuración y otros ajustes de un recorrido específico.

- URI: /apps/*projectId*/journeys/*journeyId*
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

mobiletargeting:UpdateJourneyState

Permite cancelar un recorrido activo.

- URI: /apps/*projectId*/journeys/*journeyId*/state
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/state

mobiletargeting:DeleteJourney

Elimina un viaje específico.

- URI: /apps/*projectId*/journeys/*journeyId*
- Método: DELETE
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

Plantillas de mensaje

Los siguientes permisos están relacionados con la creación y administración de plantillas de mensajes para la cuenta de Amazon Pinpoint. Una plantilla de mensaje es un conjunto de contenido y configuración que puede crear, guardar y volver a utilizar en los mensajes que envíe para cualquiera de los proyectos de Amazon Pinpoint.

`mobiletargeting:ListTemplates`

Recupere información sobre todas las plantillas de mensajes asociadas a la cuenta de Amazon Pinpoint.

- URI: `/templates`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates`

`mobiletargeting:ListTemplateVersions`

Recuperar información sobre todas las versiones de una plantilla de mensaje específica.

- URI: `/templates/template-name/template-type/versions`
- Método: GET
- ARN de recursos: no se aplica

`mobiletargeting:UpdateTemplateActiveVersion`

Designe una versión específica de una plantilla de mensaje como la versión activa de la plantilla.

- URI: `/templates/template-name/template-type/active-version`
- Método: GET
- ARN de recursos: no se aplica

`mobiletargeting:GetEmailTemplate`

Permite recuperar información sobre una plantilla de mensaje para los mensajes que se envían a través del canal de correo electrónico.

- URI: `/templates/template-name/email`
- Método: GET

- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:CreateEmailTemplate`

Permite crear una plantilla de mensaje para los mensajes que se envían a través del canal de correo electrónico.

- URI: `/templates/template-name/email`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:UpdateEmailTemplate`

Permite actualizar una plantilla de mensaje existente para los mensajes que se envían a través del canal de correo electrónico.

- URI: `/templates/template-name/email`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:DeleteEmailTemplate`

Permite eliminar una plantilla de mensaje para los mensajes enviados a través del canal de correo electrónico.

- URI: `/templates/template-name/email`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:GetPushTemplate`

Permite recuperar información sobre una plantilla de mensaje para los mensajes que se envían a través de un canal de notificación push.

- URI: `/templates/template-name/push`

- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:CreatePushTemplate`

Permite crear una plantilla de mensaje para los mensajes que se envían a través de un canal de notificación push.

- URI: `/templates/template-name/push`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:UpdatePushTemplate`

Permite actualizar una plantilla de mensaje existente para los mensajes que se envían a través de un canal de notificación push.

- URI: `/templates/template-name/push`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:DeletePushTemplate`

Permite eliminar una plantilla de mensajes para los mensajes enviados a través de un canal de notificación push.

- URI: `/templates/template-name/push`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:GetSmsTemplate`

Permite recuperar información sobre una plantilla de mensaje para los mensajes que se envían a través del canal de SMS.

- URI: /templates/*template-name*/sms
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:CreateSmsTemplate

Permite crear una plantilla de mensaje para los mensajes que se envían a través del canal de SMS.

- URI: /templates/*template-name*/sms
- Método: POST
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:UpdateSmsTemplate

Permite actualizar una plantilla de mensaje existente para los mensajes que se envían a través del canal de SMS.

- URI: /templates/*template-name*/sms
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:DeleteSmsTemplate

Permite eliminar una plantilla de mensaje para los mensajes enviados a través del canal de SMS.

- URI: /templates/*template-name*/sms
- Método: DELETE
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:GetVoiceTemplate

Permite recuperar información sobre una plantilla de mensaje para los mensajes que se envían a través del canal de voz.

- URI: /templates/*template-name*/voice
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:CreateVoiceTemplate

Permite crear una plantilla de mensaje para los mensajes que se envían a través del canal de voz.

- URI: /templates/*template-name*/voice
- Método: POST
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:UpdateVoiceTemplate

Permite actualizar una plantilla de mensaje existente para los mensajes que se envían a través del canal de voz.

- URI: /templates/*template-name*/voice
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:DeleteVoiceTemplate

Permite eliminar una plantilla de mensaje para los mensajes enviados a través del canal de voz.

- URI: /templates/*template-name*/voice
- Método: DELETE
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

Mensajes

Los siguientes permisos están relacionados con el envío de mensajes y notificaciones de inserción desde la cuenta de Amazon Pinpoint. Puede utilizar las operaciones `SendUsersMessages` y `SendMessages` para enviar mensajes a puntos de enlace específicos sin necesidad de tener que crear primero segmentos y campañas.

`mobiletargeting:SendMessages`

Permite enviar un mensaje o una notificación push a puntos de enlace específicos.

- URI: `/apps/projectId/messages`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

`mobiletargeting:SendUsersMessages`

Permite enviar un mensaje o una notificación push a todos los puntos de enlace asociados a un ID de usuario concreto.

- URI: `/apps/projectId/users-messages`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

Contraseñas únicas

Los siguientes permisos están relacionados con el envío y la validación de contraseñas de un solo uso (OTPs) en Amazon Pinpoint.

`mobiletargeting:SendOTPMessage`

Envíe un mensaje de texto que contenga una contraseña de un solo uso.

- URI: `/apps/projectId/otp`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/otp`

`mobiletargeting:VerifyOTPMessage`

Compruebe la validez de una contraseña de un solo uso (OTP) que se generó mediante la operación de envío. OTPMessage

- URI: `/apps/projectId/verify-otp`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/verify-otp`

Validación de números de teléfono

Los siguientes permisos están relacionados con el uso del servicio de validación de números de teléfono en Amazon Pinpoint.

`mobiletargeting:PhoneNumberValidate`

Permite recuperar información sobre un número de teléfono.

- URI: `/phone/number/validate`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:phone/number/validate`

Proyectos

Los siguientes permisos están relacionados con la administración de proyectos en la cuenta de Amazon Pinpoint. En un principio, los proyectos se denominaban aplicaciones. Para los objetivos de estas operaciones, una aplicación de Amazon Pinpoint es lo mismo que un proyecto de Amazon Pinpoint.

`mobiletargeting:CreateApp`

Cree un proyecto de Amazon Pinpoint.

- URI: `/apps`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:DeleteApp`

Elimine un proyecto de Amazon Pinpoint.

- URI: `/apps/projectId`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApp`

Recupere información sobre un proyecto de Amazon Pinpoint.

- URI: `/apps/projectId`

- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApps`

Recupere información sobre todos los proyectos asociados a la cuenta de Amazon Pinpoint.

- URI: `/apps`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:GetApplicationSettings`

Recupere la configuración predeterminada para un proyecto de Amazon Pinpoint.

- URI: `/apps/projectId/settings`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:UpdateApplicationSettings`

Actualice la configuración predeterminada para un proyecto de Amazon Pinpoint.

- URI: `/apps/projectId/settings`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Modelos de recomendación

Los siguientes permisos están relacionados con la administración de configuraciones de Amazon Pinpoint para recuperar y procesar datos de recomendaciones de modelos de recomendación. Un modelo de recomendación es un tipo de modelo de aprendizaje automático que predice y genera recomendaciones personalizadas mediante la búsqueda de patrones en los datos.

`mobiletargeting:CreateRecommenderConfiguration`

Crear una configuración de Amazon Pinpoint para un modelo de recomendación.

- URI: `/recommenders`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:recommenders`

mobiletargeting:GetRecommenderConfigurations

Recupere información sobre todas las configuraciones de modelos de recomendación asociadas a la cuenta de Amazon Pinpoint.

- URI: /recommenders
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:recommenders

mobiletargeting:GetRecommenderConfiguration

Recupere información sobre una configuración de Amazon Pinpoint individual para un modelo de recomendación.

- URI: /recommenders/*recommenderId*
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

mobiletargeting:UpdateRecommenderConfiguration

Actualizar una configuración de Amazon Pinpoint para un modelo de recomendación.

- URI: /recommenders/*recommenderId*
- Método: PUT
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

mobiletargeting:DeleteRecommenderConfiguration

Eliminar una configuración de Amazon Pinpoint para un modelo de recomendación.

- URI: /recommenders/*recommenderId*
- Método: DELETE
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

Segmentos

Los siguientes permisos están relacionados con la administración de segmentos en la cuenta de Amazon Pinpoint. En Amazon Pinpoint, los segmentos son grupos de destinatarios de las campañas que comparten determinados atributos que usted defina.

mobiletargeting:CreateSegment

Permite crear un segmento. Para permitir que un usuario cree un segmento mediante la importación de datos de punto de conexión desde fuera de Amazon Pinpoint, permita la acción `mobiletargeting:CreateImportJob`.

- URI: `/apps/projectId/segments`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

mobiletargeting:DeleteSegment

Permite eliminar un segmento.

- URI: `/apps/projectId/segments/segmentId`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

mobiletargeting:GetSegment

Recupere información sobre un segmento específico.

- URI: `/apps/projectId/segments/segmentId`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

mobiletargeting:GetSegmentExportJobs

Permite recuperar información sobre los trabajos que exportan definiciones de punto de enlace para un segmento.

- URI: `/apps/projectId/segments/segmentId/jobs/export`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId/jobs/export`

mobiletargeting:GetSegments

Permite recuperar información sobre todos los segmentos de un proyecto.

- URI: `/apps/projectId/segments`
- Método: GET

- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetSegmentImportJobs`

Recupere información sobre los trabajos que crean segmentos mediante la importación de definiciones de punto de conexión desde Amazon S3.

- URI: `/apps/projectId/segments/segmentId/jobs/import`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

`mobiletargeting:GetSegmentVersion`

Recupere información acerca de la versión de un segmento específico.

- URI: `/apps/projectId/segments/segmentId/versions/versionId`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

`mobiletargeting:GetSegmentVersions`

Recupere información acerca de las versiones actual y anteriores de un segmento.

- URI: `/apps/projectId/segments/segmentId/versions`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

`mobiletargeting:UpdateSegment`

Actualice un segmento específico.

- URI: `/apps/projectId/segments/segmentId`
- Método: PUT
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

Tags

Los siguientes permisos están relacionados con la consulta y administración de etiquetas para recursos de Amazon Pinpoint.

`mobiletargeting:ListTagsForResource`

Permite recuperar información sobre las etiquetas que están asociadas con un proyecto, campaña, plantilla de mensaje o segmento.

- URI: `/tags/resource-arn`
- Método: GET
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:TagResource`

Permite agregar una o más etiquetas a un proyecto, campaña, plantilla de mensaje o segmento.

- URI: `/tags/resource-arn`
- Método: POST
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:UntagResource`

Permite quitar una o varias etiquetas de un proyecto, campaña, plantilla de mensaje o segmento.

- URI: `/tags/resource-arn`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:*`

Users

Los siguientes permisos están relacionados con la administración de usuarios. En Amazon Pinpoint, el término usuarios hace referencia a las personas que reciben mensajes enviados por usted. Un usuario puede estar asociado con uno o varios puntos de enlace.

`mobiletargeting:DeleteUserEndpoints`

Permite eliminar todos los puntos de enlace asociados a un ID de usuario.

- URI: `/apps/projectId/users/userId`
- Método: DELETE
- ARN del recurso: `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

mobiletargeting:GetUserEndpoints

Permite recuperar información sobre todos los puntos de enlace que están asociados a un ID de usuario.

- URI: /apps/*projectId*/users/*userId*
- Método: GET
- ARN del recurso: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/users/*userId*

Acciones de la API de SMS y voz de Amazon Pinpoint, versión 1

En esta sección se identifican las acciones para las características que están disponibles en la API de SMS y voz de Amazon Pinpoint. Esta es una API complementaria que proporciona opciones avanzadas para usar y administrar los canales de SMS y voz en Amazon Pinpoint. Para obtener más información sobre esta API, consulte la Referencia de la API de SMS y voz de Amazon Pinpoint.

sms-voice:CreateConfigurationSet

Permite crear un conjunto de configuración para enviar mensajes de voz.

- URI: /sms-voice/configuration-sets
- Método: POST
- ARN de recurso: no disponible. Utilice *.

sms-voice:DeleteConfigurationSet

Permite eliminar un conjunto de configuración para enviar mensajes de voz.

- URI: /sms-voice/configuration-sets/ *ConfigurationSetName*
- Método: DELETE
- ARN de recurso: no disponible. Utilice *.

sms-voice:GetConfigurationSetEventDestinations

Permite obtener información acerca de un conjunto de configuración y de los destinos de eventos que contiene.

- URI: /sms-voice/configuration-sets/ *ConfigurationSetName* /event-destinations

- Método: GET
- ARN de recurso: no disponible. Utilice *.

sms-voice:CreateConfigurationSetEventDestination

Permite crear un destino de evento para eventos de voz.

- URI: /sms-voice/configuration-sets/ /event-destinations *ConfigurationSetName*
- Método: POST
- ARN de recurso: no disponible. Utilice *.

sms-voice:UpdateConfigurationSetEventDestination

Permite actualizar un destino de evento para eventos de voz.

- URI: /sms-voice/configuration-sets/ *ConfigurationSetName* /event-destinations/ *EventDestinationName*
- Método: PUT
- ARN de recurso: no disponible. Utilice *.

sms-voice:DeleteConfigurationSetEventDestination

Permite eliminar un destino de evento para eventos de voz.

- URI: /sms-voice/configuration-sets/ *ConfigurationSetName* /event-destinations/ *EventDestinationName*
- Método: DELETE
- ARN de recurso: no disponible. Utilice *.

sms-voice:SendVoiceMessage

Permite crear y enviar mensajes de voz.

- URI: /sms- voice/voice/message
- Método: POST
- ARN de recurso: no disponible. Utilice *.

Ejemplos de políticas basadas en identidades de Amazon Pinpoint

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amazon Pinpoint. Tampoco pueden realizar tareas mediante la Consola de administración de AWS AWS CLI, o una AWS API. Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de API concretas en los recursos necesarios. El administrador debe asociar esas políticas a los usuarios o grupos que necesiten esos permisos.

Para obtener más información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas de JSON de ejemplo, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM.

Temas

- Prácticas recomendadas relativas a políticas
- Uso de la consola de Amazon Pinpoint
- Ejemplo: acceder a un único proyecto de Amazon Pinpoint
- Ejemplo: visualización de recursos de Amazon Pinpoint en función de las etiquetas
- Ejemplo: permitir que los usuarios vean sus propios permisos
- Ejemplos: proporcionar acceso a acciones de la API de Amazon Pinpoint
- Ejemplos: proporcionar acceso a acciones de la API de SMS y voz de Amazon Pinpoint
- Ejemplo: restringir el acceso de un proyecto de Amazon Pinpoint a direcciones IP específicas
- Ejemplo: restringir el acceso de Amazon Pinpoint en función de las etiquetas
- Ejemplo: permitir que Amazon Pinpoint envíe correos electrónicos con identidades que se verificaron en Amazon SES

Prácticas recomendadas relativas a políticas

Las políticas basadas en identidades determinan si alguien puede crear, acceder o eliminar los recursos de Amazon Pinpoint de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos en muchos casos de uso comunes. Están

disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola de Amazon Pinpoint

Para acceder a la consola de Amazon Pinpoint, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de Amazon Pinpoint de su AWS cuenta. Si crea una política basada en identidad que aplica permisos más restrictivos que el

mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política. Para asegurarse de que esas entidades puedan utilizar la consola de Amazon Pinpoint, asocie una política a dichas entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

El siguiente ejemplo de política proporciona acceso de solo lectura a la consola Amazon Pinpoint en una región específica. AWS Incluye acceso de solo lectura a otros servicios de los que depende la consola de Amazon Pinpoint, como Amazon Simple Email Service (Amazon SES), IAM y Amazon Kinesis.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "UseConsole",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "kinesis:ListStreams",
                "s3:List*",
                "ses:Describe*",
                "ses:Get*",
                "ses:List*",
                "sns:ListTopics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
```

```
        }
    ]
}
```

En el ejemplo de política anterior, *region* sustitúyalo por el nombre de una AWS región y sustitúyelo por el identificador de tu *accountId* AWS cuenta.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Ejemplo: acceder a un único proyecto de Amazon Pinpoint

También puede crear políticas de solo lectura que proporcionan acceso únicamente a proyectos específicos. La siguiente política de ejemplo permite a los usuarios iniciar sesión en la consola y ver una lista de proyectos. También permite a los usuarios ver información sobre recursos relacionados para otros servicios de AWS de los que depende la consola de Amazon Pinpoint, como Amazon SES, IAM y Amazon Kinesis. Sin embargo, la política solo permite ver a los usuarios información adicional sobre el proyecto que está especificado en la política. Puedes modificar esta política para permitir el acceso a proyectos o AWS regiones adicionales.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
```

```
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

En el ejemplo anterior, *region* sustitúyalo por el nombre de una AWS región, *accountId* sustitúyelo por el ID de tu AWS cuenta y *projectId* sustitúyelo por el ID del proyecto de Amazon Pinpoint al que quieres dar acceso.

Del mismo modo, puede crear políticas que concedan a un usuario de su AWS cuenta con acceso de escritura limitado a uno de sus proyectos de Amazon Pinpoint, por ejemplo, el proyecto que tiene el ID del `810c7aab86d42fb2b56c8c966example` proyecto. En este caso, desea permitir que el usuario vea, agregue y actualice componentes del proyecto, como segmentos y campañas, pero que no elimine ningún componente.

Además de conceder permisos para las acciones `mobiletargeting:Get` y `mobiletargeting:List`, cree una política que conceda permisos para las siguientes acciones: `mobiletargeting:Create`; `mobiletargeting:Update`; y `mobiletargeting:Put`. Estos son los permisos adicionales necesarios para crear y administrar la mayoría de los componentes del proyecto. Por ejemplo:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LimitedWriteProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*",
                "mobiletargeting:Create*",
                "mobiletargeting:Update*",
                "mobiletargeting:Put*"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/810c7aab86d42fb2b56c8c966example",
    "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/810c7aab86d42fb2b56c8c966example/*",
                "arn:aws:mobiletargeting:us-east-1:111122223333:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Ejemplo: visualización de recursos de Amazon Pinpoint en función de las etiquetas

Puede utilizar las condiciones de una política basada en la identidad para controlar el acceso a los recursos de Amazon Pinpoint basados en etiquetas. Esta política de ejemplo muestra cómo puede crear este tipo de política para permitir la consulta de recursos de Amazon Pinpoint. Sin embargo, los permisos solo se conceden si la etiqueta de recurso `Owner` tiene el valor del nombre de usuario de dicho usuario. Esta política también proporciona los permisos necesarios para llevar a cabo esta acción en la consola.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewResourceIfOwner",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:*:*:*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
```

```
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:*"
                }
            }
        }
    ]
}
```

También puede asociar este tipo de política a los usuarios de de su cuenta. Si un usuario llamado `richard-roe` intenta ver un recurso de Amazon Pinpoint, el recurso debe estar etiquetado como `0wner=richard-roe` o `owner=richard-roe`. De lo contrario, se le deniega el acceso. La clave de la etiqueta de condición `0wner` coincide con los nombres de las claves de condición `0wner` y `owner` porque no distinguen entre mayúsculas y minúsculas. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.

Ejemplo: permitir que los usuarios vean sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la AWS CLI API o. AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
```

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

Ejemplos: proporcionar acceso a acciones de la API de Amazon Pinpoint

Esta sección proporciona políticas de ejemplo que permiten el acceso a las características que están disponibles desde la API de Amazon Pinpoint, que es la API principal de Amazon Pinpoint. Para obtener más información sobre esta API, consulte la referencia de la API de Amazon Pinpoint.

Acceso de solo lectura

El siguiente ejemplo de política permite el acceso de solo lectura a todos los recursos de su cuenta de Amazon Pinpoint en una región específica. AWS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewAllResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
```

```
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        }
    ]
}
```

En el ejemplo anterior, *region* sustitúyalo por el nombre de una AWS región y sustitúyelo por el identificador de tu *accountId* AWS cuenta.

Acceso de administrador

La política de ejemplo siguiente permite el acceso completo a todas las acciones y recursos de Amazon Pinpoint en la cuenta de Amazon Pinpoint:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        }
    ]
}
```

En el ejemplo anterior, *accountId* sustitúyalo por tu ID de AWS cuenta.

Ejemplos: proporcionar acceso a acciones de la API de SMS y voz de Amazon Pinpoint

En esta sección se proporcionan políticas de ejemplo que permiten el acceso a las características que están disponibles en la API de SMS y voz de Amazon Pinpoint. Esta es una API complementaria que proporciona opciones avanzadas para usar y administrar los canales de SMS y voz en Amazon Pinpoint. Para obtener más información sobre esta API, consulte la Referencia de la API de SMS y voz de Amazon Pinpoint.

Acceso de solo lectura

El siguiente ejemplo de política permite el acceso de solo lectura a todas las acciones y recursos de la API de SMS y voz de Amazon Pinpoint de su cuenta: AWS

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SMSVoiceReadOnly",
            "Effect": "Allow",
            "Action": [
                "sms-voice:Get*",
                "sms-voice:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:us-east-1:111122223333:*"
                }
            }
        }
    ]
}
```

Acceso de administrador

El siguiente ejemplo de política permite el acceso total a todas las acciones y recursos de la API de SMS y voz de Amazon Pinpoint de su AWS cuenta:

JSON

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

```
        {
            "Sid": "SMSVoiceFullAccess",
            "Effect": "Allow",
            "Action": [
                "sms-voice:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:us-east-1:111122223333:*"
                }
            }
        }
    ]
}
```

Ejemplo: restringir el acceso de un proyecto de Amazon Pinpoint a direcciones IP específicas

El siguiente ejemplo de política otorga permisos a cualquier usuario para realizar cualquier acción de Amazon Pinpoint en un proyecto específico ()*projectId*. Sin embargo, la solicitud debe proceder del intervalo de direcciones IP especificado en la condición.

La condición de esta declaración identifica el `54.240.143.*` rango de direcciones del Protocolo de Internet de la versión 4 (IPv4) permitidas, con una excepción:`54.240.143.188`. El `Condition` bloque usa las `NotIpAddress` condiciones `IpAddress` and y la clave de `aws:SourceIp` condición, que es una clave AWS de condición amplia. Para obtener más información acerca de estas claves de condición, consulte Especificación de condiciones en una política en la Guía del usuario de IAM. Los `aws:SourceIp` IPv4 valores utilizan la notación CIDR estándar. Para obtener más información, consulte Operadores de condición de dirección IP en la Guía del usuario de IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Id":"AMZPinpointPolicyId1",
    "Statement":[
```

```
        {
            "Sid":"IPAllow",
            "Effect":"Allow",
            "Principal":"*",
            "Action":"mobiletargeting:*",
            "Resource":[
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*"
            ],
            "Condition":{
                "IpAddress":{
                    "aws:SourceIp":"54.240.143.0/24"
                },
                "NotIpAddress":{
                    "aws:SourceIp":"54.240.143.188/32"
                }
            }
        }
    ]
}
```

Ejemplo: restringir el acceso de Amazon Pinpoint en función de las etiquetas

El siguiente ejemplo de política otorga permisos para realizar cualquier acción de Amazon Pinpoint en un proyecto específico ()*projectId*. Sin embargo, los permisos solo se conceden si la solicitud proviene de un usuario cuyo nombre es un valor en la etiqueta de recurso `Owner` del proyecto, como se especifica en la condición.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ModifyResourceIfOwner",
            "Effect": "Allow",
            "Action": "mobiletargeting:*",
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                }
            }
        }
    ]
}
```

Ejemplo: permitir que Amazon Pinpoint envíe correos electrónicos con identidades que se verificaron en Amazon SES

Cuando verifique la identidad de un correo electrónico (por ejemplo, una dirección de correo electrónico o un dominio) a través de la consola de Amazon Pinpoint, esa identidad se configura automáticamente para que Amazon Pinpoint y Amazon SES puedan utilizarla. Sin embargo, si verifica la identidad de un correo electrónico a través de Amazon SES y quiere usar esa identidad con Amazon Pinpoint, debe aplicar una política a esa identidad.

El siguiente ejemplo de política adjudica a Amazon Pinpoint permiso para enviar correos electrónicos con una identidad de correo electrónico verificada a través de Amazon SES.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws:ses:us-east-1:111122223333:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"111122223333"
                },
                "ArnLike":{
```

```
                    "aws:SourceArn":"arn:aws:mobiletargeting:us-
east-1:111122223333:apps/*"
                }
            }
        }
    ]
}
```

Si utiliza Amazon Pinpoint en la región AWS GovCloud (EE. UU.-Oeste), utilice el siguiente ejemplo de política en su lugar:

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws-us-gov:ses:us-gov-
west-1:111122223333:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"111122223333"
                },
                "ArnLike":{
                    "aws:SourceArn":"arn:aws-us-gov:mobiletargeting:us-gov-
west-1:111122223333:apps/*"
                }
            }
        }
    ]
}
```

Roles de IAM para tareas comunes de Amazon Pinpoint

Un rol de IAM es una identidad AWS Identity and Access Management (IAM) que puede crear en su AWS cuenta y conceder permisos específicos. Un rol de IAM es una AWS identidad con políticas de permisos que determinan lo que la identidad puede y no puede hacer en ella. AWS No obstante, en lugar de asociarse solo a una persona, cualquiera que lo necesite puede asumir un rol.

Además, un rol no tiene credenciales a largo plazo estándar asociadas. En lugar de ello, proporciona credenciales de seguridad temporales para una sesión. Puede utilizar las funciones de IAM para delegar el acceso a usuarios, aplicaciones, aplicaciones o servicios que normalmente no tienen acceso a sus AWS recursos.

Por estos motivos, puede usar roles de IAM para integrar Amazon Pinpoint con determinados servicios y recursos de AWS de la cuenta. Por ejemplo, es posible que desee permitir a Amazon Pinpoint acceder a las definiciones de punto de conexión que almacene en un bucket de Amazon Simple Storage Service (Amazon S3) y desee usar para segmentos. O es posible que desee permitir a Amazon Pinpoint que transmita datos de eventos a una secuencia de Amazon Kinesis para la cuenta. Del mismo modo, es posible que desee utilizar las funciones de IAM para permitir que las aplicaciones web o móviles registren puntos de conexión o generen informes sobre los datos de uso de los proyectos de Amazon Pinpoint, sin necesidad de AWS incrustar claves en las aplicaciones (ya que pueden resultar difíciles de rotar y los usuarios pueden extraerlas).

Para estos escenarios, puede delegar el acceso a Amazon Pinpoint mediante los roles de IAM. En esta sección se proporcionan explicaciones y ejemplos de tareas comunes de Amazon Pinpoint que utilizan roles de IAM para trabajar con otros servicios de AWS . Para obtener información sobre el uso de roles de IAM con aplicaciones web y móviles más específicamente, consulte Proporcionar acceso a usuarios autenticados externamente (federación de identidades) en la Guía del usuario de IAM.

Temas

- Rol de IAM para importar puntos de conexión o segmentos
- Rol de IAM para exportar puntos de conexión o segmentos
- Rol de IAM para recuperar las recomendaciones de Amazon Personalize
- Rol de IAM para la transmisión de eventos a Kinesis
- Rol de IAM para el envío de correos electrónicos con Amazon SES

Rol de IAM para importar puntos de conexión o segmentos

Con Amazon Pinpoint, puede definir un segmento de usuarios importando definiciones de puntos de conexión desde un bucket de Amazon Simple Storage Service (Amazon S3) de su cuenta. AWS Antes de importar, es necesario delegar los permisos necesarios a Amazon Pinpoint. Para ello, debe crear un rol AWS Identity and Access Management (IAM) y adjuntar las siguientes políticas al rol:

- La política administrada de `AmazonS3ReadOnlyAccess` AWS . Esta política ha sido creada y gestionada por AWS tu bucket de Amazon S3 y te concede acceso de solo lectura.
- Una política de confianza que permita a Amazon Pinpoint asumir el rol.

Tras crear el rol, puede utilizar Amazon Pinpoint para importar segmentos de un bucket de Amazon S3. Para obtener información acerca de la creación del bucket, la creación de archivos de punto de conexión y la importación de un segmento mediante la consola, consulte Importación de segmentos en la Guía del usuario de Amazon Pinpoint. Para ver un ejemplo de cómo importar un segmento mediante programación mediante el AWS SDK para Java, consulte Importar segmentos en Amazon Pinpoint. esta guía.

Creación del rol de IAM (AWS CLI)

Complete los siguientes pasos para crear el rol de IAM mediante (). AWS Command Line Interface AWS CLI Si no ha instalado el AWS CLI, consulte Instalación del AWS CLI en la Guía del AWS Command Line Interface usuario.

Para crear el rol de IAM mediante el AWS CLI

1. Cree un archivo JSON que contenga la política de confianza correspondiente al rol y guárdelo localmente. Puede utilizar la siguiente política de confianza.

   JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
          {
               "Action": "sts:AssumeRole",
               "Effect": "Allow",
               "Principal": {
```

```
                    "Service": "pinpoint.amazonaws.com"
               },
               "Condition": {
                    "StringEquals": {
                         "aws:SourceAccount": "111122223333"
                    },
               "ArnLike": {
               "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/application-id"
                    }
               }
          }
     ]
}
```

En el ejemplo anterior, realice el siguiente procedimiento:

- *region*Sustitúyalo por la AWS región en la que utiliza Amazon Pinpoint.
- *accountId*Sustitúyalo por el identificador único de su AWS cuenta.
- *application-id*Sustitúyalo por el identificador único del proyecto.

2. En la línea de comandos, utilice el comando `create-role` para crear el rol y asociar la política de confianza:

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document
 file://PinpointImportTrustPolicy.json
```

Tras el prefijo `file://`, especifique la ruta del archivo JSON que contiene la política de confianza.

Después de ejecutar este comando, verá un resultado similar al siguiente en su terminal:

3. Utilice el `attach-role-policy`comando para adjuntar la política `AmazonS3ReadOnlyAccess` AWS gestionada al rol:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport
```

Rol de IAM para exportar puntos de conexión o segmentos

Para obtener una lista de puntos de enlace, puede crear un trabajo de exportación. Al crear un trabajo de exportación, es preciso especificar un ID de proyecto y, opcionalmente, un ID de segmento. A continuación, Amazon Pinpoint exporta una lista de los puntos de conexión asociados con el proyecto o segmento a un bucket de Amazon Simple Storage Service (Amazon S3). El archivo resultante contiene una lista de puntos de enlace con formato JSON y sus atributos (tales como el canal, la dirección, el estado de alta o baja en el servicio de notificaciones, la fecha de creación o el ID del punto de enlace).

Para crear un trabajo de exportación, debe configurar un rol de IAM que permita a Amazon Pinpoint escribir en un bucket de Amazon S3. El proceso de configuración de la función consta de dos pasos:

1. Cree una política de IAM que permita a una entidad (en este caso, Amazon Pinpoint) escribir en un bucket de Amazon S3 específico.
2. Cree un rol de IAM y asóciele la política.

Este tema contiene procedimientos para completar ambos pasos. En estos procedimientos se da por hecho que ya ha creado un bucket de Amazon S3 y una carpeta en ese bucket, para almacenar los segmentos exportados. Para obtener información acerca de la creación de buckets, consulte Crear un bucket en la Guía del usuario de Amazon Simple Storage Service.

En estos procedimientos se da por hecho que ya se ha instalado y configurado la AWS Command Line Interface (AWS CLI). Para obtener información sobre la configuración del AWS CLI, consulte Instalación del AWS CLI en la Guía del AWS Command Line Interface usuario.

Paso 1: creación de la política de IAM

Una política de IAM define los permisos para una entidad (por ejemplo, de una identidad o recurso). Para crear un rol que permita exportar puntos de conexión de Amazon Pinpoint, es preciso crear una política que conceda permiso para escribir en una carpeta concreta en un bucket de Amazon S3 determinado. La siguiente política de ejemplo observa la práctica de seguridad de conceder los privilegios mínimos; es decir, se conceden solo aquellos permisos que son imprescindibles para realizar una tarea.

Creación de la política de IAM

1. En un editor de texto, cree un nuevo archivo. Pegue el código siguiente en el archivo:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::*" ]
        },
        {
            "Sid": "AllowRootAndHomeListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket" ],
            "Condition": {
                "StringEquals": {
                    "s3:delimiter": [ "/" ],
                    "s3:prefix": [
                        "",
                        "Exports/"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket" ],
            "Condition": {
                "StringLike": {
```

```
                    "s3:prefix": [
                        "Exports/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
            "Action": [ "s3:*" ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket/Exports/*" ]
        }
    ]
}
```

En el código anterior, sustituya todas las instancias de *amzn-s3-demo-bucket-example-bucket* por el nombre del bucket de Amazon S3 que contiene la carpeta a la que desea exportar la información del segmento. Además, sustituya todas las instancias *Exports* de por el nombre de la propia carpeta.

Cuando haya terminado, guarde el archivo como `s3policy.json`.

2. Con el AWS CLI, navegue hasta el directorio en el que se encuentra el `s3policy.json` archivo. A continuación, escriba el siguiente comando para crear la política:

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
 file://s3policy.json
```

Si la política se ha creado correctamente, aparecerá un resultado similar al siguiente:

```
{
    "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": "/"
    }
}
```

Copie el nombre de recurso de Amazon (ARN) de la política (`arn:aws:iam::123456789012:policy/s3ExportPolicy` en el ejemplo anterior). En la siguiente sección, debe proporcionar este ARN al crear la función.

> **Note**
>
> Si aparece un mensaje que indica que la cuenta no está autorizada a llevar a cabo la operación `CreatePolicy`, deberá conectar una política al usuario que permite crear políticas y roles de IAM nuevos. Para obtener más información, consulta Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

Paso 2: Crear el rol de IAM

Ahora que ha creado una política de IAM, puede crear un rol y conectarle la política. Cada rol de IAM; contiene una política de confianza, que es un conjunto de reglas que especifica las entidades que están autorizadas a asumir el rol. En esta sección, se crea una política de confianza que permita que Amazon Pinpoint asuma el rol. A continuación, creará la función en sí y, después, le conectará la política que creó en la sección anterior.

Creación del rol de IAM

1. En un editor de texto, cree un nuevo archivo. Pegue el código siguiente en el archivo:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"sts:AssumeRole",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/applicationId"
                }
            }
        }
    ]
}
```

Guarde el archivo como `trustpolicy.json`.

2. Con el AWS CLI, navegue hasta el directorio en el que se encuentra el `trustpolicy.json` archivo. A continuación, escriba el siguiente comando para crear un nuevo rol:

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
 file://trustpolicy.json
```

3. En la línea de comandos, escriba el siguiente comando para asociar la política que creó en la sección anterior al rol que acaba de crear:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
s3ExportPolicy --role-name s3ExportRole
```

En el comando anterior, *arn:aws:iam::123456789012:policy/s3ExportPolicy* sustitúyalo por el ARN de la política que creó en la sección anterior.

Rol de IAM para recuperar las recomendaciones de Amazon Personalize

Puede configurar Amazon Pinpoint para recuperar datos de recomendación de una solución de Amazon Personalize que se haya implementado como campaña de Amazon Personalize. Puede utilizar estos datos para enviar recomendaciones personalizadas a los destinatarios de los mensajes en función de los atributos y el comportamiento de cada destinatario. Para obtener más información, consulte Modelos de machine learning en la Guía del usuario de Amazon Pinpoint.

Antes de poder recuperar los datos de recomendación de una campaña de Amazon Personalize, debe crear un rol de AWS Identity and Access Management (IAM) que permita a Amazon Pinpoint recuperar los datos de la campaña. Amazon Pinpoint puede crear este rol para usted

automáticamente cuando utilice la consola para configurar un modelo de recomendación en Amazon Pinpoint. O bien, puede crear este rol manualmente.

Para crear el rol manualmente, utilice la API de IAM para completar los siguientes pasos:

1. Cree una política de IAM que permita a una entidad (en este caso, Amazon Pinpoint) recuperar datos de recomendación de una campaña de Amazon Personalize.
2. Cree un rol de IAM y asocie la política de IAM al rol.

En este tema se explica cómo completar estos pasos mediante el AWS Command Line Interface (AWS CLI). Supone que ya ha creado la solución de Amazon Personalize y la ha implementado como una campaña de Amazon Personalize. Para obtener información sobre cómo crear e implementar una campaña, consulte Creación de una campaña en la Guía para desarrolladores de Amazon Personalize.

En este tema se supone que ya ha instalado y configurado la AWS CLI. Para obtener información sobre cómo configurar el AWS CLI, consulte Instalación del AWS CLI en la Guía del AWS Command Line Interface usuario.

Paso 1: Crear la política de IAM

Una política de IAM define los permisos para una entidad, por ejemplo, una identidad o recurso. Para crear un rol que permita a Amazon Pinpoint recuperar datos de recomendación de una campaña de Amazon Personalize, primero debe crear una política de IAM para el rol. Esta política debe permitir a Amazon Pinpoint:

- Recupere la información de configuración de la solución implementada por la campaña (`DescribeSolution`).
- Compruebe el estado de la campaña (`DescribeCampaign`).
- Recupere los datos de recomendación de la campaña (`GetRecommendations`).

En el procedimiento siguiente, la política de ejemplo permite este acceso para una solución de Amazon Personalize concreta implementada por una campaña de Amazon Personalize concreta.

Creación de la política de IAM

1. En un editor de texto, cree un nuevo archivo. Pegue el código siguiente en el archivo:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "RetrieveRecommendationsOneCampaign",
            "Effect": "Allow",
            "Action": [
                "personalize:DescribeSolution",
                "personalize:DescribeCampaign",
                "personalize:GetRecommendations"
            ],
            "Resource": [
                "arn:aws:personalize:us-
east-1:111122223333:solution/solutionId",
                "arn:aws:personalize:us-
east-1:111122223333:campaign/campaignId"
            ]
        }
    ]
}
```

En el ejemplo anterior, sustituya el *italicized* texto por su información:

- *region*— El nombre de la AWS región que aloja la solución y la campaña Amazon Personalize.
- *accountId*— Tu Cuenta de AWS número de identidad.
- *solutionId*— El identificador de recurso único de la solución Amazon Personalize que implementa la campaña.
- *campaignId*— El identificador de recurso único de la campaña Amazon Personalize desde el que recuperar los datos de las recomendaciones.

2. Cuando haya terminado, guarde el archivo como `RetrieveRecommendationsPolicy.json`.
3. Mediante la interfaz de línea de comandos, desplácese hasta el directorio donde guardó el archivo `RetrieveRecommendationsPolicy.json`.

4. Escriba el siguiente comando para crear una política y asígnele el nombre `RetrieveRecommendationsPolicy`. Para usar un nombre diferente, *`RetrieveRecommendationsPolicy`* cámbielo por el nombre que desee.

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document
 file://RetrieveRecommendationsPolicy.json
```

> **Note**
>
> Si recibe un mensaje que indica que la cuenta no está autorizada para llevar a cabo la operación `CreatePolicy`, deberá asociar una política al usuario que permite crear nuevas políticas de IAM y roles para la cuenta. Para obtener más información, consulta Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

5. Copie el nombre de recurso de Amazon (ARN) de la política (`arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy` en el ejemplo anterior). Necesita este ARN para crear el rol de IAM en la siguiente sección.

Paso 2: Crear el rol de IAM

Después de crear la política de IAM, puede crear un rol de IAM y asociarle la política.

Cada rol de IAM contiene una política de confianza, que es un conjunto de reglas que especifica las entidades que están autorizadas a asumir el rol. En esta sección, se crea una política de confianza que permita que Amazon Pinpoint asuma el rol. A continuación, crea el rol en sí. A continuación, asocie la política al rol.

Creación del rol de IAM

1. En un editor de texto, cree un nuevo archivo. Pegue el código siguiente en el archivo:

   JSON

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

```
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                "AWS:SourceAccount": "111122223333"
                },
                "ArnLike": {
                "AWS:SourceArn": "arn:aws:mobiletargeting:us-
east-1:444455556666:apps/*"
                }
            }
        }
    ]
}
```

2. Guarde el archivo como `RecommendationsTrustPolicy.json`.
3. Mediante la interfaz de línea de comandos, desplácese hasta el directorio donde guardó el archivo `RecommendationsTrustPolicy.json`.
4. Escriba el siguiente comando para crear un nuevo rol y llámelo `PinpointRoleforPersonalize`. Para usar un nombre diferente, *`PinpointRoleforPersonalize`* cámbielo por el nombre que desee.

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-
document file://RecommendationsTrustPolicy.json
```

5. Escriba el siguiente comando para asociar la política que creó en la sección anterior al rol que acaba de crear:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

   En el comando anterior, *`arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy`* sustitúyalo por el ARN de la política que creó en la sección anterior. Además, *`PinpointRoleforPersonalize`* sustitúyalo por el nombre del rol que especificó en el paso 4, si especificó un nombre diferente para el rol.

Rol de IAM para la transmisión de eventos a Kinesis

Amazon Pinpoint puede enviar automáticamente los datos de uso de la aplicación, o datos de eventos, desde su aplicación a una transmisión de datos de Amazon Kinesis o una transmisión de entrega de Amazon Data Firehose de su cuenta. AWS Antes de que Amazon Pinpoint pueda empezar a transmitir los datos de eventos, debe delegar los permisos necesarios a Amazon Pinpoint.

Si utiliza la consola para configurar la transmisión de eventos, Amazon Pinpoint crea automáticamente un rol de AWS Identity and Access Management (IAM) con los permisos necesarios. Para obtener más información, consulte Transmisión de eventos de Amazon Pinpoint a Amazon Kinesis en la Guía del usuario de Amazon Pinpoint.

Si desea crear el rol de forma manual, conecte las siguientes políticas a ella:

- Una política de permisos que permite a Amazon Pinpoint enviar datos de eventos a la secuencia.
- Una política de confianza que permita a Amazon Pinpoint asumir el rol.

Tras crear el rol, puede configurar Amazon Pinpoint para que envíe eventos de forma automática a la secuencia. Para obtener más información, consulte la sección Transmitir datos de eventos de aplicación a través de Kinesis y Firehose con Amazon Pinpoint de esta guía.

Creación del rol de IAM (AWS CLI)

Complete los siguientes pasos para crear manualmente el rol de IAM mediante AWS Command Line Interface (AWS CLI). Para obtener información sobre cómo crear el rol mediante la consola de Amazon Pinpoint, consulte Transmisión de eventos de Amazon Pinpoint a Kinesis en la Guía del usuario de Amazon Pinpoint.

Si no la ha instalado AWS CLI, consulte Instalación de la AWS CLI en la Guía delAWS Command Line Interface usuario. También debe haber creado un flujo de Kinesis o un flujo de Firehose. Para obtener información acerca de la creación de estos recursos, consulte Creating and Managing Streams en la Guía para desarrolladores de Amazon Kinesis Data Streams o Creating an Amazon Data Firehose delivery stream en la Guía para desarrolladores de Amazon Data Firehose.

Para crear el rol de IAM mediante el AWS CLI

1. Cree un archivo nuevo. Pegue la siguiente política en el documento y realice los cambios siguientes:

   - *region*Sustitúyalo por la AWS región en la que utiliza Amazon Pinpoint.

- *accountId*Sustitúyalo por el identificador único de su AWS cuenta.
- *applicationId*Sustitúyalo por el identificador único del proyecto.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/applicationId"
                }
            }
        }
    ]
}
```

Cuando haya terminado, guarde el archivo como `PinpointEventStreamTrustPolicy.json`.

2. Utilice el comando `create-role` para crear el rol y asociar la política de confianza:

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-
document file://PinpointEventStreamTrustPolicy.json
```

3. Cree un archivo nuevo que contenga la política de permisos para el rol.

   Si configura Amazon Pinpoint para enviar datos a un flujo de Kinesis, pegue la siguiente política en el archivo y reemplace lo siguiente:

- *region*Sustitúyalo por la AWS región en la que utiliza Amazon Pinpoint.
- *accountId*Sustitúyalo por el identificador único de su AWS cuenta.
- *streamName*Sustitúyalo por el nombre de la transmisión de Kinesis.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Action": [
            "kinesis:PutRecords",
            "kinesis:DescribeStream"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:kinesis:us-east-1:111122223333:stream/streamName"
        ]
    }
}
```

O también, si configura Amazon Pinpoint para enviar datos a un flujo de Firehose, pegue la siguiente política en el archivo y reemplace lo siguiente:

- *region*Sustitúyalo por la AWS región en la que utiliza Amazon Pinpoint.
- *accountId*Sustitúyalo por el identificador único de su AWS cuenta.
- *delivery-stream-name*Sustitúyelo por el nombre de tu transmisión Firehose.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "firehose:PutRecordBatch",
            "firehose:DescribeDeliveryStream"
```

```
        ],
        "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/delivery-
stream-name"
        ]
    }
}
```

Cuando haya terminado, guarde el archivo como `PinpointEventStreamPermissionsPolicy.json`.

4. Utilice el comando `put-role-policy` para asociar la política de permisos al rol:

```
aws iam put-role-policy --role-name PinpointEventStreamRole --policy-
name PinpointEventStreamPermissionsPolicy --policy-document file://
PinpointEventStreamPermissionsPolicy.json
```

Rol de IAM para el envío de correos electrónicos con Amazon SES

Amazon Pinpoint utiliza sus recursos de Amazon SES para enviar correos electrónicos para su campaña o recorrido. Para que Amazon Pinpoint pueda utilizar los recursos de Amazon SES para enviar correos electrónicos, debe conceder los permisos necesarios a Amazon Pinpoint. Su cuenta debe tener los permisos `iam:PutRolePolicy` y `iam:UpdateAssumeRolePolicy` necesarios para actualizar o crear roles de IAM.

La consola Amazon Pinpoint puede crear automáticamente un rol AWS Identity and Access Management (IAM) con los permisos necesarios. Para obtener más información, consulte Creating an email orchestration sending role en la Guía del usuario de Amazon Pinpoint.

Si desea crear el rol de forma manual, conecte las siguientes políticas a ella:

- Una política de permisos que otorga a Amazon Pinpoint acceso a sus recursos de Amazon SES.
- Una política de confianza que permita a Amazon Pinpoint asumir el rol.

Tras crear el rol, puede configurar Amazon Pinpoint para que utilice los recursos de Amazon SES.

Puede probar las políticas de IAM con el simulador de política de IAM. Para obtener más información, consulte Testing IAM policies with the IAM policy simulator en la Guía del usuario de IAM.

Creación del rol de IAM (Consola de administración de AWS)

Complete los siguientes pasos para crear manualmente un rol de IAM para su campaña o recorrido para enviar correos electrónicos.

1. Cree una nueva política de permisos siguiendo las instrucciones que se indican en Creating policies using the JSON editor en la Guía del usuario de IAM.

   - En el paso 5, utilice la siguiente política de permisos para el rol de IAM.

     - *partition*Sustitúyalo por la partición en la que se encuentra el recurso. De forma estándar Regiones de AWS, la partición es`aws`. Si tiene recursos en otras particiones, la partición es `aws-partitionname`. Por ejemplo, la partición para los recursos en el AWS GovCloud oeste de EE. UU.) es`aws-us-gov`.
     - *region*Sustitúyalo por el nombre del Región de AWS que aloja el proyecto Amazon Pinpoint.
     - *accountId*Sustitúyalo por el identificador único de su. Cuenta de AWS

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PinpointUsesSESForEmailSends",
            "Effect": "Allow",
            "Action": [
                "ses:SendEmail",
                "ses:SendRawEmail"
            ],
            "Resource": [
            "arn:aws:ses:us-east-1:111122223333:identity/*",
    "arn:aws:ses:us-east-1:111122223333:configuration-set/*"
            ]
        }
    ]
}
```

2. Cree una nueva política de permisos siguiendo las instrucciones que se indican en la sección Creación de un rol mediante políticas de confianza personalizadas de la Guía del usuario de IAM.

   a. En el paso 4, utilice la siguiente política de confianza.

      - *accountId*Reemplácelo por el ID único de su Cuenta de AWS.

      JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPinpoint",
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

   b. En el paso 11, agregue la política de permisos que ha creado en el paso anterior.

Solución de problemas de la administración de identidades y accesos de Amazon Pinpoint

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje con Amazon Pinpoint e IAM.

Temas

- No tengo autorización para realizar una acción en Amazon Pinpoint

- No estoy autorizado a realizar lo siguiente: PassRole
- Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de Amazon Pinpoint

No tengo autorización para realizar una acción en Amazon Pinpoint

Si Consola de administración de AWS le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

En el siguiente ejemplo, el error se produce cuando el usuario de `mateojackson` intenta utilizar la consola para ver detalles sobre un proyecto pero no tiene permisos `mobiletargeting:GetApp`.

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

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *my-example-project* mediante la acción `mobiletargeting:GetApp`.

No estoy autorizado a realizar lo siguiente: PassRole

Si recibe un error que indica que no tiene autorización para llevar a cabo la acción `iam:PassRole`, las políticas se deben actualizar para permitirle pasar un rol a Amazon Pinpoint.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir la función al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon Pinpoint. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de Amazon Pinpoint

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admiten políticas basadas en recursos o listas de control de acceso (ACLs), puede usar esas políticas para permitir que las personas accedan a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si Amazon Pinpoint admite estas características, consulte Cómo funciona Amazon Pinpoint con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro de su propiedad en la Cuenta de AWS Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Registro y monitoreo en Amazon Pinpoint

El registro y el monitoreo son una parte importante del mantenimiento de la fiabilidad, disponibilidad y rendimiento de los proyectos de Amazon Pinpoint y otros tipos de recursos de Amazon Pinpoint. Debe registrar y recopilar datos de monitoreo de todas las partes de sus proyectos y recursos de Amazon Pinpoint para poder depurar más fácilmente un error multipunto en caso de que se produzca. AWS proporciona varias herramientas que pueden ayudarle a registrar y recopilar estos datos y a responder a posibles incidentes:

AWS CloudTrail

Amazon Pinpoint se integra con AWS CloudTrail, que es un servicio que proporciona un registro de las acciones realizadas en Amazon Pinpoint por un usuario, un rol u otro servicio. AWS Esto incluye acciones de la consola de Amazon Pinpoint y las llamadas mediante programación a las operaciones de la API de Amazon Pinpoint. Al utilizar la información recopilada por CloudTrail, puede determinar qué solicitudes se realizaron a Amazon Pinpoint. Para cada solicitud, puede identificar cuándo se realizó, la dirección IP desde la que se realizó, quién la realizó e información adicional. Para obtener más información, consulte la sección Registre las llamadas a la API Amazon Pinpoint con AWS CloudTrail de esta guía.

Amazon CloudWatch

Puede utilizar Amazon CloudWatch para recopilar, ver y analizar varias métricas importantes relacionadas con su cuenta y sus proyectos de Amazon Pinpoint. También puede utilizarlas CloudWatch para crear alarmas que le notifiquen si el valor de una métrica cumple determinadas condiciones y está dentro o supera un umbral que usted defina. Si creas una alarma, CloudWatch envía una notificación a un tema de Amazon Simple Notification Service (Amazon SNS) que especifiques. Para obtener más información, consulte Supervisión de Amazon Pinpoint con Amazon CloudWatch en la Guía del usuario de Amazon Pinpoint.

AWS Health Cuadros de mando

Mediante el uso de AWS Health paneles, puede comprobar y supervisar el estado de su entorno Amazon Pinpoint. Para comprobar el estado general del servicio Amazon Pinpoint, utilice el AWS Service Health Dashboard. Para comprobar, supervisar y ver los datos históricos sobre cualquier suceso o problema que pueda afectar a su AWS entorno de forma más específica, utilice el AWS Personal Health Dashboard. Para obtener más información sobre estos paneles, consulte la Guía del usuario de AWS Health.

AWS Trusted Advisor

AWS Trusted Advisor inspecciona su AWS entorno y ofrece recomendaciones sobre oportunidades para abordar las brechas de seguridad, mejorar la disponibilidad y el rendimiento del sistema y ahorrar dinero. Todos AWS los clientes tienen acceso a un conjunto básico de Trusted Advisor controles. Los clientes que tienen un plan de asistencia empresarial o empresarial tienen acceso a Trusted Advisor cheques adicionales.

Muchas de estas comprobaciones pueden ayudarle a evaluar el nivel de seguridad de sus recursos de Amazon Pinpoint como parte de su AWS cuenta en general. Por ejemplo, el conjunto básico de comprobaciones de Trusted Advisor incluye lo siguiente:

- Registra las configuraciones de su AWS cuenta, para cada AWS región compatible.
- Permisos de acceso para los buckets de Amazon Simple Storage Service (Amazon S3), que es posible que contengan archivos que importa a Amazon Pinpoint para generar segmentos.
- Uso de AWS Identity and Access Management usuarios, grupos y funciones para controlar el acceso a los recursos de Amazon Pinpoint.
- Configuraciones de IAM y ajustes de políticas que podrían comprometer la seguridad de su AWS entorno y de los recursos de Amazon Pinpoint.

Para obtener más información, consulte AWS Trusted Advisor en la Guía del usuario de Soporte .

Validación de la conformidad para Amazon Pinpoint

Los auditores externos evalúan la seguridad y la conformidad de Amazon Pinpoint en varios programas de conformidad de AWS . Estos incluyen los controles de AWS sistema y organización (SOC), FedRAMP, HIPAA, 27001:2013 para los controles de gestión de la seguridad ISO/IEC , 27017:2015 para los controles específicos de la nube ISO/IEC , 27018:2014 para la protección de datos personales, 9001:2015 para los sistemas de gestión de la ISO/IEC calidad, y otros. ISO/IEC

https://aws.amazon.com/compliance/services-in-scope/ Para obtener información general, consulte los programas de AWS cumplimiento, los programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de conformidad al utilizar Amazon Pinpoint viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en la seguridad y el cumplimiento. AWS
- Documento técnico sobre cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- AWS recursos de cumplimiento: esta colección de trabajo y guías puede aplicarse a su sector y ubicación.

- Evaluación de los recursos con las reglas de la Guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub CSPM— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad.

Amazon Pinpoint es un servicio que cumple con los requisitos de la AWS HIPAA cuando los clientes utilizan los canales de comunicación adecuados. Si desea utilizar Amazon Pinpoint para ejecutar cargas de trabajo que contengan información médica protegida (PHI) tal como se define en HIPAA y la legislación y reglamentos asociados, debe utilizar el canal de correo electrónico, el canal de notificaciones de inserción o el canal de SMS para enviar mensajes que contengan PHI. Si utiliza el canal de SMS para enviar mensajes que contienen PHI, debe enviarlos desde un código abreviado específico que haya solicitado para su AWS cuenta con el propósito explícito de enviar mensajes que contengan o puedan contener PHI. El canal de voz no cumple con los requisitos de la AWS HIPAA; no utilice el canal de voz para enviar mensajes que contengan PHI.

Resiliencia de Amazon Pinpoint

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información, consulte sobre las arquitecturas de referencia, consulte la Guía de arquitectura resiliente de Amazon Pinpoint.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte la infraestructura global.AWS

Seguridad de la infraestructura en Amazon Pinpoint

Como servicio gestionado, Amazon Pinpoint está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la

infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a Amazon Pinpoint a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Aunque puede realizar estas llamadas a la API desde cualquier ubicación de red, Amazon Pinpoint admite políticas de acceso basadas en recursos. Estas política pueden incluir restricciones basadas en la dirección IP de origen. Para obtener más información sobre este tipo de política, consulte Administración del acceso con políticas.

Además, puede configurar y utilizar diversas funciones de AWS seguridad para controlar el acceso a los recursos de Amazon Pinpoint desde cualquier aplicación móvil o web que integre con Amazon Pinpoint. Esto incluye restricciones en las llamadas a la API para tareas como agregar puntos de enlace, actualizar datos de punto de enlace, enviar datos de eventos y registrar datos de uso.

Para utilizar estas funciones, le recomendamos que utilice el AWS dispositivo móvil SDKs o AWS Amplify JavaScript las bibliotecas para integrar las aplicaciones móviles y web con Amazon Pinpoint. Para las aplicaciones de Android o iOS, le recomendamos que utilice el AWS Mobile SDK para Android o el AWS Mobile SDK for iOS, respectivamente. Para aplicaciones móviles o web JavaScript basadas en aplicaciones web, te recomendamos que utilices la AWS Amplify JavaScript Biblioteca para la Web o la AWS Amplify JavaScript Biblioteca para React Native. Para obtener más información sobre estos recursos, consulta Cómo empezar con el AWS móvil SDKs, Cómo empezar con la biblioteca AWS Amplify para la web y Cómo empezar con la biblioteca AWS Amplify para React native.

Configuración y análisis de vulnerabilidades en Amazon Pinpoint

Como servicio gestionado, Amazon Pinpoint está protegido por los procedimientos de seguridad de la red AWS global que se describen en el documento técnico Amazon Web Services: descripción general de los procesos de seguridad. Esto significa que AWS administra y lleva a cabo tareas y procedimientos de seguridad básicos para reforzar, parchear, actualizar y mantener la infraestructura

subyacente de su cuenta y sus recursos de Amazon Pinpoint. Estos procedimientos han sido revisados y certificados por los terceros pertinentes.

Para obtener más información, consulte los siguientes recursos:

- Validación de la conformidad para Amazon Pinpoint
- Modelo de responsabilidad compartida
- Amazon Web Services: Información general de procesos de seguridad (documento técnico)

Prácticas recomendadas de seguridad para Amazon Pinpoint

Utilice las cuentas de AWS Identity and Access Management (IAM) para controlar el acceso a las operaciones de la Amazon Pinpoint API, especialmente a las operaciones que crean, modifican o eliminan recursos. Amazon Pinpoint En el caso Amazon Pinpoint de la API, estos recursos incluyen proyectos, campañas y viajes. En el caso de la API de SMS y voz de Amazon Pinpoint , estos recursos incluyen números de teléfono, grupos y conjuntos de configuraciones.

- Cree un usuario individual para cada persona que administre Amazon Pinpoint los recursos, incluido usted. No utilice credenciales AWS raíz para administrar los recursos de Amazon Pinpoint.
- Asigne a cada usuario el conjunto mínimo de permisos requerido para realizar sus tareas.
- Use los grupos de IAM para administrar con eficacia los permisos para varios usuarios.
- Rote con regularidad sus credenciales de IAM.

Para obtener más información acerca Amazon Pinpoint de la seguridad, consulte Seguridad en Amazon Pinpoint. Para obtener más información acerca de IAM, consulte AWS Identity and Access Management. Para obtener información acerca de las prácticas recomendadas de IAM, consulte Prácticas recomendadas de IAM.

Cuotas de Amazon Pinpoint

En las siguientes secciones se muestran y describen las cuotas, antes denominadas límites, que se aplican a los recursos y las operaciones de Amazon Pinpoint. Algunas cuotas pueden aumentarse, mientras que otras no. Para determinar si puede solicitar un aumento de una cuota, consulte la columna o instrucción Es posible aumentarlo en cada sección.

Temas

- Cuotas de proyectos
- Cuotas de solicitudes de API
- Cuotas de campaña
- Cuotas de correo electrónico
- Cuotas de punto de enlace
- Cuotas de importación de puntos de enlace
- Cuotas de consumo de eventos
- Cuotas de recorrido
- Cuotas de Lambda
- Cuotas de aprendizaje automático
- Cuotas de plantillas de mensajes
- Cuotas de notificaciones push
- Cuotas de mensajes integrados en la aplicación
- Cuotas de segmentos
- Cuotas de SMS
- Cuotas de 10DLC
- Cuotas de voz
- Solicitud de aumento de cuota

Cuotas de proyectos

En la siguiente tabla se muestran las cuotas relacionadas con los proyectos en Amazon Pinpoint.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Proyectos	En cada uno Región de AWS, puedes tener hasta 100 proyectos.	No

Cuotas de solicitudes de API

Amazon Pinpoint implementa cuotas que restringen el tamaño y la cantidad de solicitudes que puede realizar a la API de Amazon Pinpoint desde su cuenta. AWS

El tamaño máximo de una carga de invocación (solicitud y respuesta) es de 7 MB, a menos que se especifique lo contrario para un tipo concreto de recurso. Para determinar si un recurso tiene una cuota diferente, consulte la sección correspondiente de este tema para ese tipo de recurso.

El número máximo de solicitudes varía en función del tipo de cuota y el funcionamiento de la API. Amazon Pinpoint implementa dos tipos de cuotas para las solicitudes de API:

- Cuotas de tasas: también conocidas como límites de tasas, este tipo de cuota define el número máximo de solicitudes que puede realizar por segundo para una operación concreta. Controla la tasa de solicitudes que se envían o reciben por cuenta.
- Cuotas de ráfaga: también denominadas límites de ráfaga o capacidad de ampliación, este tipo de cuota define el número máximo de solicitudes que se están tramitando simultáneamente para una cuenta.

En la siguiente tabla se muestran las cuotas de velocidad y ráfaga para la API de Amazon Pinpoint.

Operación	burst/rate Cuota predeterminada (solicitudes por segundo)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10

Operación	burst/rate Cuota predeterminada (solicitudes por segundo)
CreateImportJob	300
CreatePushTemplate	10
CreateSegment	25
CreateSmsTemplate	10
CreateVoiceTemplate	10
DeleteCampaign	25
DeleteEndpoint	5
DeleteSegment	25
GetEndpoint	10
PhoneNumberValidate	20
PutEvents	15
SendMessages	4.000
SendUsersMessages	6000
UpdateCampaign	25
UpdateEmailTemplate	10
UpdateEndpoint	10
UpdateEndpointsBatch	2
UpdateInAppTemplate	10
UpdatePushTemplate	10
UpdateSegment	25

Operación	burst/rate Cuota predeterminada (solicitudes por segundo)
UpdateSmsTemplate	10
UpdateVoiceTemplate	10
Otras operaciones	300

En la siguiente tabla se muestra la cuota de importación de archivos para `CreateImportJob`.

Operación	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de archivos de importación	10 000 archivos por trabajo de importación	No

Al superar una de esas cuotas, Amazon Pinpoint limita de forma controlada la solicitud, es decir, rechaza una solicitud válida en todo caso y devuelve un error `TooManyRequests`. La limitación se basa en el número total de solicitudes que realice desde la cuenta para una operación específica en una Región de AWS específica. Además, las decisiones de limitación se calculan de forma independiente para cada operación. Por ejemplo, si Amazon Pinpoint limita una solicitud para la operación `SendMessages`, una solicitud simultánea para la operación `UpdateEndpoint` puede completarse correctamente.

Cuotas de campaña

Las siguientes cuotas se aplican al recurso Campañas de la API de Amazon Pinpoint.

Las siguientes cuotas se aplican cada cierto Región de AWS tiempo y algunas se pueden aumentar. Para obtener más información, consulte Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Campañas activas	200 por cuenta	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
	Note Una campaña activa es una campaña que no se ha completado o que ha fallado. Las campañas activas tienen un estado `SCHEDULED`, `EXECUTING` o `PENDING_NEXT_RUN`.	
Tamaño máximo del segmento	Para segmentos importados: 100 000 000 por campaña Para segmentos dinámicos: ilimitado	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Campañas basadas en eventos	Cada proyecto puede incluir hasta 25 campañas que se envían cuando se producen eventos. Las campañas que utilizan desencadenadores basados en eventos tienen que utilizar segmentos dinámicos. No pueden usar segmentos importados. Si integra su aplicación con Amazon Pinpoint mediante un SDK para AWS dispositivos móviles, los mensajes de las campañas basadas en eventos se envían únicamente a los clientes cuyas aplicaciones ejecuten la AWS Mobile SDK para Android versión 2.7.2 o posterior, o la versión 2.6.30 o AWS Mobile SDK for iOS posterior. Si Amazon Pinpoint no puede entregar un mensaje de una campaña basada en eventos en el plazo de cinco minutos, descarta el mensaje y no vuelve a intentar entregarlo.	No

Cuotas de correo electrónico

Las cuotas de las siguientes secciones se aplican al canal de correo electrónico.

Cuotas de mensajes de correo electrónico

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Tamaño máximo de los mensajes (incluidos los archivos adjuntos)	10 MB por mensaje	No
Número de identidades verificadas	10 000 identidades **Note** Las identidades son las direcciones de correo electrónico o dominios, o bien una combinación de ambos. Todos los correos electrónicos que envíe a través de Amazon Pinpoint se deben hacer desde una identidad verificada.	No

Cuotas de los remitentes y destinatarios de correo electrónico

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Dirección del remitente	Es necesario verificar todos los dominios y direcciones de los remitentes.	No
Dirección del destinatario	Si su cuenta está en el entorno de pruebas, se deben verificar todos los dominios o direcciones de correo electrónico de los destinatarios. Si su cuenta no está en el entorno de pruebas, puede realizar envíos a cualquier dirección válida.	Sí
Número de destinatarios por mensaje	50 destinatarios por mensaje	No
Número de identidades que puede verificar	10 000 identidades por región AWS **Note** Las identidades son las direcciones de correo electrónico o dominios, o bien una combinación de ambos. Todos los correos electrónicos que envíe a través de Amazon Pinpoint se deben hacer desde	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
	una identidad verificad a.	

Cuotas de envío de correo electrónico

La cuota de envío, la tasa de envío y los límites del entorno aislado se comparten entre los dos servicios de la misma región. Si utilizas Amazon SES en us-east-1 y te han retirado del entorno de pruebas y te han aumentado los quota/rate envíos, todos esos cambios se aplicarán a tu cuenta de Pinpoint en us-east-1.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número de correos electróni cos que pueden enviarse cada 24 horas (cuota de envío)	Si la cuenta está en el entorno de pruebas: 200 correos electrónicos por un período de 24 horas. Si su cuenta está fuera del entorno de pruebas, la cuota varía en función de su caso de uso específico. **Note** Esta cuota se basa en el número de destinata rios, en lugar de la cantidad de mensajes enviados únicos. Un destinatario es cualquier dirección de	Sí

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
	correo electrónico en el campo Para:.	
Número de correos electrónicos que pueden enviarse cada segundo (tasa detasa de envío)	Si la cuenta está en el entorno de pruebas: 1 correo electrónico por segundo. Si su cuenta está fuera del entorno de pruebas, la tasa varía en función de su caso de uso específico. **Note** Esta tasa se basa en el número de destinatarios, en lugar de la cantidad de mensajes enviados únicos. Un destinatario es cualquier dirección de correo electrónico en el campo Para:.	Sí

Cuotas de punto de enlace

Las siguientes cuotas se aplican al recurso Puntos de conexión de la API de Amazon Pinpoint.

El número máximo de atributos admitidos por punto de conexión es de 250 y el tamaño máximo del punto de conexión es de 15 KB. Sin embargo, es posible que este número de atributos esté limitado por el tamaño total de un punto de conexión, que incluye todos los atributos. Si se produce algún error al agregar atributos a la plantilla, considere la posibilidad de reducir la cantidad de datos de cada atributo o disminuir el número de atributos.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Tamaño de punto de conexión	Tamaño máximo: 15 GB	No
Atributos asignados a los parámetros `Attributes` , `Metrics` y `UserAttributes` conjuntamente	250 para todos los parámetros de atributo por aplicación	No
Atributos asignados al parámetro `Attributes`	250 para todos los parámetros de atributo por aplicación	No
Atributos asignados al parámetro `Metrics`	250 para todos los parámetros de atributo por aplicación	No
Atributos asignados al parámetro `UserAttributes`	250 para todos los parámetros de atributo por aplicación	No
Longitud del nombre de atributo	50 caracteres	No
Longitud del valor de atributo	100 caracteres	No
`EndpointBatchItem` objetos en una carga de `EndpointBatchRequest`	100 por carga. El tamaño de la carga no puede ser superior a 7 MB.	No
Puntos de enlace con el mismo ID de usuario	15 puntos de conexión únicos por ID de usuario	No
Valores asignados a los atributos de parámetro `Attributes`	50 por atributo	No
Valores asignados a los atributos de parámetro `UserAttributes`	50 por atributo	No

Cuotas de importación de puntos de enlace

Las siguientes cuotas se aplican a la importación de puntos de conexión en Amazon Pinpoint.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Trabajos de importación activos	10 por cuenta Los trabajos de importación solo se tienen en cuenta para esta cuota si están en ejecución. Una vez que el trabajo de importación se haya completado, ya no se tendrá en cuenta para esta cuota.	No
Tamaño de importación	1 GB por trabajo de importación Por ejemplo, si cada punto de enlace es de 4 KB o menos, puede importar 250 000 puntos de enlace.	No

Cuotas de consumo de eventos

Las siguientes cuotas se aplican a la ingesta de eventos mediante el recurso AWS móvil SDKs y el recurso Events de la API Amazon Pinpoint.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de tipos de eventos personalizados	1500 por aplicación	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de claves de atributos personalizados	500 por aplicación	No
Número máximo de valores de atributos personalizados por clave de atributo	100 000. Cualquier número que supere los 100 000 seguirá registrado, pero no estará disponible en la consola de análisis de Amazon Pinpoint.	No
Número máximo de caracteres por clave de atributo	50	No
Número máximo de caracteres por valor de atributo	200. Si el número de caractere s supera los 200, el evento se descarta.	No
Número máximo de claves de métrica personalizada	500 por aplicación	No
Número máximo de eventos en una solicitud.	100 por solicitud	No
Tamaño máximo de una solicitud	4 MB	No
Tamaño máximo de un evento individual	1,000 KB	No
Número máximo de claves de atributo y claves de métricas para cada evento	40 por solicitud	No

Cuotas de recorrido

Las siguientes cuotas se aplican a los recorridos.

Se aplican las siguientes cuotas por año Región de AWS , algunas de las cuales pueden aumentarse. Para obtener más información, consulte Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de recorridos activos	50 por cada cuenta de	No
Número máximo de activos EventTriggeredJourneys	20 por cuenta	No
Número máximo de actividades de recorrido	40 por recorrido	No
Tamaño máximo del segmento	Para segmentos importados: 100 000 000 por recorrido. Para segmentos dinámicos: ilimitado	No
Número máximo de actividades de centro de contacto	3 por recorrido	No
Reglas de número máximo de días cerrado	20 por canal	No
Nombre de regla de duración máxima de día cerrado	150 caracteres	No
Número máximo de días entre la hora de inicio y la hora de finalización para una regla de días cerrado	7 días	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Reglas de número máximo de horas abierto	4 por día	No

Cuotas de Lambda

Las siguientes cuotas se aplican a las configuraciones de Amazon Pinpoint para recuperar y procesar datos de Lambda

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Tamaño máximo de una carga de invocación (solicitud y respuesta) para una función de Lambda	6 MB	No
Cantidad máxima de tiempo de espera para que una función de Lambda procese los datos	15 segundos	No
Número máximo de atributos de eventos por punto de conexión	5	No
Número máximo de caracteres para el nombre de un atributo de evento	128 caracteres	No
Número máximo de caracteres para el valor de un atributo de evento	128 caracteres	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Cantidad máxima de días que se puede ejecutar un recorrido	540 días	No

Cuotas de aprendizaje automático

Las siguientes cuotas se aplican a las configuraciones de Amazon Pinpoint para recuperar y procesar datos de modelos de machine learning (ML).

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de configuraciones de modelo	1 por plantilla de mensaje 100 por cuenta	No
Número máximo de recomendaciones	5 por punto de enlace o usuario	No
Número máximo de atributos recomendados por punto de conexión o usuario	1, si los valores de atributo no son procesados por una función de AWS Lambda 10, si los valores de atributo son procesados por una función de AWS Lambda	No
Longitud máxima de un nombre de atributo recomendado	50 caracteres para un nombre de atributo 25 caracteres para un nombre para mostrar de atributo (el nombre que aparece en el buscador de atributos de la consola)	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Longitud máxima de un valor de atributo recomendado que se recupera de Amazon Personalize	100 caracteres	No
Tamaño máximo de una carga de invocación (solicitud y respuesta) para una función de Lambda	6 MB	No
Cantidad máxima de tiempo de espera para que una función de Lambda procese los datos	15 segundos	No
Número máximo de intentos de invocar una función de Lambda	3 intentos	No

En función de cómo configure Amazon Pinpoint para usar un modelo de ML, pueden aplicarse cuotas adicionales. Para obtener información sobre las cuotas de Amazon Personalize, consulte Cuotas en la Guía para desarrolladores de Amazon Personalize. Para obtener más información sobre las cuotas de AWS Lambda , consulte Cuotas en la Guía para desarrolladores de AWS Lambda .

Cuotas de plantillas de mensajes

Las siguientes cuotas se aplican a las plantillas de mensajes de la cuenta de Amazon Pinpoint.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de plantillas de mensajes	20 000 por cuenta	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de versiones	5000 por plantilla	No
Número máximo de caracteres en una plantilla de correo electrónico	600 000 caracteres	No
Número máximo de caracteres en una plantilla integrada en la aplicación	200 000 caracteres	No
Número máximo de caracteres en las partes predeterminadas de una plantilla de notificación push	4000 caracteres	No
Número máximo de caracteres en las partes de la plantilla específicas de ADM de una plantilla de notificación push	6000 caracteres	No
Número máximo de caracteres en partes APNs específicas de una plantilla de notificaciones push	4000 caracteres	No
Número máximo de caracteres en las partes de la plantilla específicas de Baidu de una plantilla de notificación push	4000 caracteres	No
Número máximo de caracteres en las partes de la plantilla específicas de FCM de una plantilla de notificación de inserción	4000 caracteres	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de caracteres en una plantilla de SMS	1600 caracteres	No
El número máximo de caracteres en una plantilla de voz	10,000 caracteres	No

Cuotas de notificaciones push

Las siguientes cuotas se aplican a los mensajes que envía Amazon Pinpoint a través de canales de notificación de inserción.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de notificaciones de inserción que se pueden enviar por segundo en una campaña	25 000 notificaciones por segundo	Sí
Tamaño de carga de mensajes de Amazon Device Messaging (ADM)	6 KB por mensaje	No
Tamaño de carga útil de los mensajes del servicio de notificaciones push de Apple (APNs)	4 KB por mensaje	No
APNs tamaño de carga útil de los mensajes de entorno aislado	4 KB por mensaje	No

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Tamaño de carga de mensajes de Baidu Cloud Push	4 KB por mensaje	No
Tamaño de la carga de los mensajes de Firebase Cloud Messaging (FCM)	4 KB por mensaje	No

Cuotas de mensajes integrados en la aplicación

La siguiente cuota se aplica a los mensajes integrados en la aplicación que administre con Amazon Pinpoint.

Se aplican las siguientes cuotas Región de AWS y algunas se pueden aumentar. Para obtener más información, consulte Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de veces que puede llamar a la API de `GetInAppMessages` por segundo.	5000 solicitudes por segundo	Sí
Campañas de mensajería integradas en la aplicación	Cada proyecto puede incluir hasta 25 campañas que utilicen el canal de mensajería integrado en la aplicación.	Sí, consulte Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas

Cuotas de segmentos

Las siguientes cuotas se aplican al recurso Segmentos de la API de Amazon Pinpoint.

Recurso	Cuota predeterminada	Puede optar a un aumento de la cuota
Número máximo de dimensiones que se pueden utilizar para crear un segmento	100 por segmento	No
Número máximo de grupos de segmentos por segmento	5	No
Número máximo de segmentos de origen por segmento	5	No
Profundidad máxima de los segmentos de origen. Por ejemplo, si un segmento tiene un segmento de origen que también tiene un segmento de origen, la cadena de profundidad no supera este límite.	5	No

Cuotas de SMS

Para conocer las cuotas de SMS, consulte las cuotas de SMS en la guía del usuario de mensajería de SMS para usuarios AWS finales.

Cuotas de 10DLC

Para conocer las cuotas de 10 DLC, consulta las cuotas de 10 DLC en la guía del usuario de SMS para usuarios AWS finales.

Cuotas de voz

Para conocer las cuotas de voz, consulte Cuotas de voz en la guía del usuario de mensajería de SMS para usuarios AWS finales.

Solicitud de aumento de cuota

Si el valor de la columna Eligible for Increase (Es posible aumentarlo) de cualquiera de las tablas anteriores es Yes (Sí), puede solicitar un cambio de esa cuota.

Cómo solicitar un aumento de cuota

1. Inicie sesión Consola de administración de AWS en https://console.aws.amazon.com/.
2. Cree un nuevo caso de AWS Support en https://console.aws.amazon.com/support/home#/case/create.
3. En la pestaña Casos de soporte abiertos, elija Crear caso.
4. Elija el enlace ¿Búsqueda de aumentos del límite de servicio?.
5. En Aumento de la cuota de servicio, en Servicio, elija una de las siguientes opciones:
   - Para solicitar un aumento de cuota relacionado con el canal de correo electrónico, elija Pinpoint Email (Correo electrónico de Pinpoint).
   - Para solicitar un aumento de cuota para los límites de gasto de SMS o las tasas de envío de SMS, elija SMS de Pinpoint. Para todos los demás aumentos de cuota de SMS, elegir Pinpoint
   - Para solicitar un aumento de cuota relacionado con el canal de voz, elija Pinpoint Voice (Voz de Pinpoint).
   - Solicitud de un aumento de cuota relacionado con cualquier otra característica de Amazon Pinpoint, elija Pinpoint.
6. Según el servicio que elija, es posible que se le pida que introduzca lo siguiente:
   - (Opcional) En Proporcionar un enlace al sitio o aplicación que enviará los mensajes SMS, proporcione información sobre el sitio web, la aplicación o el servicio que enviará los mensajes SMS.
   - (Opcional) En Tipo de mensaje que tiene previsto enviar, elija el tipo de mensaje que tiene previsto enviar con el código largo:
     - Contraseña de un solo uso: mensajes que proporcionan contraseñas que sus clientes utilizan para autenticarse con su sitio web o aplicación.

    - Promocional: mensajes no importantes que promocionan su empresa o servicio, tales como anuncios u ofertas especiales.
    - Transaccional: mensajes informativos importantes que admiten transacciones del cliente, tales como confirmaciones de pedido o alertas de transacción. Los mensajes transaccionales no pueden incluir contenido promocional ni de marketing.
  - (Opcional) Para saber desde qué AWS región vas a enviar los mensajes, selecciona la región desde la que vas a enviar los mensajes.
  - (Opcional) En Países a los que tiene previsto enviar mensajes, introduzca el país o la región en el que quiere comprar códigos cortos.
  - (Opcional) En Cómo deciden sus clientes recibir mensajes suyos, facilite detalles sobre su proceso de suscripción.
  - (Opcional) En el campo Indique la plantilla de mensajes que piensa utilizar para enviar mensajes a sus clientes, incluya la plantilla que vaya a utilizar.
7. En Requests (Solicitudes), realice el siguiente procedimiento:
   - En Región, seleccione Región de AWS.
   - En Tipo de recurso, elija Límites generales. El campo Tipo de recurso solo está presente en algunos Servicios.
   - En Cuota, elija la cuota que desee cambiar.
   - En Nuevo valor de cuota, ingrese un valor nuevo para la cuota.
   - Para solicitar un aumento a la misma cuota en una solicitud adicional Región de AWS, selecciona Añadir otra solicitud y, a continuación, selecciona la adicional Región de AWS y rellena la nueva solicitud.
8. Elija la cuota que desea aumentar y, a continuación, escriba el nuevo valor que desea para la cuota.
9. En Descripción del caso, explique por qué solicita el aumento de cuota.
10. En Opciones de contacto, en Idioma de contacto preferido, elija el idioma que prefiera usar al comunicarse con el equipo de AWS Support.
11. Para el método de contacto, elija el método que prefiera para comunicarse con el equipo de AWS Support.
12. Seleccione Enviar.

El equipo de AWS Support proporciona una respuesta inicial a su solicitud en un plazo de 24 horas.

Para evitar que nuestros sistemas sean utilizados para enviar contenido no solicitado o malicioso, tenemos que considerar cada solicitud detenidamente. Si podemos hacerlo, garantizaremos una solicitud en este periodo de 24 horas. Sin embargo, si tenemos que obtener información adicional suya, puede que la solicitud tarde más tiempo en concederse.

Es posible que no podamos garantizar su solicitud si su caso de uso no está en consonancia con nuestras políticas.

Historial de documentos para Amazon Pinpoint

En la siguiente tabla se describen los cambios importantes en cada versión de la Guía para desarrolladores de Amazon Pinpoint después de diciembre de 2018. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

- Última actualización de la documentación: 16 de noviembre de 2023

Cambio	Descripción	Fecha
Notificación del fin del soporte	Aviso de fin de soporte: el 30 de octubre de 2026, AWS finalizará el soporte para Amazon Pinpoint. Después del 30 de octubre de 2026, ya no podrá acceder a la consola de Amazon Pinpoint ni a los recursos de Amazon Pinpoint (puntos de conexión, segmentos, campañas, recorridos y análisis). Para obtener más información, consulte Fin de soporte de Amazon Pinpoint. Nota: en lo APIs que respecta a los SMS, este cambio no afecta a los mensajes de voz, a las notificaciones push móviles, a las OTP y a la validación de números de teléfono, y son compatibles con la mensajería para el usuario AWS final.	20 de mayo de 2025
Se ha añadido CloudTrail soporte para el envío de mensajes	Se agregó compatibilidad con el CloudTrail PutEvents registro y las llamadas a la	17 de febrero de 2025

	SendMessages API. Consulte Acciones de la API de Amazon Pinpoint compatibles en los archivos de CloudTrail registro. SendUserMessages	
Amazon Pinpoint ha actualizado la documentación de su guía del usuario	Para obtener la información más reciente sobre cómo crear, configurar y administrar recursos push, consulte la nueva Guía del usuario de Mensajes push para usuarios finales.	22 de julio de 2024
Encabezados de correo electrónico	Puede agregar encabezados de correo electrónico a sus mensajes de correo electrónico. Para obtener más información, consulte cómo enviar un correo electrónico con encabezados de cancelación de suscripción.	7 de mayo de 2024
Orquestación de correo electrónico	Amazon Pinpoint ha actualizado la forma en que utiliza los recursos de Amazon SES para enviar correos electrónicos. Para obtener más información, consulte Rol de IAM para el envío de correos electrónicos con Amazon SES.	30 de abril de 2024

Amazon Pinpoint ha actualizado la documentación de su guía del usuario	Los temas de administración de recursos de SMS y voz ahora se han redirigido a la Guía del usuario de Mensajes SMS para usuarios finales de AWS . Para obtener más información, consulte la Guía del usuario de Mensajes SMS para usuarios finales de AWS .	8 de febrero de 2024
Cuotas de Amazon Pinpoint	Se agregaron cuotas para Reglas de número máximo de días cerrado, Nombre de regla de duración máxima de día cerrado, Número máximo de días entre hora de inicio y de finalización para una regla de días cerrado y Reglas de número máximo de horas abierto. Para obtener más información, consulte Cuotas de Amazon Pinpoint.	19 de diciembre de 2023
Amazon Pinpoint ha actualizado la documentación de su guía del usuario	Para obtener la información más reciente sobre cómo crear, configurar y administrar sus recursos de mensajería, SMS y voz para usuarios AWS finales, consulte la nueva guía del usuario de mensajes SMS para usuarios AWS finales.	16 de noviembre de 2023

Cuotas de Amazon Pinpoint	Se actualizaron las cuotas de UpdateEndpointsBatch UpdateEndpoint PutEvents, DeleteEndpoint, y GetEndpoint. Para obtener más información, consulte Cuotas de Amazon Pinpoint.	22 de septiembre de 2023
Cuotas de Amazon Pinpoint	Se actualizaron las cuotas de CreateEmailTemplate CreateSmsTemplate CreatePushTemplate,, CreateInAppTemplate, CreateVoiceTemplate, UpdateEmailTemplate, UpdateSmsTemplate, UpdatePushTemplate, UpdateInAppTemplate, UpdateVoiceTemplate y CreateImportJob. Para obtener más información, consulte Cuotas de Amazon Pinpoint.	12 de septiembre de 2023
Métricas sobre la ejecución del recorrido y la campaña	Se han agregado nuevas métricas analíticas para los recorridos y las campañas. Para obtener más información, consulte Métricas de ejecución de recorrido y campaña.	25 de abril de 2023
Creación de un punto de conexión de VPC de interfaz para Amazon Pinpoint	Amazon Pinpoint ahora admite puntos de conexión de VPC de interfaz. Para obtener más información, consulte Creación de un punto de conexión de VPC para Amazon Pinpoint.	11 de abril de 2023

Cifrado en tránsito	A partir del 22/03/2023, Amazon Pinpoint ya no admitirá la versión 1.0 de TLS, pero podrá seguir utilizando la versión 1.2 o versiones posteriores. Para obtener más información, consulte Cifrado en tránsito.	20 de marzo de 2023
Cuotas de Amazon Pinpoint	Se ha actualizado el proceso para solicitar un aumento de cuota para las campañas, los recorridos y los mensajes integrados en la aplicación. Para obtener más información, consulte Cuotas de Amazon Pinpoint.	16 de diciembre de 2022
Disponibilidad regional	Amazon Pinpoint ya está disponible en esta región: Este de EE. UU. (Ohio).	5 de octubre de 2022
Actualizaciones de ejemplos de rol de IAM	Se han actualizado varios ejemplos de roles de IAM a lo largo del documento para alinearlos mejor con las prácticas recomendadas de seguridad.	27 de mayo de 2022

API de SMS y voz, versión 2	Amazon Pinpoint ahora incluye una API dedicada para enviar mensajes SMS y de voz. Esta API incluye nuevas características, como conjuntos de configuraciones, grupos y listas de exclusión, que son útiles para los clientes que envían mensajes SMS y de voz mediante transacciones. Para obtener más información, consulte Uso de la API de SMS y voz de Amazon Pinpoint.	1 de abril de 2022
Creación y validación de contraseñas únicas	Amazon Pinpoint ahora incluye una función que genera contraseñas de un solo uso (OTPs) y las envía a los usuarios como mensajes SMS. También incluye una API para validar los códigos de OTP cuando los usuarios los ingresan en la aplicación o sitio. Para obtener más información, consulte Envío y validación de contraseñas de un solo uso () OTPs.	26 de noviembre de 2021
Mensajes en la aplicación	Se ha agregado información sobre la integración de la capacidad de mensajería integrada de Amazon Pinpoint con las aplicaciones.	10 de noviembre de 2021

Ejemplos de código	Se ha agregado una biblioteca de ejemplos de código para operaciones comunes de Amazon Pinpoint.	3 de noviembre de 2021
Cuotas de proyectos	El número máximo de proyectos de Amazon Pinpoint sigue siendo de 100, pero ahora se puede aumentar esta cuota abriendo una solicitud de aumento del límite de servicio con. Soporte	11 de octubre de 2021
Actualizaciones de la política de Lambda.	Determinadas políticas de permisos de Lambda ahora deben incluir una condición `AWS:SourceAccount` . Se han actualizado las políticas de ejemplo en los temas Creación de canales personalizados en Amazon Pinpoint y Personalización de segmentos con AWS Lambda para cumplir con este requisito.	7 de octubre de 2021
UpdateEndpoint	La UpdateEndpoint API de Amazon Pinpoint ahora está registrada por. CloudTrail	16 de noviembre de 2020
Custom attributes (Atributos personalizados)	Amazon Pinpoint ahora admite 250 atributos en plantillas de mensajería de correo electrónico. Consulte las cuotas.	18 de septiembre de 2020

Disponibilidad regional	Amazon Pinpoint ya está disponible en estas regiones: región Asia Pacífico (Tokio), región Europa (Londres) y región Canadá (centro). Tenga en cuenta que la API de SMS y voz de Amazon Pinpoint no está disponible en estas regiones.	10 de septiembre de 2020
Disponibilidad regional	Ahora Amazon Pinpoint está disponible en la región de Asia-Pacífico (Tokio). Tenga en cuenta que la API de SMS y voz de Amazon Pinpoint no admite voz en esta región.	2 de septiembre de 2020
Eventos de campaña	Se ha agregado información sobre un nuevo parámetro `delivery_type` de evento de campaña a los eventos de campaña.	2 de agosto de 2020
Disponibilidad regional	Ahora Amazon Pinpoint está disponible en la región de Asia-Pacífico (Seúl). Tenga en cuenta que la API de Amazon Pinpoint no admite voz ni SMS en esta región.	31 de julio de 2020
Disponibilidad regional	Amazon Pinpoint ya está disponible en la AWS GovCloud (US) región.	30 de abril de 2020

Canales personalizados	Información actualizada sobre la creación de canales personalizados mediante funciones de Lamda o webhooks.	23 de abril de 2020
Machine learning	Se agregó información sobre cómo recuperar recomenda ciones personalizadas de los modelos de recomenda ciones y, si lo desea, cómo mejorar esas recomenda ciones mediante el uso de funciones. AWS Lambda	4 de marzo de 2020
Seguridad	Se ha agregado un capítulo Seguridad, que proporcio na información acerca de distintos controles de seguridad y características de Amazon Pinpoint.	4 de febrero de 2020
Recorridos	Se ha agregado informaci ón sobre el uso de recorrido s de Amazon Pinpoint para desarrollar flujos de trabajo automatizados que realizan actividades de mensajerí a para proyectos. También se ha agregado información sobre la consulta de datos analíticos para un subconjunto de métricas que se aplican a los recorridos.	31 de octubre de 2019

Análisis	Se han agregado procedimientos que explican cómo consultar datos de análisis de campañas y mensajes transaccionales, e información adicional sobre el uso de los resultados de las consultas.	17 de octubre de 2019
Análisis	Se ha añadido información acerca de la consulta de datos de análisis para un subconjunto de métricas que se aplican al correo electrónico y a los mensajes SMS de transacciones.	6 de septiembre de 2019
Ejemplos de código	Se han agregado ejemplos de código que puede utilizar para enviar notificaciones de inserción transaccionales con todos los servicios compatibles con Amazon Pinpoint.	30 de julio de 2019
Análisis	Se ha añadido información sobre cómo consultar datos de análisis para un subconjunto de métricas que se aplican a los proyectos (aplicaciones) y a las campañas.	24 de julio de 2019
Segmentos	Se ha agregado un tutorial que describe una solución para importar los datos de los clientes a Amazon Pinpoint desde sistemas externos, como Salesforce o Marketo.	14 de mayo de 2019

Disponibilidad regional	Amazon Pinpoint ya está disponible en las regiones de AWS Asia Pacífico (Bombay) y Asia Pacífico (Sídney).	25 de abril de 2019
Uso de Postman con Amazon Pinpoint	Se ha agregado un tutorial que describe cómo utilizar Postman para interactuar con la API de Amazon Pinpoint.	8 de abril de 2019
Etiquetado	Se ha agregado informaci ón sobre el etiquetado de recursos de Amazon Pinpoint.	27 de febrero de 2019
Registro de SMS	Se ha agregado un capítulo Tutoriales y se ha agregado un tutorial en el que se describe cómo crear una solución que administra el registro de usuario de SMS.	27 de febrero de 2019
Ejemplos de código	Se han añadido ejemplos de código en varios lenguajes de programación que muestran cómo enviar mensajes de correo electrónico, SMS y de voz mediante programación.	6 de febrero de 2019

Actualizaciones anteriores

En la siguiente tabla se describen los cambios importantes en cada versión de la Guía para desarrolladores de Amazon Pinpoint hasta diciembre de 2018.

Cambio	Descripción	Fecha
Disponibilidad regional	Amazon Pinpoint ya está disponible en las regiones	21 de diciembre de 2018

Cambio	Descripción	Fecha
	AWS EE.UU. Oeste (Oregón) y Europa (Fráncfort).	
Canal de voz	Puede utilizar el nuevo canal de voz de Amazon Pinpoint para crear mensajes de voz y, a continuación, entregar esos mensajes a los clientes por teléfono. Actualmente, solo se pueden enviar mensajes de voz si se utiliza la API de SMS y voz de Amazon Pinpoint.	15 de noviembre de 2018
Disponibilidad de Europa (Irlanda)	Amazon Pinpoint ya está disponible en la región de AWS Europa (Irlanda).	25 de octubre de 2018
API de eventos	Utilice la API de Amazon Pinpoint para registrar eventos y asociarlos con puntos de conexión.	7 de agosto de 2018
Ejemplos de código para definir y buscar puntos de enlace	Se añaden ejemplos de código que muestran cómo definir, actualizar, eliminar y buscar puntos de enlace. Se proporcionan ejemplos para la AWS CLI AWS SDK para Java, y la API Amazon Pinpoint. Para obtener más información, consulte Utilizar puntos de conexión para representar a su público en Amazon Pinpoint.	7 de agosto de 2018

Cambio	Descripción	Fecha
Permisos de exportación de puntos de enlace	Configure una política de IAM que le permita exportar puntos de conexión de Amazon Pinpoint a un bucket de Amazon S3.	1 de mayo de 2018
Verificación de números de teléfono para SMS	Use la API de Amazon Pinpoint para verificar un número de teléfono para determinar si se trata de un destino válido para los mensajes SMS.	23 de abril de 2018
Temas actualizados para la integración de Amazon Pinpoint	Integre Amazon Pinpoint con su JavaScript aplicació n Android, iOS AWS SDKs o mediante nuestras bibliotecas.	23 de marzo de 2018
AWS CloudTrail registro	Se agregó información sobre el registro de llamadas a la API Amazon Pinpoint con. CloudTrail	6 de febrero de 2018
Service quotas actualizadas	Se ha actualizado la sección Cuotas con información adicional sobre las cuotas de correo electrónico.	19 de enero de 2018
Versión beta pública de las extensiones de Amazon Pinpoint	Utilice AWS Lambda las funciones para personalizar segmentos o crear canales de mensajería personalizados.	28 de noviembre de 2017

Cambio	Descripción	Fecha
Cuotas de carga de las notificaciones push	Las cuotas incluyen tamaños de carga para los mensajes de inserción de dispositivos móviles.	25 de octubre de 2017
Service quotas actualizadas	Se ha añadido información de canal de correo electrónico y SMS a Cuotas.	9 de octubre de 2017
Inserción en móvil de ADM y Baidu	Actualice el código de la aplicación para admitir notificaciones push de los canales push de Baidu y ADM para móviles.	27 de septiembre de 2017
Eventos de usuario IDs y autenticación con grupos de usuarios de Amazon Cognito.	Si utiliza grupos de usuarios de Amazon Cognito para gestionar el inicio de sesión de los usuarios en sus aplicacio nes móviles, Amazon Cognito asigna los usuarios a los puntos de enlace e informa de los eventos de autenticación IDs a Amazon Pinpoint.	26 de septiembre de 2017
Usuario IDs	Asigne el usuario IDs a los puntos finales para supervisar el uso de las aplicaciones por parte de los usuarios individua les. Se proporcionan ejemplos para AWS Mobile SDKs y SDK for Java.	31 de agosto de 2017

Cambio	Descripción	Fecha
Eventos de autenticación	Genere informes de los eventos de autenticación para conocer la frecuencia con la que los usuarios se autentican en su aplicación. Se ofrecen ejemplos en Notificar eventos de Amazon Pinpoint en la aplicación.	31 de agosto de 2017
Eventos de muestra actualizados	Entre los eventos de ejemplo se incluyen algunos que Amazon Pinpoint transmite para la actividad de SMS y correo electrónico.	08 de junio de 2017
Administración de sesiones de Android	Administre sesiones en Android mediante la utilización de una clase proporcionada por la aplicación de muestra de AWS Mobile Hub .	20 de abril de 2017
Muestras de eventos de monetización actualizadas	El código de muestra se ha actualizado para la notificación de eventos de monetización. .	31 de marzo de 2017
Secuencias de eventos	Puede configurar Amazon Pinpoint para enviar los eventos de aplicación y de campaña a un flujo de Kinesis.	24 de marzo de 2017

Cambio	Descripción	Fecha
Permisos	Consulte Cómo funciona Amazon Pinpoint con IAM para obtener información sobre cómo conceder acceso a Amazon Pinpoint a AWS los usuarios de su cuenta y a los usuarios de su aplicación móvil.	12 de enero de 2017
Disponibilidad general de Amazon Pinpoint	Esta versión presenta Amazon Pinpoint.	1 de diciembre de 2016