



Panduan Developer

Amazon Pinpoint



Amazon Pinpoint: Panduan Developer

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin atau mungkin tidak berafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu Amazon Pinpoint?	1
Fitur Amazon Pinpoint	1
Menentukan segmen audiens	1
Libatkan audiens Anda dengan kampanye pengiriman pesan	1
Mengirim pesan SMS	2
Menganalisis perilaku pengguna	2
Ketersediaan wilayah	2
Tutorial	3
Menggunakan Postman dengan Amazon Pinpoint	3
Tentang tutorial ini tutorial tutorial tutorial tutorial tutorial tutorial	3
Prasyarat	4
Langkah 1: Buat kebijakan dan peran IAM	5
Langkah 2: Mengatur Tukang Pos	9
Langkah 3: Kirim permintaan tambahan	17
Menyiapkan sistem pendaftaran SMS	23
Tentang keikutsertaan ganda	24
Tentang solusi ini	25
Prasyarat	27
Langkah 1: Siapkan Amazon Pinpoint	28
Langkah 2: Buat kebijakan dan peran IAM	34
Langkah 3: Buat fungsi Lambda	37
Langkah 4: Siapkan Amazon API Gateway	49
Langkah 5: Buat dan gunakan formulir web	54
Langkah selanjutnya	62
Integrasi dengan aplikasi Anda	65
Bekerja dengan AWS SDK	66
Menghubungkan aplikasi front end Anda dengan AWS Amplify	67
Langkah selanjutnya	67
Mendaftarkan titik akhir	67
Sebelum Anda memulai	68
AWSSDK seluler	68
AWSMemperkuat	68
Langkah selanjutnya	68
Melaporkan peristiwa	69

Sebelum Anda memulai	70
AWSSDK seluler	70
Web dan bereaksi asli	68
API Amazon Pinpoint	71
Langkah selanjutnya	71
Menangani pemberitahuan push	71
Menyiapkan notifikasi push	71
Menangani notifikasi push	74
Mendefinisikan audiens Anda	75
Menambahkan titik akhir	76
Contoh	76
Informasi terkait	82
Mengaitkan pengguna dengan titik akhir	82
Contoh	83
Informasi terkait	88
Menambahkan sekumpulan titik akhir	88
Contoh	88
Informasi terkait	96
Mengimpor titik akhir	96
Sebelum kamu memulai	97
Contoh	97
Informasi terkait	110
Menghapus titik akhir	110
Contoh	110
Mengelola endpoint anggota audiens	113
Mengakses data pemirsa	114
Mencari titik akhir	115
Contoh	115
Informasi terkait	120
Mengekspor titik akhir	121
Sebelum Anda memulai	121
Contoh	121
Informasi terkait	132
Daftar ID titik akhir	133
Membuat segmen	136
Membangun segmen	136

Membangun segmen dengan AWS SDK for Java	136
Mengimpor segmen	140
Mengimpor segmen	140
Menyesuaikan segmen dengan AWS Lambda	142
Data peristiwa	144
Membuat fungsi Lambda	145
Menetapkan kebijakan fungsi Lambda	147
Menetapkan fungsi Lambda ke kampanye	150
Membuat kampanye	151
Membuat kampanye standar	151
Membuat kampanye dengan AWS SDK for Java	151
Membuat kampanye uji A/B	154
Membuat kampanye pengujian A/B dengan AWS SDK for Java	154
Menggunakan SMS dan Voice API	157
Mengirim dan memvalidasi kata sandi satu kali	159
Mengirim pesan OTP	159
SendOtpMessage Respons	162
Memvalidasi pesan OTP	162
VerifyOtpMessage Respons	163
Contoh kode	164
Menghasilkan ID referensi	164
Mengirim kode OTP	164
Memvalidasi kode OTP	166
Mengirim dan mengambil pesan dalam aplikasi	168
Mengambil pesan dalam aplikasi untuk titik akhir	168
Memahami responsGetInAppMessages API	171
Objek InAppMessageCampaigns	172
Objek InAppMessage	174
Objek HeaderConfig	175
Objek BodyConfig	176
Objek InAppMessageContent	176
Objek Schedule	177
Objek InAppMessageButton	178
Objek DefaultButtonConfig	179
Objek OverrideButtonConfig	180
Memvalidasi nomor telepon	182

Kasus penggunaan validasi nomor telepon	182
Menggunakan layanan validasi nomor telepon	183
Tanggapan validasi nomor telepon	184
Mengirim pesan	188
Kirim email	188
Memilih metode pengiriman email	189
Memilih antara Amazon Pinpoint dan Amazon Simple Email Service (SES)	189
Menggunakan API	189
Mengirim email dengan header berhenti berlangganan	204
Kirim pesan SMS	206
Kirim pesan suara	219
Kirim notifikasi push	227
Membuat saluran khusus	237
Membuat kampanye yang mengirim pesan melalui saluran khusus	237
Memahami data peristiwa	238
Mengonfigurasi webhooks	240
Mengonfigurasi fungsi Lambda	240
Contoh fungsi Lambda	240
Format respons fungsi Lambda untuk Amazon Pinpoint	244
Memberikan izin Amazon Pinpoint untuk mengaktifkan fungsi Lambda	246
Acara streaming	249
Menyiapkan streaming acara	250
Prasyarat	250
AWS CLI	250
AWS SDK for Java	251
Menonaktifkan streaming acara	252
Acara aplikasi	252
Contoh	252
Atribut peristiwa	253
Acara kampanye	258
Contoh acara	258
Atribut acara kampanye	259
Acara perjalanan	266
Contoh acara	266
Atribut acara perjalanan	267
Peristiwa email	272

Contoh peristiwa	273
Atribut peristiwa email	279
Acara SMS	286
Contoh	286
Atribut acara SMS	287
Menanyakan data analitik	297
Metrik yang didukung	297
Dasar-dasar kueri	298
Kebijakan IAM	300
Metrik standar	303
Metrik aplikasi untuk kampanye	305
Metrik aplikasi untuk pesan email transaksional	310
Metrik aplikasi untuk pesan SMS transaksional	321
Metrik kampanye	328
Metrik keterlibatan perjalanan	339
Metrik eksekusi perjalanan	346
Metrik eksekusi aktivitas perjalanan	348
Metrik eksekusi perjalanan dan kampanye	351
Mengkueri data kampanye	354
Prasyarat	355
Mengkueri data untuk satu kampanye	356
Mengkueri data untuk beberapa kampanye	361
Mengkueri data pesan transaksional	367
Prasyarat	368
Mengkueri data untuk pesan email transaksional	369
Data kueri untuk pesan SMS transaksional	373
Menggunakan hasil kueri	379
Struktur JSON	380
Objek dan bidang JSON	385
Mencatat panggilan API	387
Amazon Pinpoint informasi di CloudTrail	387
Amazon Pinpoint Tindakan API yang dapat dicatat oleh CloudTrail	389
Amazon Pinpoint tindakan API email yang dapat dicatat CloudTrail	393
Amazon Pinpoint SMS dan voice API versi 1 tindakan yang dapat dicatat CloudTrail	394
Contoh: Entri file log Amazon Pinpoint	394
Penandaan sumber daya	400

Mengelola tag	400
Menggunakan tag dalam kebijakan IAM	401
Menambahkan tag ke sumber daya	402
Menambahkan tag dengan menggunakan API	402
Menambahkan tag dengan menggunakanAWS CLI	403
Penunjukan tag untuk sumber daya	404
Menampilkan tag dengan menggunakan API	405
Menampilkan tag dengan menggunakanAWS CLI	405
Memperbarui tag untuk sumber daya	406
Menghapus tag dari sumber daya	407
Menghapus tag dengan menggunakan API	407
Menghapus tag dengan menggunakanAWS CLI	408
Informasi terkait	408
Menyesuaikan rekomendasi denganAWS Lambda	409
Menggunakan rekomendasi dalam pesan	409
Buat fungsi Lambda	412
Data kejadian	412
Data dan persyaratan respons	414
Menetapkan kebijakan fungsi Lambda	419
Mengotorisasi Amazon Pinpoint untuk memanggil fungsi	421
Mengkonfigurasi Model Recommender	422
Menghapus data	423
Menghapus titik akhir	423
Menghapus Segmen dan Data Titik Akhir dari Amazon S3	424
Menghapus semua data proyek	424
Menghapus Semua Data AWS	425
Contoh kode	427
Amazon Pinpoint	428
Tindakan	428
API SMS dan Suara Amazon Pinpoint	513
Tindakan	514
Keamanan	523
Perlindungan data	524
Enkripsi data	526
Privasi lalu lintas jaringan internet	527
Membuat titik akhir VPC antarmuka untuk Amazon Pinpoint	528

Pengelolaan identitas dan akses	529
Audiens	530
Mengautentikasi dengan identitas	530
Mengelola akses menggunakan kebijakan	534
Bagaimana Amazon Pinpoint bekerja dengan IAM	536
Tindakan kebijakan Amazon Pinpoint	544
Contoh kebijakan berbasis identitas	575
Peran IAM untuk tugas-tugas umum	589
Pemecahan Masalah	607
Pencatatan dan pemantauan	609
Validasi kepatuhan	610
Ketangguhan	612
Keamanan infrastruktur	612
Konfigurasi dan analisis kerentanan	613
Praktik terbaik keamanan	613
Kuota	615
Kuota proyek	615
Kuota permintaan API	616
Kuota kampanye	618
Kuota email	621
Kuota pesan email	621
Kuota pengirim dan penerima email	621
Kuota pengiriman email	622
Kuota titik akhir	624
Kuota impor titik akhir	625
Kuota konsumsi acara	626
Kuota perjalanan	627
Kuota Lambda	628
Kuota pembelajaran mesin	629
Kuota templat pesan	631
Kuota pemberitahuan push	632
Kuota pesan dalam aplikasi	633
Kuota segmen	633
Kuota SMS	634
Kuota 10DLC	636
Kuota suara	637

Meminta peningkatan kuota	640
Riwayat dokumen	643
Pembaruan sebelumnya	652
.....	dclvii

Apa itu Amazon Pinpoint?

Amazon Pinpoint adalah AWS layanan yang dapat Anda gunakan untuk berinteraksi dengan pelanggan Anda di beberapa saluran pesan. Anda dapat menggunakan Amazon Pinpoint untuk mengirim pemberitahuan push, email, pesan teks SMS, atau pesan suara.

Informasi dalam panduan pengembang ini ditujukan untuk pengembang aplikasi. Panduan ini berisi informasi tentang penggunaan fitur Amazon Pinpoint. Ini juga berisi informasi yang menarik bagi pengembang aplikasi seluler, seperti prosedur untuk [mengintegrasikan fitur analitik dan perpesanan dengan aplikasi Anda](#).

Ada beberapa dokumen lain yang merupakan sahabat dokumen ini. Dokumen-dokumen berikut memberikan informasi referensi terkait dengan Amazon Pinpoint API:

- [Referensi API Amazon Pinpoint](#)
- [API SMS dan Suara Amazon Pinpoint](#)

Jika Anda baru mengenal Amazon Pinpoint, Anda mungkin merasa bermanfaat untuk meninjau [Panduan Pengguna Amazon Pinpoint](#) sebelum melanjutkan dokumen ini.

Fitur Amazon Pinpoint

Bagian ini menjelaskan fitur-fitur utama Amazon Pinpoint dan tugas yang dapat Anda lakukan dengan menggunakannya.

Menentukan segmen audiens

Jangkau audiens yang tepat untuk pesan Anda dengan [menentukan segmen audiens](#). Segmen menunjukkan pengguna mana yang menerima pesan yang dikirim dari kampanye. Anda dapat menentukan segmen dinamis berdasarkan data yang dilaporkan oleh aplikasi Anda, seperti sistem operasi atau jenis perangkat seluler. Anda juga dapat mengimpor segmen statis yang Anda tentukan dengan menggunakan layanan atau aplikasi lain.

Libatkan audiens Anda dengan kampanye pengiriman pesan

Libatkan audiens Anda dengan [membuat kampanye pesan](#). Kampanye mengirim pesan yang disesuaikan sesuai jadwal yang Anda tetapkan. Anda dapat membuat kampanye yang mengirim pesan push, email, atau SMS seluler.

Untuk bereksperimen dengan strategi kampanye alternatif, siapkan kampanye Anda sebagai pengujian A/B, dan analisis hasilnya dengan analisis Amazon Pinpoint.

Mengirim pesan SMS

Beri informasi pelanggan Anda dengan mengirimkan pesan push dan SMS seluler transaksional — seperti pesan aktivasi akun baru, konfirmasi pesanan, dan pemberitahuan pengaturan ulang kata sandi - langsung ke pengguna tertentu. Anda dapat mengirim pesan transaksional dengan menggunakan Amazon Pinpoint REST API.

Menganalisis perilaku pengguna

Dapatkan wawasan tentang audiens Anda dan efektivitas kampanye Anda dengan menggunakan analitik yang disediakan Amazon Pinpoint. Anda dapat melihat tren tentang tingkat keterlibatan pengguna, aktivitas pembelian, demografi, dan banyak lagi. Anda juga dapat memantau lalu lintas pesan Anda dengan melihat metrik seperti jumlah total pesan yang dikirim atau dibuka untuk kampanye atau aplikasi. Melalui Amazon Pinpoint API, aplikasi Anda dapat melaporkan data kustom, yang disediakan Amazon Pinpoint untuk analisis, dan Anda dapat melakukan kueri data analitik untuk metrik standar tertentu.

Untuk menganalisis atau menyimpan data analitik di luar Amazon Pinpoint, Anda dapat mengonfigurasi Amazon Pinpoint untuk [melakukan streaming data](#) ke Amazon Kinesis.

Ketersediaan wilayah

Amazon Pinpoint tersedia di beberapa Wilayah AWS di Amerika Utara, Eropa, Asia, dan Oseania. Di setiap Wilayah, AWS mempertahankan beberapa Availability Zone. Availability Zone ini secara fisik terisolasi satu sama lain, tetapi disatukan oleh koneksi jaringan privat, latensi rendah, throughput tinggi, dan sangat redundan. Availability Zone ini memungkinkan kami untuk menyediakan tingkat ketersediaan dan redundansi yang sangat tinggi, sekaligus meminimalkan latensi.

Untuk mempelajari lebih lanjut Wilayah AWS, lihat [Mengelola Wilayah AWS](#) di bagian Referensi Umum Amazon Web. Untuk daftar semua Wilayah tempat Amazon Pinpoint tersedia saat ini, lihat titik akhir [Amazon Pinpoint dan kuota serta titik akhir AWS layanan](#) di. Referensi Umum Amazon Web Untuk mempelajari selengkapnya tentang jumlah Availability Zone yang tersedia di setiap Wilayah, lihat [Infrastruktur AWS Global](#).

Tutorial

Tutorial di bagian ini dimaksudkan untuk menunjukkan pengguna Amazon Pinpoint baru bagaimana menyelesaikan beberapa tugas penting. Jika Anda baru mengenal Amazon Pinpoint, atau hanya tidak terbiasa dengan fitur tertentu, tutorial ini adalah tempat yang baik untuk memulai.

Tutorial dalam panduan ini mencakup tugas-tugas yang berorientasi pada pemirsa pengembang atau administrator sistem. Tutorial ini menunjukkan cara melakukan tugas dengan menggunakan API Amazon Pinpoint, AWSSDK, dan AWS CLI. Jika Anda terutama berinteraksi dengan Amazon Pinpoint dengan menggunakan konsol berbasis web, lihat bagian Tutorial dari Panduan Pengguna Amazon Pinpoint.

Tutorial

- [Tutorial: Menggunakan Postman dengan Amazon Pinpoint API](#)
- [Tutorial: Menyiapkan sistem pendaftaran SMS](#)

Tutorial: Menggunakan Postman dengan Amazon Pinpoint API

Postman adalah alat yang populer untuk menguji API di lingkungan easy-to-use grafis. Anda dapat menggunakan Postman untuk mengirim permintaan API ke REST API apa pun, dan untuk menerima tanggapan atas permintaan Anda. Menggunakan Postman adalah cara mudah untuk menguji dan memecahkan masalah panggilan yang Anda lakukan ke Amazon Pinpoint API. Tutorial ini mencakup prosedur untuk menyiapkan dan menggunakan Postman dengan Amazon Pinpoint.

Note

Postman dikembangkan oleh perusahaan pihak ketiga. Ini tidak dikembangkan atau didukung oleh Amazon Web Services (AWS). Untuk mempelajari lebih lanjut tentang menggunakan Postman, atau untuk bantuan terkait masalah yang terkait dengan Postman, lihat [Pusat Support](#) di situs web Postman.

Tentang tutorial ini tutorial tutorial tutorial tutorial tutorial tutorial

Bagian ini berisi ikhtisar tutorial ini.

Audiens yang Dimaksudkan

Tutorial ini ditujukan untuk pengembang dan pelaksana sistem. Anda tidak harus terbiasa dengan Amazon Pinpoint atau Postman untuk menyelesaikan langkah-langkah dalam tutorial ini. Anda harus nyaman mengelola kebijakan IAM dan memodifikasi contoh kode JSON.

Prosedur dalam tutorial ini dirancang untuk mencegah pengguna baru menggunakan operasi API yang dapat menghapus sumber daya Amazon Pinpoint secara permanen. Pengguna tingkat lanjut dapat menghapus pembatasan ini dengan mengubah kebijakan yang terkait dengan penggunaannya.

Fitur yang Digunakan

Tutorial ini mencakup contoh penggunaan untuk fitur Amazon Pinpoint berikut:

- Berinteraksi dengan Amazon Pinpoint API dengan menggunakan Postman

Waktu yang dibutuhkan

Diperlukan waktu sekitar 15 menit untuk menyelesaikan tutorial ini.

Pembatasan regional

Tidak ada batasan regional yang terkait dengan penggunaan solusi ini.

Biaya Penggunaan Sumber Daya

Tidak ada biaya untuk membuat akun AWS. Namun, dengan menerapkan solusi ini, Anda mungkin dikenakan biaya AWS penggunaan jika Anda menggunakan Postman untuk melakukan salah satu dari berikut ini:

- Mengirim email, SMS, push seluler, atau pesan suara
- Membuat dan mengirim kampanye
- Menggunakan fitur validasi nomor telepon

Untuk informasi selengkapnya tentang biaya yang terkait dengan penggunaan Amazon Pinpoint, lihat [harga Amazon Pinpoint](#).

Prasyarat

Sebelum menggunakan fungsi fungsi fungsi prasyarat berikut:

- Anda harus memiliki AWS akun. Untuk membuat AWS akun, buka <https://console.aws.amazon.com/> dan pilih Buat AWS akun baru.

- Pastikan akun yang Anda gunakan untuk masuk keAWS Management Console dapat membuat kebijakan dan peran IAM baru.
- Pastikan Anda memiliki setidaknya satu contoh proyek yang telah dihidupkan email dan identitas email terverifikasi. Lihat [Membuat proyek Amazon Pinpoint dengan dukungan email](#) di Panduan Pengguna Amazon Pinpoint.
- Pastikan Anda memiliki IDAWS Akun. IDAWS Akun dapat ditemukan di sudut kanan atas konsol atau menggunakan antarmuka baris perintah (CLI). Lihat [Menemukan IDAWS akun Anda](#).
- Anda harus mengunduh dan menginstal Postman di komputer Anda. Anda dapat mengunduh Postman dari situs [web Postman](#).
- Setelah Anda menginstal Postman di komputer Anda, buat akun Postman. Saat pertama kali memulai aplikasi Postman, Anda diminta untuk masuk atau membuat akun baru. Ikuti instruksi yang diberikan oleh Postman untuk masuk ke akun atau membuat akun jika belum memiliki akun.

Langkah 1: Buat kebijakan dan peran IAM

Saat Anda menggunakan Postman untuk menguji Amazon Pinpoint API, langkah pertama adalah membuat pengguna. Di bagian ini, Anda membuat kebijakan yang memungkinkan pengguna berinteraksi dengan semua sumber daya Amazon Pinpoint. Kemudian, Anda membuat pengguna dan melampirkan kebijakan langsung ke pengguna.

Buat kebijakan IAM

Pelajari cara membuat kebijakan IAM. Pengguna dan peran yang menggunakan kebijakan ini dapat berinteraksi dengan semua sumber daya di Amazon Pinpoint API. Ini juga menyediakan akses ke sumber daya yang terkait dengan Amazon Pinpoint Email API, serta Amazon Pinpoint SMS dan Voice API.

Untuk membuat kebijakan

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan dan kemudian pilih Buat kebijakan.
3. Di Editor kebijakan pilih JSON. Hapus JSON apa pun yang ada di editor Kebijakan sehingga kosong. Salin dan tempel JSON berikut ke editor Kebijakan dan kemudian di editor Kebijakan ganti semua instance **123456789012** dengan ID Anda. Akun AWS

Akun AWS ID Anda dapat ditemukan di sudut kanan atas konsol, atau Anda dapat menggunakan CLI, lihat [Menemukan ID AWS akun Anda](#).

Note

Untuk melindungi data di akun Amazon Pinpoint Anda, kebijakan ini hanya mencakup izin yang memungkinkan Anda membaca, membuat, dan memodifikasi sumber daya. Itu tidak termasuk izin yang memungkinkan Anda untuk menghapus sumber daya. Anda dapat mengubah kebijakan ini dengan menggunakan editor visual di konsol IAM. Untuk informasi selengkapnya, lihat [Mengelola kebijakan IAM](#) di Panduan Pengguna IAM. Anda juga dapat menggunakan [CreatePolicyVersion](#) operasi di IAM API untuk memperbarui kebijakan ini.

Selain itu, kebijakan ini mencakup izin yang memungkinkan Anda berinteraksi dengan sms-voice layanan ses dan layanan, selain mobiletargeting layanan. sms-voice izin ses dan memungkinkan Anda untuk berinteraksi dengan Amazon Pinpoint Email API dan Amazon Pinpoint SMS dan Voice API, masing-masing. mobiletargeting izin memungkinkan Anda untuk berinteraksi dengan Amazon Pinpoint API.

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



```

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

```

Pilih Selanjutnya.

4. Untuk nama Kebijakan, masukkan nama untuk kebijakan, seperti **PostmanAccessPolicy**. Pilih Buat kebijakan.
5. (Opsional) Anda dapat menambahkan tag ke kebijakan dengan memilih Tambah Tag.
6. Pilih Berikutnya: Tinjau.

Mmebuat pengguna IAM

Warning

Pengguna IAM memiliki kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

Setelah membuat kebijakan, Anda dapat membuat pengguna dan melampirkan kebijakan tersebut. Saat Anda membuat pengguna, IAM menyediakan serangkaian kredensial yang memungkinkan Postman menjalankan operasi Amazon Pinpoint API.

Untuk membuat pengguna

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di konsol IAM, di panel navigasi, pilih Pengguna, lalu pilih Buat pengguna.
3. Di bawah Rincian pengguna, untuk nama Pengguna, masukkan nama yang mengidentifikasi pengguna, seperti **PostmanUser**. Lalu pilih Selanjutnya.
4. Di bawah Setel izin, untuk opsi Izin, pilih Lampirkan kebijakan secara langsung.
5. Di bawah Kebijakan izin, pilih kebijakan (**PostmanAccessPolicy**) yang Anda buat di [Membuat kebijakan IAM](#). Lalu pilih Selanjutnya.
6. Pada halaman Tinjau dan buat, secara opsional tambahkan tag yang membantu Anda mengidentifikasi pengguna. Untuk informasi selengkapnya tentang penggunaan tag, lihat [Menandai sumber daya IAM](#) di Panduan Pengguna IAM.
7. Saat Anda siap membuat pengguna, pilih Buat pengguna.

Buat kunci akses

Warning

Skenario ini mengharuskan pengguna IAM dengan akses terprogram dan kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan. Kunci

akses dapat diperbarui jika perlu. Untuk informasi selengkapnya, lihat [Memperbarui kunci akses](#) di Panduan Pengguna IAM.

IAM menyediakan serangkaian kredensi yang dapat Anda gunakan untuk memungkinkan Postman menjalankan operasi Amazon Pinpoint API.

Untuk membuat pengguna

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di konsol IAM, di panel navigasi, pilih Pengguna. Pilih user (**PostmanUser**) yang dibuat di [Buat pengguna IAM](#), lalu pilih tab Security credentials.
3. Pada bagian Access key, pilih Buat access key.
4. Pada halaman Praktik & alternatif terbaik kunci Akses, pilih Aplikasi yang berjalan di luar AWS.

Lalu pilih Selanjutnya.

5. (Opsional) Anda dapat menambahkan tag deskripsi ke kebijakan.
6. Pilih Buat access key.
7. Pada halaman Ambil kunci akses, salin kredensial yang ditampilkan di kolom kunci Akses dan kunci akses Rahasia.

Note

Anda harus memberikan ID kunci akses dan kunci akses rahasia nanti dalam tutorial ini. Ini adalah satu-satunya saat Anda dapat melihat kunci akses rahasia. Kami menyarankan Anda menyalinnya dan menyimpannya di lokasi yang aman.

8. Setelah Anda menyimpan kedua kunci, pilih Selesai.

Langkah 2: Mengatur Tukang Pos

Sekarang setelah Anda membuat pengguna yang dapat mengakses Amazon Pinpoint API, Anda dapat mengatur Postman. Di bagian ini, Anda membuat satu atau lebih lingkungan di Postman. Selanjutnya, Anda mengimpor koleksi yang berisi templat permintaan untuk setiap operasi di Amazon Pinpoint API.

Buat ruang kerja Postman

Di Postman, ruang kerja adalah wadah organisasi untuk proyek dan lingkungan. Di bagian ini, Anda membuat setidaknya satu ruang kerja untuk digunakan dengan Amazon Pinpoint.

Buat ruang kerja

Di Postman, pilih tindakan lainnya pilih File, lalu pilih Baru.

1. Pada jendela Create New, pilih Workspace.
2. Masukkan nama, ringkasan, dan atur visibilitas ke pribadi. Kemudian pilih Create Workspace.

Buat lingkungan Postman

Dalam Postman, lingkungan adalah seperangkat variabel yang disimpan sebagai pasangan kunci-nilai. Anda dapat menggunakan lingkungan untuk mengubah konfigurasi permintaan yang Anda buat melalui Postman, tanpa harus mengubah permintaan API sendiri.

Di bagian ini, Anda membuat setidaknya satu lingkungan untuk digunakan dengan Amazon Pinpoint. Setiap lingkungan yang Anda buat berisi serangkaian variabel yang spesifik untuk akun Anda dalam satu Wilayah AWS. Jika Anda menggunakan prosedur di bagian ini untuk membuat lebih dari satu lingkungan, Anda dapat mengubah antar Wilayah dengan memilih lingkungan yang berbeda dari menu Lingkungan di Tukang Pos.

Untuk membuat lingkungan

1. Di Postman, pilih menu tindakan lainnya, pilih File, lalu pilih Baru.
2. Pada jendela Create New, pilih Environment.
3. Pada jendela KELOLA LINGKUNGAN, untuk Nama Lingkungan, masukkan **Amazon Pinpoint - *Region Name***. Ganti *Nama Wilayah* dengan salah satu nilai berikut:
 - AS Timur (Virginia Utara)
 - US West (Oregon)
 - Asia Pasifik (Mumbai)
 - Asia Pasifik (Sydney)
 - Europe (Frankfurt)
 - Eropa (Irlandia)

Note

Minimal, Anda hanya perlu membuat satu lingkungan untuk satu Wilayah AWS, dan itu Wilayah AWS harus berisi satu proyek. Jika Anda belum membuat proyek di salah satu yang terdaftar sebelumnya Wilayah AWS, lihat [Membuat proyek Amazon Pinpoint dengan dukungan email di Panduan Pengguna Amazon Pinpoint](#).

4. Buat enam variabel baru: `endpoint`, `region`, `serviceName`, `accountId`, `accessKey`, dan `secretAccessKey`. Gunakan tabel berikut untuk menentukan nilai mana yang akan dimasukkan dalam kolom Nilai Awal dan Nilai Saat Ini untuk setiap variabel.

Wilayah	Variabel	Nilai Awal dan Saat Ini
AS Timur (Virginia Utara)	<code>endpoint</code>	pinpoint.us-east-1 .amazonaws.com
	<code>region</code>	us-east-1
	<code>serviceName</code>	mobiletargeting
	<code>accountId</code>	(ID AWS akun Anda)
	<code>accessKey</code>	(ID kunci akses IAM Anda)
	<code>secretAccessKey</code>	(kunci akses rahasia IAM Anda)
AS Barat (Oregon)	<code>endpoint</code>	pinpoint.us-west-2 .amazonaws.com
	<code>region</code>	us-west-2
	<code>serviceName</code>	mobiletargeting
	<code>accountId</code>	(ID AWS akun Anda)

Wilayah	Variabel	Nilai Awal dan Saat Ini
	accessKey	(ID kunci akses IAM Anda)
	secretAccessKey	(kunci akses rahasia IAM Anda)
Asia Pasifik (Mumbai)	endpoint	pinpoint.ap-south-1.amazonaws.com
	region	ap-south-1
	serviceName	mobiletargeting
	accountId	(ID AWS akun Anda)
	accessKey	(ID kunci akses IAM Anda)
	secretAccessKey	(kunci akses rahasia IAM Anda)
Asia Pasifik (Sydney)	endpoint	pinpoint.ap-southeast-2.amazonaws.com
	region	ap-southeast-2
	serviceName	mobiletargeting
	accountId	(ID AWS akun Anda)
	accessKey	(ID kunci akses IAM Anda)
	secretAccessKey	(kunci akses rahasia IAM Anda)

Wilayah	Variabel	Nilai Awal dan Saat Ini
Eropa (Frankfurt)	endpoint	pinpoint.eu-central-1.amazonaws.com
	region	eu-central-1
	serviceName	mobiletargeting
	accountId	(ID AWS akun Anda)
	accessKey	(ID kunci akses IAM Anda)
	secretAccessKey	(kunci akses rahasia IAM Anda)
Eropa (Irlandia)	endpoint	pinpoint.eu-west-1.amazonaws.com
	region	eu-west-1
	serviceName	mobiletargeting
	accountId	(ID AWS akun Anda)
	accessKey	(ID kunci akses IAM Anda)
	secretAccessKey	(kunci akses rahasia IAM Anda)

Setelah Anda membuat variabel-variabel ini, jendela MANAGE ENVIRONMENTS menyerupai contoh yang ditunjukkan pada gambar berikut.

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

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

Setelah selesai, pilih Simpan.

⚠️ Important

Kunci akses yang ditunjukkan pada gambar sebelumnya adalah fiktif. Jangan bagikan kunci akses IAM Anda dengan orang lain.

Tukang pos menyertakan fitur yang memungkinkan Anda untuk berbagi dan mengekspor lingkungan. Jika Anda menggunakan fitur-fitur ini, pastikan untuk tidak membagikan ID kunci akses dan kunci akses rahasia Anda dengan siapa pun yang seharusnya tidak memiliki akses ke kredensial ini.

Untuk informasi selengkapnya, lihat [Praktik terbaik IAM](#) dalam Panduan Pengguna IAM.

- (Opsional) Ulangi langkah 1-4 untuk setiap lingkungan tambahan yang ingin Anda buat.

ℹ️ Tip

Di Postman, Anda dapat membuat lingkungan sebanyak yang Anda butuhkan. Anda dapat menggunakan lingkungan dengan cara berikut:

- Buat lingkungan terpisah untuk setiap Wilayah tempat Anda harus menguji Amazon Pinpoint API.
- Buat lingkungan yang terkait dengan yang berbeda Akun AWS.
- Buat lingkungan yang menggunakan kredensial yang terkait dengan pengguna lain.

- Setelah Anda selesai membuat lingkungan, lanjutkan ke bagian berikutnya.

Buat koleksi Amazon Pinpoint di Postman

Di Postman, koleksi adalah sekelompok permintaan API. Permintaan dalam koleksi biasanya disatukan oleh tujuan bersama. Di bagian ini, Anda membuat koleksi baru yang berisi templat permintaan untuk setiap operasi di Amazon Pinpoint API.

Untuk membuat koleksi Amazon Pinpoint

1. Di Postman, pilih menu tindakan lainnya, pilih File, lalu pilih Impor.
2. Pada jendela Impor, pilih Impor Dari Tautan, lalu masukkan URL berikut: https://raw.githubusercontent.com/awsdocs/amazon-pinpoint-developer-guide/master/Amazon%20pinpoint.postman_collection.json.

Pilih Impor. Postman mengimpor koleksi Amazon Pinpoint, yang berisi 120 contoh permintaan.

Uji konfigurasi Tukang Pos Anda

Setelah Anda mengimpor koleksi Amazon Pinpoint, kami sarankan Anda melakukan tes cepat untuk memverifikasi bahwa semua komponen telah dikonfigurasi dengan benar. Anda dapat menguji konfigurasi Anda dengan mengirimkan permintaan. GetApps Permintaan ini mengembalikan daftar semua proyek yang ada di akun Amazon Pinpoint Anda saat ini. Wilayah AWS Permintaan ini tidak memerlukan konfigurasi tambahan, jadi ini adalah cara yang baik untuk menguji konfigurasi Anda.

Untuk menguji konfigurasi koleksi Amazon Pinpoint

1. Di panel navigasi kiri, pilih Koleksi, perluas koleksi Amazon Pinpoint, lalu perluas folder Apps.
2. Dalam daftar permintaan, pilih GetApps.
3. Gunakan pemilih Lingkungan untuk memilih lingkungan yang Anda buat di lingkungan [Create Postman](#).
4. Pilih Kirim. Jika permintaan berhasil dikirim, panel respons menampilkan status. 200 OK Anda melihat respons yang menyerupai contoh pada gambar berikut.

```

1 {
2   "Item": [
3     {
4       "Name": "SampleProject1",
5       "Id": "12345678901234567890123456789012",
6       "Arn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/1234567890123456789012",
7       "tags": {}
8     },
9     {
10      "Name": "SampleProject2",
11      "Id": "98765432109876543210987654321098",
12      "Arn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/98765432109876543210987654321098",
13      "tags": {}
14    }
15  ]
16 }

```

Note

Jika Anda tidak memiliki proyek apa pun yang dibuat di Wilayah AWS Amazon Pinpoint kembali. { "Item": [] }

Tanggapan ini menampilkan daftar semua proyek Amazon Pinpoint yang ada di akun Anda di Wilayah yang Anda pilih pada langkah 3.

Pemecahan Masalah

Ketika Anda mengirimkan permintaan Anda, Anda mungkin melihat kesalahan. Lihat daftar berikut untuk beberapa kesalahan umum yang mungkin Anda temui, dan untuk langkah-langkah yang dapat Anda ambil untuk mengatasinya.

Pesan kesalahan	Masalah	Resolusi
Tidak bisa mendapatkan respons apa pun Ada kesalahan saat menghubungkan ke <code>https://%7B%7Bendpoint%7D%7D/v1/apps</code> .	Tidak ada nilai saat ini untuk <code>{{endpoint}}</code> variabel, yang diatur ketika Anda memilih lingkungan.	Gunakan pemilih lingkungan untuk memilih lingkungan.

Pesan kesalahan	Masalah	Resolusi
Token keamanan yang disertakan dalam permintaan tidak valid.	Tukang pos tidak dapat menemukan nilai saat ini dari ID kunci akses atau kunci akses rahasia Anda.	Pilih ikon roda gigi di dekat pemilih lingkungan, lalu pilih lingkungan saat ini. Verifikasi bahwa <code>secretAccessKey</code> nilai <code>accessKey</code> dan muncul di kolom NILAI AWAL dan NILAI SAAT INI, dan Anda memasukkan kredensialnya dengan benar.
<p>“Message”: “User: arn:aws:iam: :123456789012:user / tidak berwenang untuk melakukan: mobiletargeting: on resource: arn:aws:mobiletargeting:us-west-2:123456789012PinpointPostmanUser: *” GetApps</p>	Kebijakan IAM yang terkait dengan pengguna Anda tidak menyertakan izin yang sesuai.	<p>Verifikasi bahwa pengguna Anda memiliki izin yang dijelaskan dalam kebijakan Buat IAM, dan Anda memberikan kredensial yang benar saat membuat lingkungan di ruang kerja Create Postman.</p>

Langkah 3: Kirim permintaan tambahan

Setelah selesai mengonfigurasi dan menguji Postman, Anda dapat mulai mengirim permintaan tambahan ke Amazon Pinpoint API. Bagian ini mencakup informasi yang harus Anda ketahui sebelum mulai mengirim permintaan. Ini juga mencakup dua permintaan sampel yang menjelaskan cara menggunakan koleksi Amazon Pinpoint.

Important

Saat Anda menyelesaikan prosedur di bagian ini, Anda mengirimkan permintaan ke Amazon Pinpoint API. Permintaan ini membuat sumber daya baru di akun Amazon Pinpoint Anda, mengubah sumber daya yang ada, mengirim pesan, mengubah konfigurasi proyek Amazon Pinpoint Anda, dan menggunakan fitur Amazon Pinpoint lainnya. Berhati-hatilah saat Anda melaksanakan permintaan ini.

Tentang contoh-contoh dalam koleksi Amazon Pinpoint Postman

Anda harus mengonfigurasi sebagian besar operasi dalam koleksi Amazon Pinpoint Postman sebelum Anda dapat menggunakannya. Untuk GET dan DELETE operasi, Anda biasanya hanya perlu memodifikasi variabel yang diatur pada tab Pra-pemintaan Script.

Note

Bila Anda menggunakan kebijakan IAM yang ditampilkan di [Buat kebijakan IAM](#), Anda tidak dapat menjalankan DELETE permintaan apa pun yang disertakan dalam koleksi ini.

Misalnya, GetCampaign operasi mengharuskan Anda untuk menentukan a `projectId` dan `acampaignId`. Pada tab Pra-pemintaan Script, kedua variabel ini hadir, dan diisi dengan nilai contoh. Hapus nilai contoh dan ganti dengan nilai yang berlaku untuk proyek dan kampanye Amazon Pinpoint Anda.

Dari variabel-variabel ini, yang paling umum digunakan adalah `projectId` variabel. Nilai untuk variabel ini harus menjadi pengidentifikasi unik untuk proyek yang permintaan Anda terapkan. Untuk mendapatkan daftar pengidentifikasi ini untuk proyek Anda, lihat respons terhadap GetApps permintaan yang Anda kirim pada langkah sebelumnya dari tutorial ini. Dalam tanggapan itu, Id bidang menyediakan pengenalan unik untuk sebuah proyek. Untuk mempelajari lebih lanjut tentang GetApps operasi dan arti dari setiap bidang dalam respons, lihat [Aplikasi di Referensi](#) API Amazon Pinpoint.

Note

Di Amazon Pinpoint, “proyek” sama dengan “aplikasi” atau “aplikasi”.

Untuk POST dan PUT operasi, Anda juga harus memodifikasi badan permintaan untuk menyertakan nilai yang ingin Anda kirim ke API. Misalnya, ketika Anda mengirimkan CreateApp permintaan, yang merupakan POST permintaan, Anda harus menentukan nama untuk proyek yang Anda buat. Anda dapat memodifikasi permintaan pada tab Tubuh. Dalam contoh ini, ganti nilai di sebelahnya "Name" dengan nama proyek. Jika Anda ingin menambahkan tag ke proyek, Anda dapat menentukannya di tags objek. Atau, jika Anda tidak ingin menambahkan tag, Anda dapat menghapus seluruh tags objek.

Note

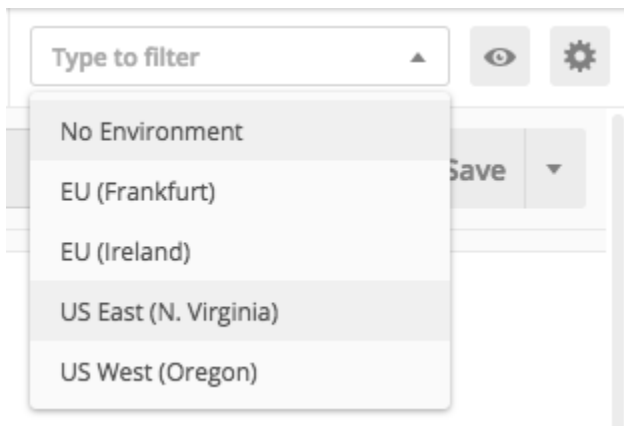
UntagResourceOperasi ini juga mengharuskan Anda untuk menentukan parameter URL. Anda dapat menentukan parameter ini pada tab Params. Ganti nilai di kolom VALUE dengan tag yang ingin Anda hapus untuk sumber daya yang ditentukan.

Contoh permintaan: Membuat proyek dengan menggunakan **CreateApp** operasi

Sebelum Anda membuat segmen dan kampanye di Amazon Pinpoint, Anda harus terlebih dahulu membuat proyek. Di Amazon Pinpoint, sebuah proyek terdiri dari segmen, kampanye, konfigurasi, dan data yang disatukan oleh tujuan bersama. Misalnya, Anda dapat menggunakan proyek untuk memuat semua konten yang terkait dengan aplikasi tertentu, atau dengan merek atau inisiatif pemasaran tertentu. Saat Anda menambahkan informasi pelanggan ke Amazon Pinpoint, informasi tersebut terkait dengan proyek.

Untuk membuat proyek dengan mengirimkan permintaan CreateApp API

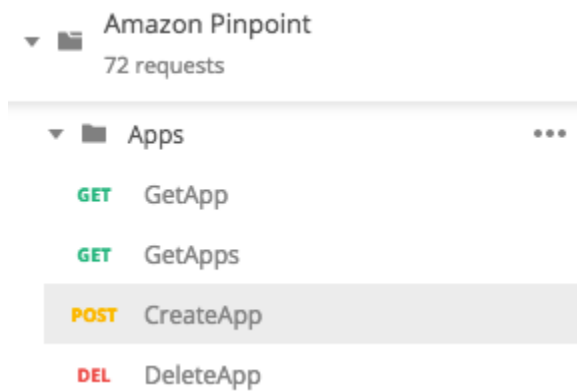
1. Pada menu Lingkungan, pilih tempat Wilayah AWS Anda ingin membuat proyek.



Dalam contoh ini, Postman telah dikonfigurasi sehingga menu Lingkungan menunjukkan empat opsi berikut:

- AS Timur (Virginia Utara)
- AS Barat (Oregon)
- Eropa (Frankfurt)
- Eropa (Irlandia)

2. Di folder Aplikasi, pilih CreateAppoperasi>.



Folder Apps dalam koleksi Amazon Pinpoint Postman diperluas dan menampilkan permintaan berikut:

- GetApp
 - GetApps
 - CreateApp
 - DeleteApp
3. Pada tab Body, di samping "Name", ganti nilai placeholder ("string") dengan nama untuk kampanye, seperti. **"MySampleProject"**
 4. Hapus koma setelah nama kampanye, lalu hapus seluruh tags objek pada baris 3 hingga 5. Ketika Anda selesai, permintaan Anda harus menyerupai contoh yang ditampilkan dalam cuplikan kode berikut.

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

Postman dikonfigurasi untuk mengirim permintaan sebagai payload JSON mentah.

5. Pilih Kirim. Jika kampanye berhasil dibuat, panel respons akan menampilkan status. 201 Created

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

```
}
```


Contoh: Mengirim email dengan menggunakan **SendMessages** operasi

Sangat umum untuk menggunakan Amazon Pinpoint SendMessages API untuk mengirim pesan transaksional. Salah satu keuntungan mengirim pesan dengan menggunakan SendMessages API sebagai lawan membuat kampanye, adalah Anda dapat mengirim pesan ke alamat apa pun, seperti alamat email, nomor telepon, atau token perangkat. Alamat yang Anda kirim pesan tidak harus ada di akun Amazon Pinpoint Anda. Mari kita bandingkan metode ini dengan mengirim pesan dengan membuat kampanye. Sebelum mengirim kampanye di Amazon Pinpoint, Anda harus menambahkan titik akhir ke akun Amazon Pinpoint, membuat segmen, membuat kampanye, dan menjalankan kampanye.

Contoh di bagian ini menunjukkan cara mengirim pesan email transaksional langsung ke alamat email tertentu. Anda dapat mengubah permintaan ini untuk mengirim pesan melalui saluran lain, seperti SMS, push seluler, atau suara.

Untuk mengirim pesan email dengan mengirimkan permintaan SendMessages

1. Verifikasi bahwa saluran email diaktifkan untuk proyek dan alamat email atau domain yang ingin Anda gunakan untuk mengirim dan menerima pesan dikonfigurasi. Untuk informasi selengkapnya, lihat [Mengaktifkan dan menonaktifkan saluran email](#) dan [Memverifikasi identitas email di Panduan Pengguna](#) Amazon Pinpoint.

 Note

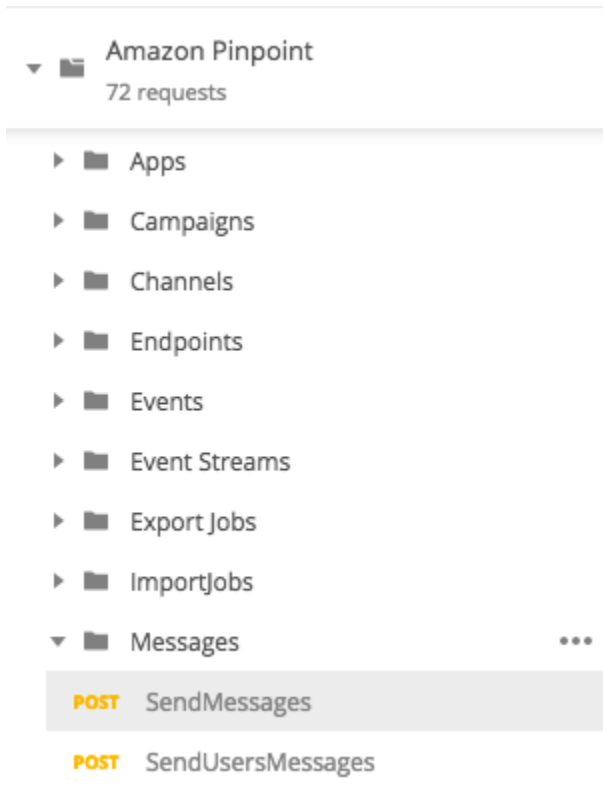
Untuk menyelesaikan prosedur di bagian ini, Anda harus memverifikasi dan alamat email

2. Pada menu Lingkungan, pilih Wilayah AWS yang ingin Anda kirim pesan.

Dalam contoh ini, Postman telah dikonfigurasi sehingga menu Lingkungan menunjukkan empat opsi berikut:

- AS Timur (Virginia Utara)
- AS Barat (Oregon)
- Eropa (Frankfurt)
- Eropa (Irlandia)

3. Di folder Pesan, pilih SendMessagesoperasi.



4. Pada tab Pra-permintaan Script, ganti nilai `projectId` variabel dengan ID proyek yang sudah ada di Wilayah yang Anda pilih di langkah 2 bagian ini.
5. Pada tab Body, hapus permintaan contoh yang ditampilkan di editor permintaan. Tempel kode berikut:

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



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

6. Pada kode sebelumnya, ganti *sender@example.com* dengan *alamat* email terverifikasi Anda. Ganti *recipient@example.com* dengan alamat email terverifikasi yang ingin Anda kirim pesan.

Note

Jika akun Anda masih dalam kotak pasir email Amazon Pinpoint, Anda hanya dapat mengirim email ke alamat atau domain yang diverifikasi di akun Amazon Pinpoint Anda. Untuk informasi selengkapnya tentang menghapus akun Anda dari kotak pasir, lihat [Meminta akses produksi untuk email di Panduan Pengguna Amazon Pinpoint](#).

7. Pilih Kirim. Jika pesan berhasil dikirim, panel respons menampilkan status. 200 OK

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

Tutorial: Menyiapkan sistem pendaftaran SMS

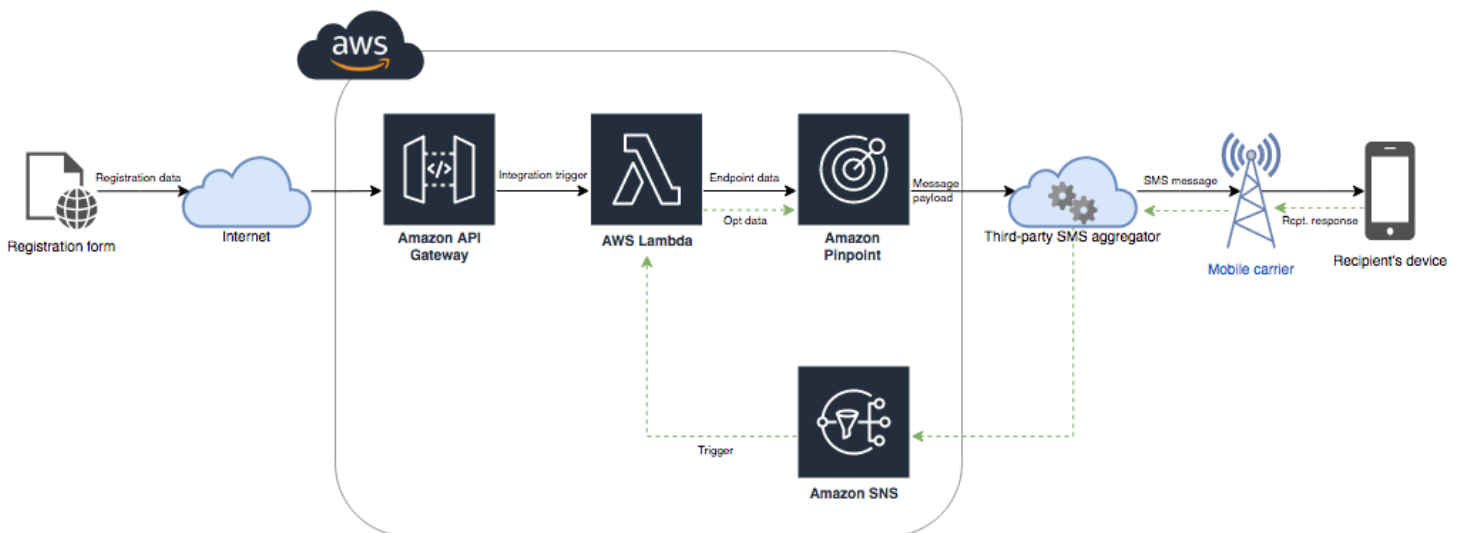
Pesan SMS (pesan teks) adalah cara yang bagus untuk mengirim pesan sensitif waktu kepada pelanggan Anda. Saat ini, banyak orang menyimpan ponsel mereka di dekatnya setiap saat. Selain

itu, pesan SMS cenderung menarik perhatian orang lebih dari pemberitahuan push, email, atau panggilan telepon.

Cara umum untuk menangkap nomor ponsel pelanggan adalah dengan menggunakan formulir berbasis web. Setelah Anda memverifikasi nomor telepon pelanggan dan mengonfirmasi langganannya mereka, Anda dapat mulai mengirim pesan SMS promosi, transaksional, dan informasi kepada pelanggan tersebut.

Tutorial ini menunjukkan cara mengatur formulir web untuk menangkap informasi kontak pelanggan. Formulir web mengirimkan informasi ini ke Amazon Pinpoint. Selanjutnya, Amazon Pinpoint memverifikasi bahwa nomor telepon valid, dan menangkap metadata lain yang terkait dengan nomor telepon. Setelah itu, Amazon Pinpoint mengirimkan pesan kepada pelanggan yang meminta mereka untuk mengonfirmasi langganannya mereka. Setelah pelanggan mengonfirmasi langganannya mereka, Amazon Pinpoint memilih mereka untuk menerima pesan Anda.

Diagram arsitektur berikut menunjukkan aliran data dalam solusi ini.



Tentang keikutsertaan ganda

Tutorial ini menunjukkan cara mengatur sistem opt-in ganda di Amazon Pinpoint yang menggunakan pesan SMS dua arah.

Dalam sistem keikutsertaan ganda SMS, pelanggan memberi Anda nomor telepon mereka dengan mengirimkannya dalam formulir web atau dalam aplikasi Anda. Saat menerima permintaan dari pelanggan, Anda membuat titik akhir baru di Amazon Pinpoint. Titik akhir yang baru harus dipilih keluar dari komunikasi Anda. Selanjutnya, Anda mengirim pesan ke nomor telepon itu. Dalam pesan Anda, Anda meminta penerima untuk mengonfirmasi langganannya mereka dengan membalas dengan

kata atau frasa tertentu (seperti “Ya” atau “Konfirmasi”). Jika pelanggan menanggapi pesan dengan kata atau frasa yang Anda tentukan, Anda mengubah status titik akhir menjadi opted-in. Jika tidak, jika pelanggan tidak merespons atau mereka merespons dengan kata atau frasa yang berbeda, Anda dapat meninggalkan titik akhir dengan status opted-out.

Tentang solusi ini

Bagian ini berisi informasi tentang solusi yang Anda sedang membangun dalam tutorial ini.

Audiens yang Dituju

Tutorial ini ditujukan untuk pengembang dan pemirsa pelaksana sistem. Anda tidak harus terbiasa dengan Amazon Pinpoint untuk menyelesaikan langkah-langkah dalam tutorial ini. Namun, Anda harus nyaman mengelola kebijakan IAM, membuat fungsi Lambda di Node.js, dan menerapkan konten web.

Fitur yang Digunakan

Tutorial ini mencakup contoh penggunaan untuk fitur Amazon Pinpoint berikut:

- Mengirim pesan SMS transaksional
- Memperoleh informasi tentang nomor telepon dengan menggunakan validasi nomor telepon
- Menerima pesan SMS masuk dengan menggunakan pesan SMS dua arah
- Membuat segmen dinamis
- Membuat kampanye
- Berinteraksi dengan Amazon Pinpoint API dengan menggunakan AWS Lambda

Waktu yang dibutuhkan

Diperlukan waktu sekitar satu jam untuk menyelesaikan tutorial ini. Setelah Anda menerapkan solusi ini, ada langkah-langkah tambahan yang dapat Anda ambil untuk menyempurnakan solusi agar sesuai dengan kasus penggunaan unik Anda.

Pembatasan Regional

Tutorial ini mengharuskan Anda untuk menyewa kode panjang dengan menggunakan konsol Amazon Pinpoint. Anda dapat menggunakan konsol Amazon Pinpoint untuk menyewa kode panjang

khusus yang berbasis di beberapa negara. Namun, hanya kode panjang yang berbasis di Kanada yang dapat digunakan untuk mengirim pesan SMS. (Anda dapat menggunakan kode panjang yang berbasis di negara dan wilayah lain untuk mengirim pesan suara.)

Kami mengembangkan contoh kode dalam tutorial ini dengan batasan ini dalam pikiran. Misalnya, contoh kode mengasumsikan bahwa nomor telepon penerima selalu memiliki 10 digit, dan kode negara 1. Jika Anda menerapkan solusi ini di negara atau wilayah selain Amerika Serikat atau Kanada, Anda harus memodifikasi contoh kode dengan tepat.

Biaya Penggunaan Sumber Daya

Tidak ada biaya untuk membuat AWS akun. Namun, dengan menerapkan solusi ini, Anda mungkin dikenakan biaya berikut:

- Biaya sewa kode panjang — Untuk menyelesaikan tutorial ini, Anda harus menyewa kode panjang. Kode panjang yang berbasis di Kanada berharga \$1,00 per bulan.
- Penggunaan validasi nomor telepon — Solusi dalam tutorial ini menggunakan fitur validasi nomor telepon Amazon Pinpoint untuk memverifikasi bahwa setiap nomor yang Anda terima valid dan diformat dengan benar, dan untuk mendapatkan informasi tambahan tentang nomor telepon. Anda membayar \$0,006 untuk setiap permintaan validasi nomor telepon.
- Biaya pengiriman pesan — Solusi dalam tutorial ini mengirimkan pesan SMS keluar. Anda membayar untuk setiap pesan yang Anda kirim melalui Amazon Pinpoint. Harga yang Anda bayar untuk setiap pesan tergantung pada negara atau wilayah penerima. Jika Anda mengirim pesan ke penerima di Amerika Serikat (tidak termasuk Wilayah AS), Anda membayar \$0,00645 per pesan. Jika Anda mengirim pesan ke penerima di Kanada, Anda membayar antara \$0,00109—\$0,02, tergantung pada operator dan lokasi penerima.
- Biaya penerimaan pesan — Solusi ini juga menerima dan memproses pesan SMS yang masuk. Anda membayar untuk setiap pesan masuk yang dikirim ke nomor telepon yang terkait dengan akun Amazon Pinpoint Anda. Harga yang Anda bayar tergantung di mana nomor telepon penerima berada. Jika nomor penerima Anda berbasis di Amerika Serikat (tidak termasuk Wilayah AS), Anda membayar \$0,0075 per pesan masuk. Jika nomor Anda berbasis di Kanada, Anda membayar \$0,00155 per pesan masuk.
- Penggunaan Lambda — Solusi ini menggunakan dua fungsi Lambda yang berinteraksi dengan Amazon Pinpoint API. Ketika Anda memanggil fungsi Lambda, Anda akan dikenakan biaya berdasarkan jumlah permintaan untuk fungsi Anda, untuk waktu yang diperlukan untuk mengeksekusi kode Anda, dan untuk jumlah memori yang digunakan fungsi Anda. Fungsi dalam tutorial ini menggunakan memori yang sangat sedikit, dan biasanya berjalan selama 1-3

detik. Beberapa atau semua penggunaan Anda atas solusi ini mungkin termasuk dalam tingkat penggunaan gratis Lambda. Untuk informasi lebih lanjut, lihat harga [Lambda](#).

- Penggunaan API Gateway — Formulir web dalam solusi ini memanggil API yang dikelola oleh API Gateway. Untuk setiap juta panggilan ke API Gateway, Anda membayar \$3,50—\$3,70, tergantung pada Wilayah mana Anda AWS menggunakan Amazon Pinpoint. Untuk informasi selengkapnya, lihat [harga API Gateway](#).
- Biaya hosting web — Solusi ini mencakup formulir berbasis web yang harus Anda host di situs web Anda. Harga yang Anda bayar untuk hosting konten ini tergantung pada penyedia hosting web Anda.

Note

Semua harga yang ditampilkan dalam daftar ini dalam Dolar AS (USD).

Berikutnya: [Prasyarat](#)

Prasyarat

Sebelum memulai tutorial ini, Anda harus menyelesaikan prasyarat berikut:

- Anda harus memiliki AWS akun. Membuat AWS akun, pergi ke <https://console.aws.amazon.com/> dan pilihlah Membuat AWS akun.
- Akun yang Anda gunakan untuk masuk ke akun AWS Management Console harus dapat melakukan tugas-tugas berikut:
 - Membuat kebijakan dan peran IAM baru
 - Membuat proyek Amazon Pinpoint
 - Buat fungsi Lambda baru
 - Membuat API baru di API Gateway
- Anda harus memiliki metode hosting halaman web, dan Anda harus tahu cara mempublikasikan halaman web. Meskipun Anda dapat menggunakan AWS layanan untuk meng-host halaman web Anda, Anda tidak diharuskan.

Tip

Untuk mempelajari selengkapnya tentang hosting halaman web menggunakan AWS layanan, lihat [Menyiapkan halaman web statis](#).

Selanjutnya: [Menyiapkan Amazon Pinpoint](#)

Langkah 1: Siapkan Amazon Pinpoint

Langkah pertama dalam menerapkan solusi ini adalah menyiapkan Amazon Pinpoint. Di bagian ini, Anda melakukan hal berikut:

- Buat proyek Amazon Pinpoint
- Aktifkan saluran SMS dan sewa nomor telepon
- Konfigurasi pesan SMS dua arah

Sebelum Anda mulai dengan tutorial ini, Anda harus meninjau [prasyarat](#).

Buat proyek Amazon Pinpoint

Untuk memulai, Anda perlu membuat proyek Amazon Pinpoint. Di Amazon Pinpoint, sebuah proyek terdiri dari segmen, kampanye, konfigurasi, dan data yang disatukan oleh tujuan bersama. Misalnya, Anda dapat menggunakan proyek untuk memuat semua konten yang terkait dengan aplikasi tertentu, atau dengan merek atau inisiatif pemasaran tertentu. Saat Anda menambahkan informasi pelanggan ke Amazon Pinpoint, informasi tersebut terkait dengan proyek.

Langkah-langkah yang terlibat dalam membuat proyek baru berbeda tergantung pada apakah Anda telah membuat proyek di Amazon Pinpoint sebelumnya.

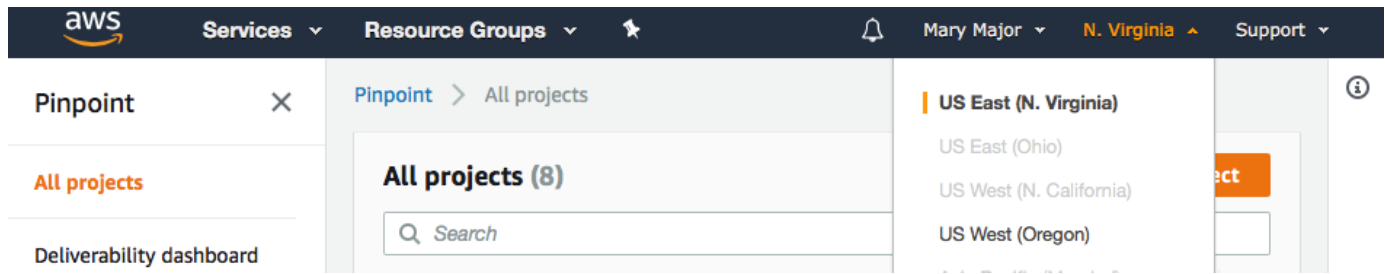
Membuat proyek (pengguna Amazon Pinpoint baru)

Langkah-langkah ini menjelaskan proses pembuatan proyek Amazon Pinpoint baru jika Anda belum pernah membuat proyek di Wilayah saat ini AWS .

Untuk membuat proyek

1. [Masuk ke AWS Management Console dan buka konsol Amazon Pinpoint di https://console.aws.amazon.com/pinpoint/.](https://console.aws.amazon.com/pinpoint/)

- Gunakan pemilih Wilayah untuk memilih AWS Wilayah yang ingin Anda gunakan, seperti yang ditunjukkan pada gambar berikut. Jika Anda tidak yakin, pilih Wilayah yang terletak paling dekat dengan Anda.



- Di bawah Memulai, untuk Nama, masukkan nama untuk kampanye (seperti **SMSRegistration**), lalu pilih Buat proyek.
- Pada halaman Konfigurasi fitur, pilih Lewati langkah ini.
- Di panel navigasi, pilih Semua proyek.
- Pada halaman Semua proyek, di samping proyek yang baru saja Anda buat, salin nilai yang ditampilkan di kolom Project ID.

Tip

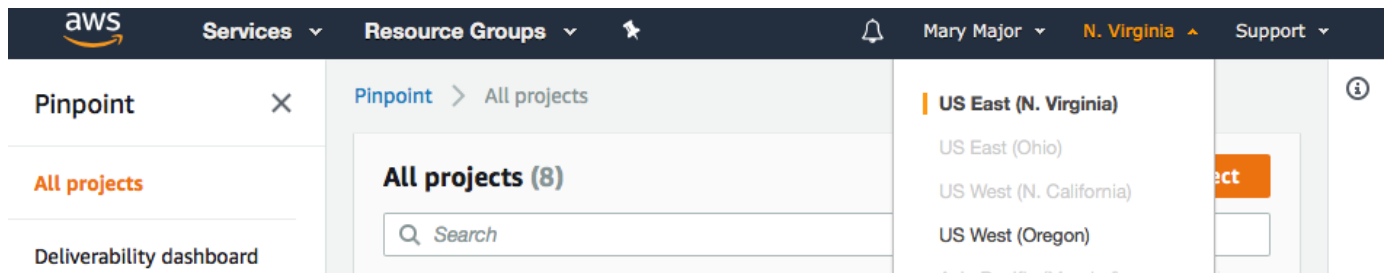
Anda perlu menggunakan ID ini di beberapa tempat berbeda dalam tutorial ini. Simpan ID proyek di tempat yang nyaman sehingga Anda dapat menyalinnya nanti.

Membuat proyek (pengguna Amazon Pinpoint yang ada)

Langkah-langkah ini menjelaskan proses pembuatan proyek Amazon Pinpoint baru jika Anda telah membuat proyek di Wilayah saat ini AWS .

Untuk membuat proyek

- [Masuk ke AWS Management Console dan buka konsol Amazon Pinpoint di https://console.aws.amazon.com/pinpoint/.](https://console.aws.amazon.com/pinpoint/)
- Gunakan pemilih Wilayah untuk memilih AWS Wilayah yang ingin Anda gunakan, seperti yang ditunjukkan pada gambar berikut. Jika Anda tidak yakin, pilih Wilayah yang terletak paling dekat dengan Anda.



3. Pada halaman Semua proyek, pilih Buat proyek.
4. Pada jendela Buat proyek, untuk nama Proyek, masukkan nama untuk proyek (seperti **SMSRegistration**). Pilih Buat.
5. Pada halaman Konfigurasi fitur, pilih Lewati langkah ini.
6. Di panel navigasi, pilih Semua proyek.
7. Pada halaman Semua proyek, di samping proyek yang baru saja Anda buat, salin nilai yang ditampilkan di kolom Project ID.

i Tip

Anda perlu menggunakan ID ini di beberapa tempat berbeda dalam tutorial ini. Simpan ID proyek di tempat yang nyaman sehingga Anda dapat menyalinnya nanti.

Dapatkan nomor telepon khusus

i Note

Amazon Pinpoint telah memperbarui dokumentasi panduan pengguna mereka. Untuk mendapatkan informasi terbaru mengenai cara membuat, mengonfigurasi, dan mengelola sumber daya SMS dan suara Amazon Pinpoint Anda, lihat panduan pengguna SMS [Amazon Pinpoint](#) yang baru.

Setelah Anda membuat proyek, Anda dapat mulai mengkonfigurasi fitur dalam proyek itu. Di bagian ini, Anda mengaktifkan saluran SMS, dan mendapatkan nomor telepon khusus untuk digunakan saat mengirim pesan SMS.

Note

Bagian ini mengasumsikan bahwa Anda menyewa nomor telepon 10DLC Amerika Serikat setelah pendaftaran merek dan kampanye, nomor Bebas Pulsa Amerika Serikat, atau kode panjang Kanada. Jika Anda mengikuti prosedur di bagian ini, tetapi memilih negara selain Amerika Serikat atau Kanada, Anda tidak akan dapat menggunakan nomor tersebut untuk mengirim pesan SMS. Untuk mempelajari selengkapnya tentang penyewaan kode panjang berkemampuan SMS di negara selain Amerika Serikat atau Kanada, lihat [Negara dan wilayah yang didukung \(saluran SMS\) di Panduan Pengguna SMS Amazon Pinpoint](#).

Untuk mengaktifkan saluran SMS menggunakan konsol Amazon Pinpoint, ikuti langkah-langkah berikut:

Aktifkan saluran SMS

1. [Masuk ke AWS Management Console dan buka konsol Amazon Pinpoint di https://console.aws.amazon.com/pinpoint/.](https://console.aws.amazon.com/pinpoint/)
2. Di panel navigasi, di bawah Settings (Pengaturan), pilih SMS and voice (SMS dan suara).
3. Di samping pengaturan SMS, pilih Edit.
4. Di bawah Pengaturan umum, pilih Aktifkan saluran SMS untuk proyek ini, lalu pilih Simpan perubahan.

Untuk meminta nomor telepon menggunakan konsol SMS Amazon Pinpoint, ikuti langkah-langkah berikut:

Minta nomor telepon (Konsol)

1. [Buka konsol SMS Amazon Pinpoint di https://console.aws.amazon.com/sms-voice/.](https://console.aws.amazon.com/sms-voice/)

Note

Pastikan Anda meminta nomor telepon Wilayah AWS yang sama dengan saat Anda membuat proyek Amazon Pinpoint.

2. Di panel navigasi, di bawah Konfigurasi, pilih Nomor telepon, lalu Minta pencetus.


3. Pada halaman Pilih negara untuk negara tujuan Pesan pilih Amerika Serikat atau Kanada. Pilih Selanjutnya.
4. Pada bagian Kasus penggunaan pesan, masukkan yang berikut ini:
 - Di bawah kemampuan Nomor pilih SMS

 Important

Kemampuan untuk SMS dan Voice tidak dapat diubah setelah nomor telepon telah dibeli.

- Untuk pesan dua arah pilih Ya.
5. Pilih Selanjutnya.
 6. Di bawah Pilih jenis pencetus pilih kode Panjang atau 10DLC.

Jika Anda memilih 10DLC dan sudah memiliki kampanye terdaftar, Anda dapat memilih kampanye dari Associate ke kampanye terdaftar.
 7. Pilih Selanjutnya.
 8. Pada Tinjauan dan permintaan Anda dapat memverifikasi dan mengedit permintaan Anda sebelum mengirimkannya. Pilih Minta.
 9. Jendela Registrasi yang Diperlukan dapat muncul tergantung pada jenis nomor telepon yang Anda minta. Nomor telepon Anda dikaitkan dengan pendaftaran ini dan Anda tidak dapat mengirim pesan sampai pendaftaran Anda disetujui. Untuk informasi selengkapnya tentang persyaratan pendaftaran, lihat [Pendaftaran](#).
 - a. Untuk nama formulir Pendaftaran masukkan nama yang ramah.
 - b. Pilih Mulai pendaftaran untuk menyelesaikan pendaftaran nomor telepon atau Daftar nanti.

 Important

Nomor telepon Anda tidak dapat mengirim pesan sampai pendaftaran Anda disetujui.

Anda masih ditagih biaya sewa bulanan berulang untuk nomor telepon terlepas dari status pendaftaran. Untuk informasi selengkapnya tentang persyaratan pendaftaran, lihat [Pendaftaran](#).

Aktifkan SMS dua arah

Sekarang setelah Anda memiliki nomor telepon khusus, Anda dapat mengatur SMS dua arah. Mengaktifkan SMS dua arah memungkinkan pelanggan Anda untuk menanggapi pesan SMS yang Anda kirimkan kepada mereka. Dalam solusi ini, Anda menggunakan SMS dua arah untuk memberi pelanggan Anda cara untuk mengonfirmasi bahwa mereka ingin berlangganan program SMS Anda.

Untuk mengaktifkan SMS dua arah menggunakan konsol SMS Amazon Pinpoint, ikuti langkah-langkah berikut:

Aktifkan SMS dua arah

1. [Buka konsol SMS Amazon Pinpoint di https://console.aws.amazon.com/sms-voice/](https://console.aws.amazon.com/sms-voice/).
2. Di panel navigasi, di bawah Konfigurasi, pilih Nomor telepon.
3. Pada halaman Nomor telepon pilih nomor telepon.
4. Pada tab SMS dua arah pilih tombol Edit pengaturan.
5. Pada halaman Edit pengaturan pilih Aktifkan pesan dua arah.
6. Untuk jenis Tujuan pilih Amazon SNS.
 - Topik Amazon SNS baru - Amazon Pinpoint SMS membuat topik di akun Anda. Topik dibuat secara otomatis dengan semua izin yang diperlukan. Untuk informasi selengkapnya tentang topik Amazon SNS, lihat [Mengonfigurasi Amazon SNS di panduan pengembang Amazon SNS](#).
 - Untuk tujuan pesan masuk masukkan nama topik, seperti **SMSRegistrationFormTopic**.
7. Untuk peran saluran dua arah pilih Gunakan kebijakan topik SNS.
8. Pilih Simpan perubahan.

Gunakan konsol SMS Amazon Pinpoint untuk menambahkan kata kunci ke nomor telepon yang dikirimkan pelanggan kepada Anda untuk mengonfirmasi langganan mereka (seperti **Yes** atau).

Confirm

Tambahkan kata kunci

1. [Buka konsol SMS Amazon Pinpoint di https://console.aws.amazon.com/sms-voice/](https://console.aws.amazon.com/sms-voice/).
2. Di panel navigasi, di bawah Konfigurasi, pilih Nomor telepon.
3. Pada halaman Nomor telepon pilih nomor telepon untuk menambahkan kata kunci.
4. Pada tab Kata Kunci pilih tombol Tambah kata kunci.

5. Di panel Kata Kunci Kustom tambahkan yang berikut ini:
 - Kata kunci — Kata kunci baru untuk ditambahkan (seperti **Yes** atau **Confirm**).
 - Pesan respons — Pesan untuk dikirim kembali ke penerima.
 - Tindakan kata kunci — Tindakan yang harus dilakukan ketika kata kunci diterima. Pilih Respons otomatis.
6. Pilih Tambah kata kunci.

Berikutnya: [Buat Kebijakan dan Peran IAM](#)

Langkah 2: Buat kebijakan dan peran IAM

Langkah selanjutnya dalam menerapkan solusi pendaftaran SMS adalah mengkonfigurasi kebijakan dan peran dalam AWS Identity and Access Management (IAM). Untuk solusi ini, Anda perlu membuat kebijakan yang menyediakan akses ke sumber daya tertentu yang terkait dengan Amazon Pinpoint. Anda kemudian membuat peran dan melampirkan kebijakan padanya. Kemudian dalam tutorial ini, Anda membuat AWS Lambda fungsi yang menggunakan peran ini untuk memanggil operasi tertentu di Amazon Pinpoint API.

Buat kebijakan IAM

Bagian ini menunjukkan cara membuat kebijakan IAM. Pengguna dan peran yang menggunakan kebijakan ini dapat melakukan hal berikut:

- Gunakan fitur Validasi Nomor Telepon
- Melihat, membuat, dan memperbarui titik akhir Amazon Pinpoint
- Kirim pesan ke titik akhir Amazon Pinpoint

Dalam tutorial ini, Anda ingin memberi Lambda kemampuan untuk melakukan tugas-tugas ini. Namun, untuk keamanan tambahan, kebijakan ini menggunakan prinsip pemberian hak istimewa paling sedikit. Dengan kata lain, itu hanya memberikan izin yang diperlukan untuk menyelesaikan solusi ini, dan tidak lebih. Kebijakan ini dibatasi dengan cara-cara berikut:

- Anda hanya dapat menggunakannya untuk memanggil API Validasi Nomor Telepon di Wilayah tertentu.
- Anda hanya dapat menggunakannya untuk melihat, membuat, atau memperbarui titik akhir yang terkait dengan proyek Amazon Pinpoint tertentu.

- Anda hanya dapat menggunakannya untuk mengirim pesan ke titik akhir yang terkait dengan proyek Amazon Pinpoint tertentu.

Untuk membuat kebijakan

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan dan kemudian pilih Buat kebijakan.
3. Pada tab JSON, tempel kode berikut.

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

```

        "Resource": "arn:aws:mobiletargeting:region:accountId:phone/number/
validate"
    }
]
}

```

Pada contoh sebelumnya, lakukan hal berikut:

- Ganti *wilayah* dengan AWS Wilayah tempat Anda menggunakan Amazon Pinpoint, seperti `us-east-1` atau `eu-central-1`

Tip

Untuk daftar lengkap AWS Wilayah tempat Amazon Pinpoint tersedia, lihat [AWS wilayah dan titik akhir](#) di. Referensi Umum AWS

- Ganti *accountId* dengan ID unik untuk akun Anda. AWS
- Ganti *ProjectID* dengan ID unik proyek yang Anda buat di Create [an Amazon Pinpoint project tutorial](#) ini.

Note

LogsTindakan memungkinkan Lambda untuk mencatat outputnya di CloudWatch Log.

4. Pilih Selanjutnya.
5. Untuk nama Kebijakan, masukkan nama untuk kebijakan, seperti **RegistrationFormPolicy**. Pilih Buat kebijakan.

Membuat peran IAM

Untuk membuat peran

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di konsol IAM, di panel navigasi, pilih Peran, lalu pilih Buat peran.
3. Di bawah Jenis entitas tepercaya, pilih AWS layanan, lalu untuk Layanan atau kasus pengguna pilih Lambda dari daftar drop-down.
4. Pilih Selanjutnya.

5. Di bawah Kebijakan izin, pilih atau cari kebijakan yang Anda buat di bagian sebelumnya, lalu pilih Berikutnya.
6. Di bawah Detail peran, untuk nama Peran, masukkan nama untuk peran tersebut, seperti **SMSRegistrationForm**. Pilih Buat peran.

Berikutnya: [Buat fungsi Lambda](#)

Langkah 3: Buat fungsi Lambda

Solusi ini menggunakan dua fungsi Lambda. Bagian ini menunjukkan cara membuat dan mengkonfigurasi fungsi-fungsi ini. Kemudian, Anda menyiapkan API Gateway dan Amazon Pinpoint untuk menjalankan fungsi ini ketika peristiwa tertentu terjadi. Kedua fungsi ini membuat dan memperbarui titik akhir dalam proyek Amazon Pinpoint yang Anda tentukan. Fungsi pertama juga menggunakan fitur validasi nomor telepon.

Buat fungsi yang memvalidasi informasi pelanggan dan membuat titik akhir

Fungsi pertama mengambil input dari formulir pendaftaran Anda, yang diterimanya dari Amazon API Gateway. Ini menggunakan informasi ini untuk mendapatkan informasi tentang nomor telepon pelanggan dengan menggunakan fitur [validasi nomor telepon](#) Amazon Pinpoint. Fungsi tersebut kemudian menggunakan data yang divalidasi untuk membuat titik akhir baru dalam proyek Amazon Pinpoint yang Anda tentukan. Secara default, titik akhir yang dibuat fungsi dipilih keluar dari komunikasi future dari Anda, tetapi status ini dapat diubah oleh fungsi kedua. Akhirnya, fungsi ini mengirimkan pesan kepada pelanggan yang meminta mereka untuk memverifikasi bahwa mereka ingin menerima komunikasi SMS dari Anda.

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda/>.
2. Pilih Buat fungsi.
3. Di bawah Buat fungsi, pilih Gunakan cetak biru.
4. Di bidang pencarian, masukkan **hello**, lalu tekan Enter. Dalam daftar hasil, pilih fungsi `hello-world Node.js`, seperti yang ditunjukkan pada gambar berikut.

Create function Info

Choose one of the following options to create your function.

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Basic information Info

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

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime
 nodejs18.x

Architecture
 x86_64

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Create a new role with basic Lambda permissions
 Use an existing role
 Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

🔄

[View the SMSRegistrationForm role](#) on the IAM console.

5. Di bagian Informasi dasar, lakukan hal berikut:

- Untuk Nama, masukkan nama untuk fungsi tersebut, seperti **RegistrationForm**.
- Untuk Peran, pilih Pilih peran yang sudah ada.
- Untuk peran yang ada, pilih RegistrationForm peran SMS yang Anda buat di [Buat peran IAM](#).

Setelah selesai, pilih Create function.

6. Untuk Sumber kode hapus fungsi sampel di editor kode, lalu tempel kode berikut:

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

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

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

// This message is spread across multiple lines for improved readability.
```



```
var message = "ExampleCorp: Reply YES to confirm your subscription. 2 msgs per "
  + "month. No purchase req'd. Msg&data rates may apply. Terms: "
  + "example.com/terms-sms";

var messageType = "TRANSACTIONAL";

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

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

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

async function createEndpoint(data, firstName, lastName, source) {
  var destinationNumber = data['NumberValidateResponse']
  ['CleansedPhoneNumberE164'];
```

```
var endpointId = data['NumberValidateResponse']
['CleansedPhoneNumberE164'].substring(1);

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

try{
  const UpdateEndpointresponse = await pinClient.send(new
  UpdateEndpointCommand(params));
```

```
        console.log(UpdateEndpointresponse);
        await sendConfirmation(destinationNumber);
    }catch(err){
        console.log(err);
    }
}

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

7. Pada tab Konfigurasi untuk variabel Lingkungan, pilih Edit dan kemudian Tambahkan variabel lingkungan, lakukan hal berikut:
 - Di baris pertama, buat variabel dengan kunci **originationNumber**. Selanjutnya, atur nilainya ke nomor telepon kode panjang khusus yang Anda terima di [Langkah 1.2](#).

Note

Pastikan untuk menyertakan tanda plus (+) dan kode negara untuk nomor telepon. Jangan sertakan karakter khusus lainnya, seperti tanda hubung (-), periode (.), atau tanda kurung.

- Di baris kedua, buat variabel dengan kunci **projectId**. Selanjutnya, tetapkan nilai ke ID unik proyek yang Anda buat di [Langkah 1.1](#).
- Di baris ketiga, buat variabel dengan kunci **region**. Selanjutnya, tetapkan nilainya ke Wilayah tempat Anda menggunakan Amazon Pinpoint, seperti **us-east-1** atau **us-west-2**

Ketika Anda selesai, bagian Variabel Lingkungan harus menyerupai contoh yang ditunjukkan pada gambar berikut.

Environment variables

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

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

► Encryption configuration

8. Di bagian atas halaman, pilih Simpan.

Uji fungsi

Setelah Anda membuat fungsi, Anda harus mengujinya untuk memastikan bahwa itu dikonfigurasi dengan benar. Juga, pastikan bahwa peran IAM yang Anda buat memiliki izin yang sesuai.

Untuk menguji fungsi

1. Pilih tab Uji.

2. Pilih Buat acara baru, lakukan hal berikut:

- Untuk Nama peristiwa, masukkan nama untuk peristiwa pengujian, seperti **MyPhoneNumber**.
- Hapus kode contoh di editor kode. Tempel kode berikut:

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

- Dalam contoh kode sebelumnya, ganti nilai `destinationNumber`, `firstName`, dan `lastName` atribut dengan nilai yang ingin Anda gunakan untuk pengujian, seperti detail kontak pribadi Anda. Saat Anda menguji fungsi ini, ia mengirimkan pesan SMS ke nomor telepon yang Anda tentukan dalam `destinationNumber` atribut. Pastikan nomor telepon yang Anda tentukan dapat menerima pesan SMS.
- Pilih Buat.

3. Pilih Uji.

4. Di bawah Hasil eksekusi: berhasil, pilih Detail. Di bagian Keluaran log, tinjau output dari fungsi tersebut. Pastikan fungsi berjalan tanpa kesalahan.

Periksa perangkat yang terkait dengan `destinationNumber` yang Anda tentukan untuk memastikan bahwa perangkat tersebut menerima pesan pengujian.

5. Buka konsol Amazon Pinpoint di <https://console.aws.amazon.com/pinpoint/>.
6. Pada halaman Semua proyek, pilih proyek yang Anda buat di [Buat proyek Amazon Pinpoint](#).
7. Di panel navigasi, pilih Segmen. Pada halaman Segmen, pilih Buat segmen.
8. Di Segmen grup 1, di bawah Tambahkan filter untuk menyempurnakan segmen Anda, pilih Filter menurut pengguna.
9. Untuk Pilih atribut pengguna, pilih `FirstName`. Kemudian, untuk Pilih nilai, pilih nama depan yang Anda tentukan dalam acara pengujian.

Bagian estimasi Segmen harus menunjukkan bahwa tidak ada titik akhir yang memenuhi syarat, dan satu titik akhir total, seperti yang ditunjukkan pada gambar berikut. Hasil ini diharapkan.

Saat fungsi membuat titik akhir baru, titik akhir dipilih keluar. Segmen di Amazon Pinpoint secara otomatis mengecualikan titik akhir yang dipilih keluar.

Segment group 1 Info

A segment group contains filters that you apply to base segments. If you choose an imported segment as a base segment, you can't use other imported segments as base segments nor add an additional segment group.

Include endpoints that are in **any** of the following segments **All segments**

Endpoints that match **any** of the following filters:

Filter 1: User

FirstName is Choose values
Carlos

Add more attributes or metrics to this filter Info
+ Add an attribute or metric

OR

Add filters to refine your segment.
Add a filter

Segment estimate Info

Eligible endpoints
The number of customers who will receive campaigns that target this segment.

0 endpoints

No matches found
Your segment didn't produce any results. Remove or modify your segment filters until the segment contains at least one member.

Total endpoints
The number of recipients who meet the criteria for this segment.

1 endpoints

Buat fungsi yang memilih pelanggan untuk komunikasi Anda

Fungsi kedua hanya dijalankan ketika pelanggan membalas pesan yang dikirim oleh fungsi pertama. Jika balasan pelanggan menyertakan kata kunci yang Anda tentukan dalam [Aktifkan SMS dua arah](#), fungsi akan memperbarui catatan titik akhir mereka untuk memilih mereka dalam komunikasi future. Amazon Pinpoint juga secara otomatis merespons dengan pesan yang Anda tentukan di [Aktifkan SMS dua arah](#).

Jika pelanggan tidak merespons, atau merespons dengan apa pun selain kata kunci yang ditunjuk, maka tidak ada yang terjadi. Titik akhir pelanggan tetap berada di Amazon Pinpoint, tetapi tidak dapat ditargetkan berdasarkan segmen.

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda/>.
2. Pilih Buat fungsi.
3. Di bawah fungsi Create, pilih Blueprints.
4. Di bidang pencarian, masukkan **hello**, lalu tekan Enter. Dalam daftar hasil, pilih fungsi `hello-world Node.js`, seperti yang ditunjukkan pada gambar berikut. Pilih Konfigurasi.
5. Di bagian Informasi dasar, lakukan hal berikut:
 - Untuk Nama, masukkan nama untuk fungsi tersebut, seperti **RegistrationForm_OptIn**.
 - Untuk Peran, pilih Pilih peran yang sudah ada.

- Untuk peran yang ada, pilih RegistrationForm peran SMS yang Anda buat di [Buat peran IAM](#).

Setelah selesai, pilih Create function.

6. Hapus fungsi sampel di editor kode, lalu tempel kode berikut:

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

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

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

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

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

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

// This function updates the opt-in status of a Pinpoint endpoint
async function updateEndpointOptIn(originationNumber, timestamp) {
```

```
// Extract the endpoint ID from the origination number
const endpointId = originationNumber.substring(1);

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

try {
  // Send the UpdateEndpointCommand to update the endpoint's opt-in status
  const updateEndpointResponse = await pinClient.send(new
UpdateEndpointCommand(params));
  console.log(updateEndpointResponse);
  console.log(`Successfully changed the opt status of endpoint ID
${endpointId}`);
} catch (error) {
  console.error('An error occurred while updating endpoint:', error);
  throw error; // Rethrow the error to handle it upstream
}
}
```

7. Di bawah variabel Lingkungan, lakukan hal berikut:

- Di baris pertama, buat variabel dengan kunci **projectId**. Selanjutnya, tetapkan nilai ke ID unik proyek yang Anda buat di [Buat proyek Amazon Pinpoint](#).
- Di baris kedua, buat variabel dengan kunci **region**. Selanjutnya, tetapkan nilainya ke Wilayah tempat Anda menggunakan Amazon Pinpoint, seperti **us-east-1** atau **us-west-2**.
- Di baris ketiga, buat variabel dengan kunci **confirmKeyword**. Selanjutnya, atur nilai ke kata kunci konfirmasi yang Anda buat di [Aktifkan SMS dua arah](#).

Note

Kata kunci tidak peka huruf besar/kecil. Fungsi ini mengubah pesan masuk ke huruf kecil.

Ketika Anda selesai, bagian Variabel Lingkungan harus menyerupai contoh yang ditunjukkan pada gambar berikut.

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

► Encryption configuration

8. Di bagian atas halaman, pilih Simpan.

Uji fungsi

Setelah Anda membuat fungsi, Anda harus mengujinya untuk memastikan bahwa itu dikonfigurasi dengan benar. Juga, pastikan bahwa peran IAM yang Anda buat memiliki izin yang sesuai.

Untuk menguji fungsi

1. Pilih Uji.
2. Pada jendela Configure test event, lakukan hal berikut:
 - a. Pilih Buat peristiwa pengujian baru.
 - b. Untuk Nama peristiwa, masukkan nama untuk peristiwa pengujian, seperti **MyResponse**.
 - c. Hapus kode contoh di editor kode. Tempel kode berikut:

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

Dalam contoh kode sebelumnya, ganti nilai `originationNumber` atribut dengan nomor telepon yang Anda gunakan saat menguji fungsi Lambda sebelumnya. Ganti nilai `messageBody` dengan kata kunci SMS dua arah yang Anda tentukan di [Aktifkan SMS dua arah](#). Secara opsional, Anda dapat mengganti nilai `Timestamp` dengan tanggal dan waktu saat ini.

- d. Pilih Buat.
3. Pilih UjiLagi.
4. Di bawah Hasil eksekusi: berhasil, pilih Detail. Di bagian Keluaran log, tinjau output dari fungsi tersebut. Pastikan fungsi berjalan tanpa kesalahan.
5. Buka konsol Amazon Pinpoint di <https://console.aws.amazon.com/pinpoint/>.
6. Pada halaman Semua proyek, pilih proyek yang Anda buat di [Buat proyek Amazon Pinpoint](#).
7. Di panel navigasi, pilih Segmen. Pada halaman Segmen, pilih Buat segmen.
8. Di Segmen grup 1, di bawah Tambahkan filter untuk menyempurnakan segmen Anda, pilih Filter menurut pengguna.
9. Untuk Pilih atribut pengguna, pilih `FirstName`. Kemudian, untuk Pilih nilai, pilih nama depan yang Anda tentukan dalam acara pengujian.

Bagian estimasi Segmen harus menunjukkan bahwa ada satu titik akhir yang memenuhi syarat, dan satu titik akhir total.

Berikutnya: [Mengatur Amazon API Gateway](#)

Langkah 4: Siapkan Amazon API Gateway

Di bagian ini, Anda membuat API baru dengan menggunakan Amazon API Gateway. Formulir pendaftaran yang Anda terapkan dalam solusi ini memanggil API ini. API Gateway kemudian meneruskan informasi yang diambil pada formulir pendaftaran ke fungsi Lambda yang Anda buat dalam fungsi Buat [Lambda](#).

Buat API

Pertama, Anda harus membuat API baru di API Gateway. Prosedur berikut menunjukkan cara membuat REST API baru.

Untuk membuat API baru

1. Buka konsol API Gateway di <https://console.aws.amazon.com/apigateway/>.
2. Pilih Buat API. Buat pilihan berikut:
 - Di bawah Pilih protokol, pilih REST.
 - Di bawah Create new API, pilih New API.
 - Di bawah Pengaturan, untuk Nama, masukkan nama, seperti **RegistrationForm**. Untuk Deskripsi, secara opsional masukkan beberapa teks yang menjelaskan tujuan API. Untuk Jenis Titik Akhir, pilih Regional. Kemudian, pilih Create API.

Contoh pengaturan ini ditunjukkan pada gambar berikut.

Choose the protocol

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

REST WebSocket

Create new API

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

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

Settings

Choose a friendly name and description for your API.

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

* Required


Create API

Buat sumber daya

Sekarang setelah Anda membuat API, Anda dapat mulai menambahkan sumber daya ke dalamnya. Setelah itu, Anda menambahkan metode POST ke sumber daya, dan memberi tahu API Gateway untuk meneruskan data yang Anda terima dari metode ini ke fungsi Lambda Anda.

1. Pada menu Tindakan, pilih Buat Sumber Daya. Di panel Sumber Daya Anak Baru, untuk Nama Sumber Daya, masukkan **register**, seperti yang ditunjukkan pada gambar berikut. Pilih Buat Sumber Daya.

New Child Resource

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

Configure as [proxy resource](#)

Resource Name*

Resource Path*

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

Enable API Gateway CORS

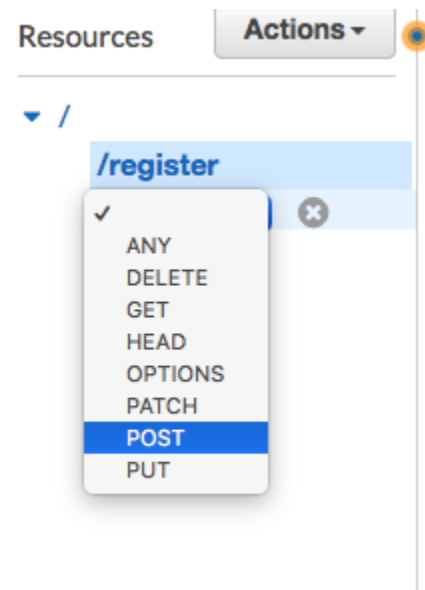
 

* Required

Cancel

Create Resource

2. Pada menu Actions, pilih Create Method. Dari menu yang muncul, pilih POST, seperti yang ditunjukkan pada gambar berikut. Kemudian pilih tombol tanda centang



3. Di panel /register - POST - Setup, buat pilihan berikut:

- Untuk jenis Integrasi, pilih Fungsi Lambda.

- Pilih Gunakan Integrasi Proxy Lambda.
- Untuk Wilayah Lambda, pilih Wilayah tempat Anda membuat fungsi Lambda.
- Untuk Fungsi Lambda, pilih fungsi yang Anda buat dalam RegisterEndpoint fungsi Buat [Lambda](#).

Contoh pengaturan ini ditunjukkan pada gambar berikut.

/register - POST - Setup



Choose the integration point for your new method.

Integration type Lambda Function ⓘ

HTTP ⓘ

Mock ⓘ

AWS Service ⓘ

VPC Link ⓘ

Use Lambda Proxy integration ⓘ

Lambda Region

Lambda Function

ⓘ

Use Default Timeout ⓘ

Save

Pilih Simpan. Pada jendela yang muncul, pilih OK untuk memberikan izin API Gateway untuk menjalankan fungsi Lambda Anda.

Terapkan API

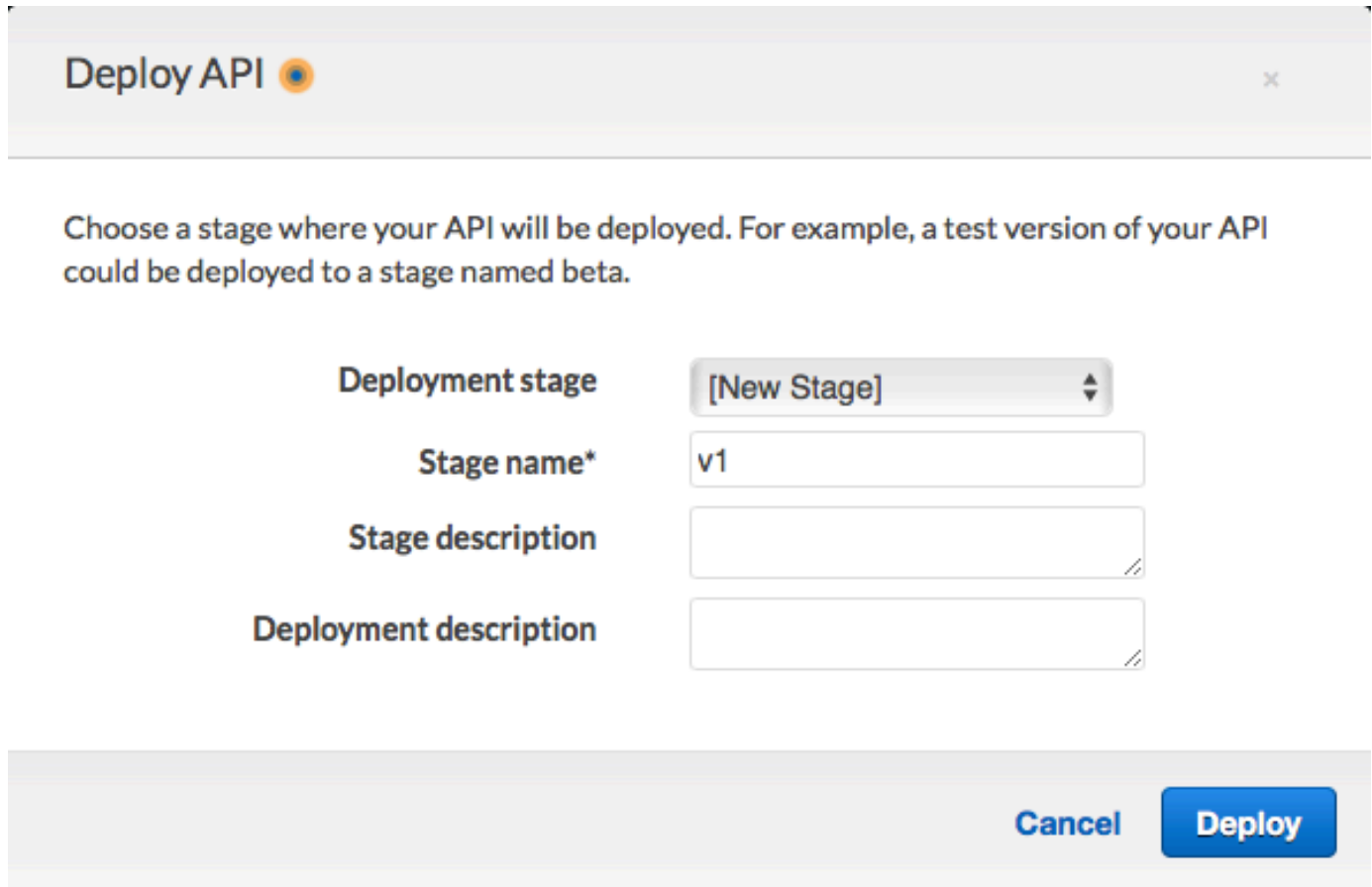
API sekarang siap digunakan. Pada titik ini, Anda harus menerapkannya untuk membuat titik akhir yang dapat diakses publik.

1. Pada menu Tindakan, pilih Deploy API. Pada jendela Deploy API, buat pilihan berikut:

- Untuk tahap Deployment, pilih [New Stage].

- Untuk nama Panggung, masukkan **v1**.
- Pilih Deploy.

Contoh pilihan ini ditunjukkan pada gambar berikut.



Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Deployment stage [New Stage]

Stage name* v1

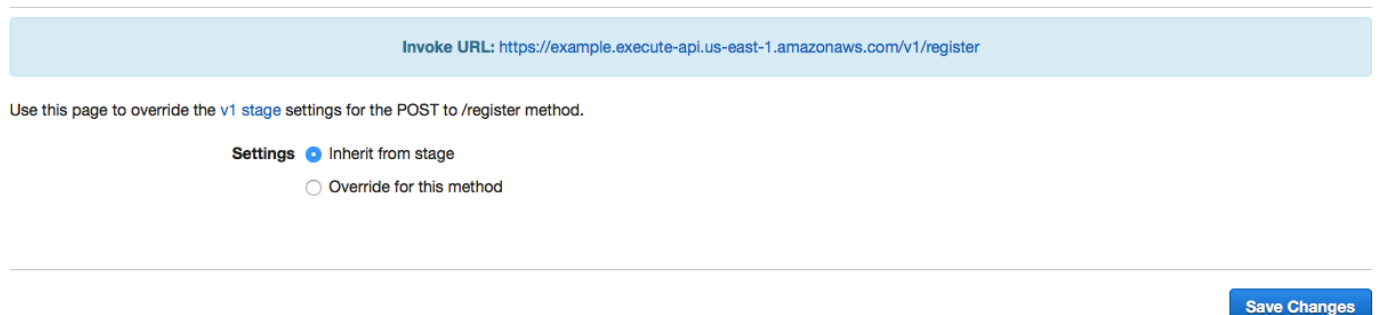
Stage description

Deployment description

Cancel Deploy

2. Di panel Editor Tahap v1, pilih sumber daya `/register`, lalu pilih metode POST. Salin alamat yang ditampilkan di sebelah URL Invoke, seperti yang ditunjukkan pada gambar berikut.

v1 - POST - `/register`



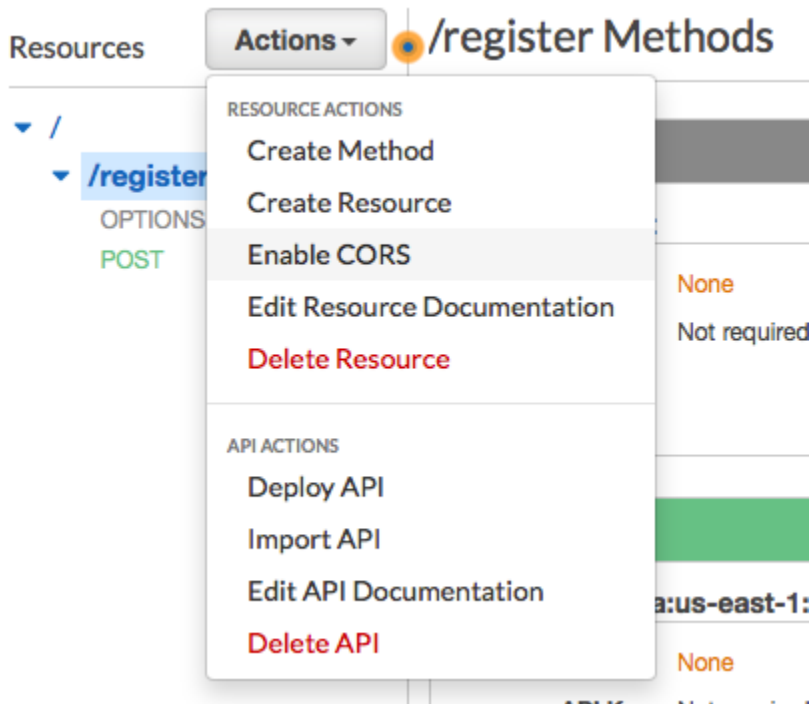
Invoke URL: `https://example.execute-api.us-east-1.amazonaws.com/v1/register`

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

Settings Inherit from stage Override for this method

Save Changes

3. Pada panel navigasi, silakan pilih Sumber Daya. Dalam daftar sumber daya, pilih sumber daya /register. Terakhir, pada menu Tindakan, pilih Aktifkan CORS, seperti yang ditunjukkan pada gambar berikut.



4. Pada panel Aktifkan CORS, pilih Aktifkan CORS dan ganti header CORS yang ada.

Berikutnya: [Membuat dan menyebarkan formulir web](#)

Langkah 5: Buat dan gunakan formulir web

Semua komponen solusi ini yang menggunakan AWS layanan sekarang ada. Langkah terakhir adalah membuat dan menyebarkan formulir web yang menangkap data pelanggan.

Buat JavaScript handler formulir

Di bagian ini, Anda membuat JavaScript fungsi yang mem-parsing konten formulir web yang Anda buat di bagian berikutnya. Setelah mengurai konten, fungsi ini mengirimkan data ke API yang Anda buat di [Mengatur Amazon API Gateway](#).

Untuk membuat handler formulir

1. Di editor teks, buat file baru.
2. Di editor, tempel kode berikut.


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

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

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

        e.preventDefault();

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

            timestamp.setUTCSeconds(utcSeconds);

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

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

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

3. Pada contoh sebelumnya, ganti *https://example.execute-api.us-east-1.amazonaws.com/v1/register* dengan URL Invoke yang Anda peroleh di [Deploy API](#).
4. Simpan file tersebut.

Buat file formulir

Di bagian ini, Anda membuat file HTML yang berisi formulir yang digunakan pelanggan untuk mendaftar program SMS Anda. File ini menggunakan penanganan JavaScript formulir yang Anda buat di bagian sebelumnya untuk mengirimkan data formulir ke fungsi Lambda Anda.

Important

Ketika pengguna mengirimkan formulir ini, itu memicu fungsi Lambda yang memanggil beberapa operasi Amazon Pinpoint API. Pengguna jahat dapat meluncurkan serangan pada formulir Anda yang dapat menyebabkan sejumlah besar permintaan dibuat. Jika Anda berencana untuk menggunakan solusi ini untuk kasus penggunaan produksi, Anda harus mengamankannya dengan menggunakan sistem seperti [Google reCAPTCHA](#).

Untuk membuat formulir

1. Di editor teks, buat file baru.
2. Di editor, tempel kode berikut.

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

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

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-gg0yR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-U02eT0CpHqDz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
```

```

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

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

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

```

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

3. Pada contoh sebelumnya, ganti *SMS FormHandler .js* dengan path lengkap ke JavaScript file handler formulir yang Anda buat di bagian sebelumnya.
4. Simpan file tersebut.

Unggah file formulir

Sekarang Anda telah membuat formulir HTML dan penanganan JavaScript formulir, langkah terakhir adalah mempublikasikan file-file ini ke internet. Bagian ini mengasumsikan bahwa Anda memiliki penyedia hosting web yang ada. Jika Anda tidak memiliki penyedia hosting yang ada, Anda dapat meluncurkan situs web dengan menggunakan Amazon Route 53, Amazon Simple Storage Service (Amazon S3), dan Amazon CloudFront Untuk informasi selengkapnya, lihat [Menghosting situs web statis](#).

Jika Anda menggunakan penyedia hosting web lain, lihat dokumentasi penyedia untuk informasi lebih lanjut tentang penerbitan halaman web.

Uji formulir

Setelah Anda mempublikasikan formulir, Anda harus mengirimkan beberapa acara pengujian untuk memastikan bahwa itu berfungsi seperti yang diharapkan.

Untuk menguji formulir pendaftaran

1. Di browser web, buka lokasi di mana Anda mengunggah formulir pendaftaran. Jika Anda menggunakan contoh kode dari [Buat penanganan JavaScript formulir](#), Anda melihat formulir yang menyerupai contoh pada gambar berikut.

Register for SMS Alerts

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

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

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

First name

Last name

Phone number

() -

Copyright © 2019, ExampleCorp or its affiliates.

2. Masukkan informasi kontak Anda di kolom Nama depan, Nama belakang, dan Nomor telepon.

Note

Saat Anda mengirimkan formulir, Amazon Pinpoint mencoba mengirim pesan ke nomor telepon yang Anda tentukan. Karena fungsi ini, Anda harus menggunakan nomor telepon asli untuk menguji solusi dari awal hingga akhir.

Jika Anda menguji fungsi Lambda dalam fungsi Buat [Lambda](#), proyek Amazon Pinpoint Anda sudah berisi setidaknya satu titik akhir. Saat menguji formulir ini, Anda harus mengirimkan nomor telepon yang berbeda pada formulir, atau menghapus titik akhir yang ada dengan menggunakan operasi [DeleteEndpointAPI](#).

- Periksa perangkat yang terkait dengan nomor telepon yang Anda tentukan untuk memastikan bahwa itu menerima pesan.
- Buka konsol Amazon Pinpoint di <https://console.aws.amazon.com/pinpoint/>.
- Pada halaman Semua proyek, pilih proyek yang Anda buat di [Buat proyek Amazon Pinpoint](#).
- Di panel navigasi, pilih Segmen. Pada halaman Segmen, pilih Buat segmen.
- Di Segmen grup 1, di bawah Tambahkan filter untuk menyempurnakan segmen Anda, pilih Filter menurut pengguna.
- Untuk Pilih atribut pengguna, pilih FirstName. Kemudian, untuk Pilih nilai, pilih nama depan yang Anda tentukan saat Anda mengirimkan formulir.

Bagian estimasi Segmen harus menunjukkan bahwa tidak ada titik akhir yang memenuhi syarat, dan satu titik akhir (di bawah Total titik akhir), seperti yang ditunjukkan pada contoh berikut. Hasil ini diharapkan. Saat fungsi Lambda membuat titik akhir baru, titik akhir dipilih keluar secara default.

The screenshot displays the Amazon Pinpoint console interface. On the left, the 'Segment group 1' configuration is shown. It includes a description of segment groups and options to include endpoints from 'any' of the following segments ('All segments'). Under 'Endpoints that match any of the following filters:', there is a filter named 'Filter 1: User' with the attribute 'FirstName' set to 'Carlos'. Below this, there is an 'OR' section for adding more filters. On the right, the 'Segment estimate' panel shows '0 endpoints' and a red warning box stating 'No matches found. Your segment didn't produce any results. Remove or modify your segment filters until the segment contains at least one member.' Below the warning, it shows 'Total endpoints' as '1 endpoints'.

- Pada perangkat yang menerima pesan, balas pesan dengan kata kunci SMS dua arah yang Anda tentukan di [Aktifkan SMS dua arah](#). Amazon Pinpoint segera mengirimkan pesan respons.

10. Di konsol Amazon Pinpoint, ulangi langkah 4 hingga 8. Kali ini, saat Anda membuat segmen, Anda akan melihat satu titik akhir yang memenuhi syarat, dan satu titik akhir total. Hasil ini diharapkan, karena titik akhir sekarang dipilih.

Langkah selanjutnya

Dengan menyelesaikan tutorial ini, Anda telah melakukan hal berikut ini:

- Membuat proyek Amazon Pinpoint, mengkonfigurasi saluran SMS, dan memperoleh kode panjang khusus.
- Membuat kebijakan IAM yang menggunakan prinsip hak istimewa paling sedikit untuk memberikan hak akses, dan mengaitkan kebijakan itu dengan peran.
- Membuat dua fungsi Lambda yang menggunakan operasi `PhoneNumberValidate`, `UpdateEndPoint`, dan `SendMessage` di Amazon Pinpoint API.
- Membuat API REST menggunakan API Gateway.
- Dibuat dan dikerahkan formulir berbasis web yang mengumpulkan informasi kontak pelanggan.
- Melakukan tes pada solusi untuk memastikannya bekerja.

Bagian ini membahas beberapa cara agar Anda dapat menggunakan informasi pelanggan yang Anda kumpulkan dengan menggunakan solusi ini. Ini juga mencakup beberapa saran cara agar Anda dapat menyesuaikan solusi ini agar sesuai dengan kasus penggunaan unik Anda.

Buat segmen pelanggan

Semua detail pelanggan yang Anda kumpulkan melalui formulir ini disimpan sebagai titik akhir. Solusi ini menciptakan titik akhir yang berisi beberapa atribut yang dapat Anda gunakan untuk tujuan segmentasi.

Misalnya, solusi ini menangkap atribut endpoint yang disebut `Source`. Atribut ini berisi path lengkap ke lokasi di mana formulir di-host. Ketika Anda membuat segmen, Anda dapat memfilter segmen berdasarkan endpoint, dan kemudian lebih menyempurnakan filter dengan memilih `SourceAtribut`.

Membuat segmen berdasarkan `Source` atribut dapat berguna dalam beberapa cara. Pertama, ini memungkinkan Anda untuk dengan cepat membuat segmen pelanggan yang mendaftar untuk menerima pesan SMS dari Anda. Selain itu, alat segmentasi di Amazon Pinpoint secara otomatis mengecualikan titik akhir yang tidak ikut serta untuk menerima pesan.

Parameter `Source` atribut ini juga berguna jika Anda memutuskan untuk menjadi tuan rumah formulir pendaftaran di beberapa lokasi yang berbeda. Misalnya, materi pemasaran Anda dapat merujuk ke formulir yang di-host di satu lokasi, sementara pelanggan yang menemukan formulir saat menjelajah situs web Anda dapat melihat versi yang di-host di tempat lain. Ketika Anda melakukan ini, atribut `Source` untuk pelanggan yang melengkapi formulir setelah melihat materi pemasaran Anda berbeda dari mereka yang melengkapi formulir setelah menemukannya di situs web Anda. Anda dapat menggunakan perbedaan ini untuk membuat segmen yang berbeda, dan kemudian mengirim komunikasi yang disesuaikan ke masing-masing pemirsa tersebut.

Kirim pesan kampanye yang dipersonalisasi

Setelah membuat segmen, Anda dapat mulai mengirim kampanye ke segmen tersebut. Saat membuat pesan kampanye, Anda dapat mempersonalisasikannya dengan menentukan atribut endpoint mana yang ingin Anda sertakan dalam pesan. Misalnya, formulir web yang digunakan dalam solusi ini mengharuskan pelanggan untuk memasukkan nama depan dan terakhir mereka. Nilai-nilai ini disimpan dalam catatan pengguna yang terkait dengan endpoint.

Misalnya, jika Anda menggunakan `GetEndpoint` Operasi API untuk mengambil informasi tentang endpoint yang dibuat menggunakan solusi ini, Anda melihat bagian yang menyerupai contoh berikut:

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

Jika ingin menyertakan nilai atribut ini dalam pesan kampanye, Anda dapat menggunakan notasi titik untuk merujuk ke atribut. Kemudian lampirkan seluruh referensi dalam kurung kurawal ganda. Misalnya, untuk menyertakan nama depan setiap penerima dalam pesan kampanye, sertakan string berikut dalam pesan: `{{User.UserAttributes.FirstName}}`. Saat Amazon Pinpoint mengirimkan pesan, pesan tersebut akan menggantikan string dengan nilai `FirstName` atribut.

Gunakan formulir untuk mengumpulkan informasi tambahan

Anda dapat memodifikasi solusi ini untuk mengumpulkan informasi tambahan pada formulir pendaftaran. Misalnya, Anda dapat meminta pelanggan untuk memberikan alamat mereka, dan kemudian menggunakan data alamat untuk mengisi `Location.City`, `Location.Country`, `Location.Region`, dan `Location.PostalCode` bidang di `Endpoint` sumber daya. Mengumpulkan informasi alamat pada formulir pendaftaran dapat mengakibatkan titik akhir yang berisi informasi yang lebih akurat. Untuk membuat perubahan ini, Anda perlu menambahkan bidang yang sesuai ke formulir web. Anda juga harus memodifikasi kode JavaScript untuk formulir untuk lulus nilai-nilai baru. Akhirnya, Anda harus memodifikasi fungsi Lambda yang menciptakan endpoint untuk menangani informasi baru yang masuk.

Anda juga dapat memodifikasi formulir sehingga mengumpulkan informasi kontak di saluran lain. Misalnya, Anda dapat menggunakan formulir untuk mengumpulkan alamat email pelanggan selain nomor telepon mereka. Untuk membuat perubahan ini, Anda perlu memodifikasi HTML dan JavaScript untuk formulir web. Anda juga harus memodifikasi fungsi Lambda yang menciptakan endpoint sehingga menciptakan dua endpoint terpisah (satu untuk endpoint email, dan satu untuk endpoint SMS). Anda juga harus memodifikasi fungsi Lambda sehingga menghasilkan nilai unik untuk `User.UserId` atribut, dan kemudian mengaitkan nilai itu dengan kedua titik akhir.

Rekam atribut tambahan untuk tujuan audit

Solusi ini mencatat dua atribut berharga saat membuat dan memperbarui titik akhir. Pertama, ketika fungsi Lambda pertama awalnya menciptakan endpoint, ia mencatat URL dari bentuk itu sendiri di `Attributes.Source` atribut. Jika pelanggan merespon pesan, fungsi Lambda kedua membuat `Attributes.OptInTimestamp` atribut. Atribut ini berisi tanggal dan waktu yang tepat ketika pelanggan memberikan persetujuan mereka untuk menerima pesan dari Anda.

Kedua bidang ini dapat berguna jika Anda pernah diminta oleh operator seluler atau badan pengatur untuk memberikan bukti persetujuan pelanggan. Anda dapat mengambil informasi ini kapan saja dengan menggunakan [GetEndPoint](#) Operasi API.

Anda juga dapat memodifikasi fungsi Lambda untuk merekam data tambahan yang mungkin berguna untuk tujuan audit, seperti alamat IP yang diajukan dari permintaan pendaftaran.

Mengintegrasikan Amazon Pinpoint dengan aplikasi Anda

Integrasikan Amazon Pinpoint dengan kode klien Anda untuk memahami dan melibatkan pengguna Anda.

Setelah Anda mengintegrasikan, saat pengguna meluncurkan aplikasi Anda, itu terhubung ke layanan Amazon Pinpoint untuk menambah atau memperbarui titik akhir. Titik akhir mewakili tujuan yang dapat Anda kirim pesan—seperti perangkat pengguna, alamat email, atau nomor telepon.

Selain itu, aplikasi Anda menyediakan data penggunaan, atau acara. Lihat data peristiwa di konsol Amazon Pinpoint untuk mempelajari berapa banyak pengguna yang Anda miliki, seberapa sering mereka menggunakan aplikasi Anda, kapan mereka menggunakannya, dan banyak lagi.

Setelah aplikasi Anda menyediakan titik akhir dan acara, Anda dapat menggunakan informasi ini untuk menyesuaikan kampanye pesan untuk audiens, atau segmen tertentu. (Anda juga dapat langsung mengirim pesan daftar penerima sederhana tanpa membuat kampanye.)

Gunakan topik di bagian ini untuk mengintegrasikan Amazon Pinpoint dengan aplikasi seluler atau web. Topik ini mencakup contoh kode dan prosedur untuk diintegrasikan dengan aplikasi JavaScript, Android, Swift, atau Flutter. Untuk mulai mengintegrasikan aplikasi Anda, lihat [the section called “Menghubungkan aplikasi front end Anda dengan AWS Amplify”](#).

Di luar klien, Anda dapat menggunakan [AWS SDK yang didukung](#) atau [Amazon Pinpoint API untuk mengimpor titik](#) akhir, mengeksport data peristiwa, menentukan segmen pelanggan, membuat dan menjalankan kampanye, dan banyak lagi.

Topik

- [Menggunakan Amazon Pinpoint dengan SDK AWS](#)
- [Menghubungkan aplikasi frontend Anda ke Amazon Pinpoint dengan Amplify AWS](#)
- [Mendaftarkan titik akhir dalam aplikasi Anda](#)
- [Melaporkan peristiwa dalam aplikasi Anda](#)
- [Menangani pemberitahuan push](#)

Menggunakan Amazon Pinpoint dengan SDK AWS

AWS kit pengembangan perangkat lunak (SDK) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK for C++	AWS SDK for C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK for Go	AWS SDK for Go contoh kode
AWS SDK for Java	AWS SDK for Java contoh kode
AWS SDK for JavaScript	AWS SDK for JavaScript contoh kode
AWS SDK for Kotlin	AWS SDK for Kotlin contoh kode
AWS SDK for .NET	AWS SDK for .NET contoh kode
AWS SDK for PHP	AWS SDK for PHP contoh kode
AWS Tools for PowerShell	Alat untuk contoh PowerShell kode
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) contoh kode
AWS SDK for Ruby	AWS SDK for Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Untuk contoh khusus untuk Amazon Pinpoint, lihat [Contoh kode untuk Amazon Pinpoint menggunakan AWS SDK](#)

Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Menghubungkan aplikasi frontend Anda ke Amazon Pinpoint dengan Amplify AWS

Gunakan AWS Amplify untuk mengintegrasikan aplikasi Anda. AWS Untuk aplikasi Swift, lihat [Memulai di dokumentasi](#) Amplify for Swift. Untuk aplikasi Android, lihat [Memulai dalam dokumentasi](#) Amplify for Android SDK. Untuk aplikasi React Native, lihat [Memulai](#) dalam JavaScript dokumentasi Amplify. Untuk aplikasi Flutter, lihat [Memulai](#) dalam dokumentasi Flutter SDK. Topik-topik ini membantu Anda:

- Siapkan sumber daya backend Anda.
- Hubungkan aplikasi Anda ke resource backend menggunakan library Amplify.

[Untuk mempelajari selengkapnya tentang menghubungkan aplikasi frontend Anda ke Amazon Pinpoint for Analytics, In-App Messaging, dan Notifikasi Push, lihat Memperkuat. AWS](#)

Langkah selanjutnya

Anda telah terintegrasi AWS Amplify dengan aplikasi Anda. Selanjutnya, perbarui kode Anda untuk mendaftarkan perangkat pengguna Anda sebagai titik akhir. Lihat [Mendaftarkan titik akhir dalam aplikasi Anda](#).

Mendaftarkan titik akhir dalam aplikasi Anda

Ketika pengguna memulai sesi (misalnya, dengan meluncurkan aplikasi seluler Anda), aplikasi seluler atau web Anda dapat secara otomatis mendaftarkan (atau memperbarui) titik akhir dengan Amazon Pinpoint. Titik akhir mewakili perangkat tempat pengguna memulai sesi. Ini mencakup atribut yang menggambarkan perangkat, dan juga dapat menyertakan atribut khusus yang Anda tentukan. Endpoint juga dapat mewakili metode lain untuk berkomunikasi dengan pelanggan, seperti alamat email atau nomor ponsel.

Setelah aplikasi Anda mendaftarkan titik akhir, Anda dapat mengelompokkan audiens berdasarkan atribut endpoint. Anda kemudian dapat melibatkan segmen ini dengan kampanye pesan yang disesuaikan. Anda juga dapat menggunakan halaman Analytics di konsol Amazon Pinpoint untuk melihat bagan tentang pendaftaran dan aktivitas titik akhir, seperti titik akhir baru dan titik akhir aktif harian.

Anda dapat menetapkan satu ID pengguna ke beberapa titik akhir. ID pengguna mewakili satu pengguna, sedangkan setiap titik akhir yang ditetapkan ID pengguna mewakili salah satu perangkat pengguna. Setelah menetapkan ID pengguna ke titik akhir, Anda dapat melihat bagan tentang aktivitas pengguna di konsol, seperti pengguna aktif harian dan pengguna aktif Bulanan.

Sebelum Anda memulai

Jika belum melakukannya, integrasikan AWS Mobile SDK untuk Android atau iOS, atau integrasikan JavaScript library AWS Amplify dengan aplikasi Anda. Lihat [Menghubungkan aplikasi frontend Anda ke Amazon Pinpoint dengan Amplify AWS](#).

Mendaftarkan endpoint dengan SDK AWS seluler untuk Android atau iOS

Anda dapat menggunakan SDK AWS Seluler untuk Android atau iOS untuk mendaftar dan menyesuaikan titik akhir. Untuk informasi lebih lanjut, dan untuk melihat contoh kode, lihat dokumen berikut:

- [Mendaftarkan titik akhir dalam aplikasi Anda](#) dalam dokumentasi Android SDK.
- [Mendaftarkan titik akhir dalam aplikasi Anda](#) dalam dokumentasi iOS SDK.

Mendaftarkan titik akhir dengan pustaka Amplify AWS JavaScript

Anda dapat menggunakan JavaScript library AWS Amplify untuk mendaftar dan memperbarui titik akhir di aplikasi Anda. Untuk informasi selengkapnya, dan untuk melihat contoh kode, lihat [Memperbarui titik akhir dalam dokumentasi](#) AWS AmplifyJavaScript.

Langkah selanjutnya

Anda telah memperbarui aplikasi Anda untuk mendaftarkan titik akhir. Sekarang, saat pengguna meluncurkan aplikasi Anda, informasi perangkat dan atribut khusus disediakan untuk Amazon Pinpoint. Anda dapat menggunakan informasi ini untuk menentukan segmen audiens. Di konsol, Anda dapat melihat metrik tentang titik akhir dan, jika berlaku, pengguna yang diberi ID pengguna.

Selanjutnya, selesaikan langkah-langkah [Melaporkan peristiwa dalam aplikasi Anda](#) untuk memperbarui aplikasi Anda untuk melaporkan data penggunaan.

Melaporkan peristiwa dalam aplikasi Anda

Dalam aplikasi seluler atau web Anda, Anda dapat menggunakan [AWS SDK Seluler](#) atau [API Amazon Pinpoint](#) untuk melaporkan data penggunaan, atau acara, ke Amazon Pinpoint. Anda dapat melaporkan peristiwa untuk menangkap informasi seperti waktu sesi, perilaku pembelian pengguna, upaya masuk, atau jenis acara kustom apa pun yang Anda butuhkan.

Setelah aplikasi melaporkan peristiwa, Anda dapat melihat analitik di konsol Amazon Pinpoint. Grafik di [Analitik Halaman](#) memberikan metrik untuk banyak aspek perilaku pengguna. Untuk informasi selengkapnya, lihat [Referensi bagan untuk analitik Amazon Pinpoint](#) di Panduan Pengguna Amazon Pinpoint.

Untuk menganalisis dan menyimpan data peristiwa Anda di luar Amazon Pinpoint, Anda dapat mengonfigurasi Amazon Pinpoint untuk melakukan streaming data ke Amazon Kinesis. Untuk informasi selengkapnya, lihat [Streaming acara Amazon Pinpoint ke Kinesis](#).

Dengan menggunakan [AWS SDK Seluler](#) dan [AWS Amplify](#) pustaka JavaScript, Anda dapat menghubungi Amazon Pinpoint API untuk melaporkan jenis peristiwa berikut:

Peristiwa sesi

Tunjukkan kapan dan seberapa sering pengguna membuka dan menutup aplikasi Anda.

Setelah aplikasi Anda melaporkan peristiwa sesi, gunakan [Analitik halaman](#) di konsol Amazon Pinpoint untuk Sesi, Titik akhir aktif, Tingkat retensi 7 hari, dan banyak lagi.

Peristiwa kustom

Apakah peristiwa tidak standar yang Anda tentukan dengan menetapkan jenis acara kustom. Anda dapat menambahkan atribut dan metrik khusus ke acara khusus.

Pada [Analitik halaman](#) di konsol, Peristiwa tab menampilkan metrik untuk semua acara kustom yang dilaporkan oleh aplikasi Anda.

Peristiwa monetisasi

Laporkan pendapatan yang dihasilkan oleh aplikasi Anda dan jumlah item yang dibeli oleh pengguna.

Pada **Analitik** halaman **Pendapatan** tab menampilkan grafik untuk **Pendapatan**, **Pengguna yang Membayar**, **Unit terjual**, dan banyak lagi.

Acara autentikasi

Tunjukkan seberapa sering pengguna mengautentikasi dengan aplikasi Anda.

Pada **Analitik** halaman **Pengguna** tab menampilkan grafik untuk **Masuk**, **Daftar**, dan **Kegagalan autenti**.

Sebelum Anda memulai

Jika Anda belum melakukannya, lakukan hal berikut:

- Mengintegrasikan aplikasi Anda dengan **AWS Amplify**. Lihat [Menghubungkan aplikasi frontend Anda ke Amazon Pinpoint dengan Amplify AWS](#).
- Perbarui aplikasi Anda untuk mendaftarkan titik akhir. Lihat [Mendaftarkan titik akhir dalam aplikasi Anda](#).

Melaporkan peristiwa dengan **AWSSDK** seluler untuk Android atau iOS

Anda dapat mengaktifkan aplikasi seluler untuk melaporkan peristiwa ke Amazon Pinpoint dengan menggunakan **AWSSDK** seluler untuk iOS dan Android.

Untuk informasi selengkapnya tentang memperbarui aplikasi Anda untuk merekam dan mengirimkan peristiwa ke Amazon Pinpoint, lihat halaman berikut di **AWS Dokumentasi Amplify**:

- [Analitik](#) dalam dokumentasi SDK iOS
- [Analitik](#) di dokumentasi Android SDK

Melaporkan peristiwa dengan **AWS Amplify JavaScript**

Anda dapat mengaktifkan aplikasi JavaScript dan React Native untuk melaporkan peristiwa penggunaan aplikasi ke Amazon Pinpoint dengan menggunakan **AWS Amplify JavaScript**. Untuk informasi lebih lanjut tentang memperbarui aplikasi Anda untuk mengirimkan peristiwa ke Amazon Pinpoint, lihat [Analitik](#) di **AWS Amplify dokumentasi JavaScript**.

Melaporkan peristiwa dengan menggunakan API Amazon Pinpoint

Anda dapat menggunakan API Amazon Pinpoint atau AWS SDK untuk mengirimkan acara ke Amazon Pinpoint secara massal. Untuk informasi selengkapnya, lihat [Peristiwa](#) di Referensi API Amazon Pinpoint.

Langkah selanjutnya

Anda telah memperbarui aplikasi Anda untuk melaporkan peristiwa. Sekarang ketika pengguna berinteraksi dengan aplikasi Anda, aplikasi akan mengirimkan data penggunaan ke Amazon Pinpoint. Anda dapat melihat data ini di konsol, dan Anda dapat mengalirkannya ke Amazon Kinesis.

Selanjutnya, perbarui aplikasi Anda untuk menangani notifikasi push yang Anda kirim dengan Amazon Pinpoint. Lihat [Menangani pemberitahuan push](#).

Menangani pemberitahuan push

Topik berikut menjelaskan cara memodifikasi aplikasi Swift, Android, React Native, atau Flutter Anda sehingga menerima notifikasi push yang Anda kirim dengan menggunakan Amazon Pinpoint.

Topik

- [Menyiapkan notifikasi push untuk Amazon Pinpoint](#)
- [Menangani notifikasi push](#)

Menyiapkan notifikasi push untuk Amazon Pinpoint

Untuk menyiapkan Amazon Pinpoint agar dapat mengirim notifikasi push ke aplikasi Anda, pertama-tama Anda harus memberikan kredensi yang mengizinkan Amazon Pinpoint untuk mengirim pesan ke aplikasi Anda. Kredensi yang Anda berikan bergantung pada sistem notifikasi push mana yang Anda gunakan:

- Untuk aplikasi iOS, Anda memberikan sertifikat SSL, yang diperoleh dari portal Pengembang Apple. Sertifikat mengizinkan Amazon Pinpoint untuk mengirim pesan ke aplikasi Anda melalui layanan Pemberitahuan Push Apple.
- Untuk aplikasi Android, Anda menyediakan Kunci API Web yang diperoleh dari Firebase console. Kredensi ini mengotorisasi Amazon Pinpoint untuk mengirim pesan ke aplikasi Anda melalui Firebase Cloud Messaging.

Setelah mendapatkan kredensi untuk saluran notifikasi push, Anda harus membuat proyek di Amazon Pinpoint dan memberikannya kredensi untuk layanan notifikasi push.

Topik

- [Menyiapkan Pemberitahuan Push Swift](#)
- [Menyiapkan notifikasi push Android](#)
- [Menyiapkan Pemberitahuan Push Flutter](#)
- [Menyiapkan React Native Push Notifications](#)
- [Buat proyek di Amazon Pinpoint](#)

Menyiapkan Pemberitahuan Push Swift

Notifikasi push untuk aplikasi iOS dikirim menggunakan layanan Apple Push Notification (APN). Sebelum dapat mengirim notifikasi push ke perangkat iOS, Anda harus membuat ID aplikasi di portal Pengembang Apple, dan Anda harus membuat sertifikat yang diperlukan. Anda dapat menemukan informasi selengkapnya tentang menyelesaikan langkah-langkah ini dalam [Menyiapkan layanan notifikasi push](#) di dokumentasi AWS Amplify.

Bekerja dengan token APN

Sebagai praktik terbaik, Anda harus mengembangkan aplikasi agar token perangkat pelanggan diregenerasi saat aplikasi diinstal ulang.

Jika penerima meningkatkan perangkat mereka ke iOS versi mayor baru (misalnya, dari iOS 12 ke iOS 13), dan kemudian menginstal ulang aplikasi Anda, aplikasi akan menghasilkan token baru. Jika aplikasi Anda tidak me-refresh token, token lama digunakan untuk mengirim notifikasi. Akibatnya, Apple Push Notification service (APN) menolak notifikasi, karena token sekarang tidak valid. Ketika Anda mencoba untuk mengirim pemberitahuan, Anda menerima pemberitahuan kegagalan pesan dari APN.

Menyiapkan notifikasi push Android

Notifikasi push untuk aplikasi Android dikirim menggunakan Firebase Cloud Messaging (FCM), yang menggantikan Google Cloud Messaging (GCM). Sebelum dapat mengirim notifikasi push ke perangkat Android, Anda harus mendapatkan kredensi FCM. Anda kemudian dapat menggunakan kredensi tersebut untuk membuat project Android dan meluncurkan contoh aplikasi yang dapat menerima notifikasi push. Anda dapat menemukan informasi selengkapnya tentang menyelesaikan langkah-langkah ini di bagian [Pemberitahuan push](#) di dokumentasi AWS Amplify.

Menyiapkan Pemberitahuan Push Flutter

Notifikasi push untuk aplikasi Flutter dikirim menggunakan Firebase Cloud Messaging (FCM) untuk Android, dan APN untuk iOS. Anda dapat menemukan informasi lebih lanjut tentang menyelesaikan langkah-langkah ini di bagian Pemberitahuan push pada dokumentasi [AWS Amplify Flutter](#).

Menyiapkan React Native Push Notifications

Notifikasi push untuk aplikasi React Native dikirim menggunakan Firebase Cloud Messaging (FCM) untuk Android, dan APN untuk iOS. Anda dapat menemukan informasi selengkapnya tentang menyelesaikan langkah-langkah ini di bagian Pemberitahuan push pada JavaScript dokumentasi [AWS Amplify](#).

Buat proyek di Amazon Pinpoint

Di Amazon Pinpoint, sebuah proyek adalah kumpulan pengaturan, data, kampanye, dan segmen yang semuanya berbagi tujuan bersama. Di Amazon Pinpoint API, proyek juga disebut sebagai aplikasi. Bagian ini menggunakan kata “proyek” secara eksklusif ketika mengacu pada konsep ini.

Untuk mulai mengirim pemberitahuan push di Amazon Pinpoint, Anda harus membuat proyek. Selanjutnya, Anda harus mengaktifkan saluran notifikasi push yang ingin Anda gunakan dengan memberikan kredensi yang sesuai.

Anda dapat membuat proyek baru dan menyiapkan saluran notifikasi push dengan menggunakan konsol Amazon Pinpoint. Untuk informasi selengkapnya, lihat [Menyiapkan saluran pemberitahuan push Amazon Pinpoint](#) di Panduan Pengguna Amazon Pinpoint.

Anda juga dapat membuat dan mengatur proyek dengan menggunakan [API Amazon Pinpoint](#), sebuah [AWS SDK](#), atau [AWS Command Line Interface](#) (AWS CLI). Untuk membuat proyek, gunakan `AppSource`. Untuk mengkonfigurasi saluran notifikasi push, gunakan sumber daya berikut:

- [Saluran APN](#) untuk mengirim pesan ke pengguna perangkat iOS dengan menggunakan layanan Pemberitahuan Push Apple.
- [Saluran ADM](#) untuk mengirim pesan ke pengguna perangkat Amazon Kindle Fire.
- [Saluran Baidu](#) untuk mengirim pesan ke pengguna Baidu.
- [Saluran GCM](#) untuk mengirim pesan ke perangkat Android menggunakan Firebase Cloud Messaging (FCM), yang menggantikan Google Cloud Messaging (GCM).

Menangani notifikasi push

Setelah mendapatkan kredensial yang diperlukan untuk mengirim notifikasi push, Anda dapat memperbarui aplikasi agar mereka dapat menerima notifikasi push. Untuk informasi selengkapnya, lihat [Pemberitahuan Push — Memulai](#) di AWS Amplify dokumentasi

Mendefinisikan audiens Anda ke Amazon Pinpoint

Di Amazon Pinpoint, setiap anggota audiens Anda diwakili oleh satu atau beberapa titik akhir. Saat Anda menggunakan Amazon Pinpoint untuk mengirim pesan, Anda mengarahkan pesan ke titik akhir yang mewakili anggota audiens target Anda. Setiap definisi titik akhir mencakup tujuan pesan—seperti token perangkat, alamat email, atau nomor telepon. Ini juga mencakup data tentang pengguna Anda dan perangkat mereka. Sebelum Anda menganalisis, menyegmentasikan, atau melibatkan audiens Anda, langkah pertama adalah menambahkan titik akhir ke proyek Amazon Pinpoint Anda.

Untuk menambahkan titik akhir, Anda dapat:

- Integrasikan Amazon Pinpoint dengan Android, iOS, atau JavaScript klien Anda sehingga titik akhir ditambahkan secara otomatis saat pengguna mengunjungi aplikasi Anda.
- Gunakan Amazon Pinpoint API untuk menambahkan titik akhir secara individual atau dalam batch.
- Impor definisi titik akhir yang disimpan di luar Amazon Pinpoint.

Setelah menambahkan titik akhir, Anda dapat:

- Lihat analisis tentang audiens Anda di konsol Amazon Pinpoint.
- Pelajari tentang audiens Anda dengan mencari atau mengekspor data titik akhir.
- Tentukan segmen audiens berdasarkan atribut endpoint, seperti data demografis atau minat pengguna.
- Libatkan audiens target Anda dengan kampanye pesan khusus.
- Kirim pesan langsung ke daftar titik akhir.

Gunakan topik di bagian ini untuk menambahkan, memperbarui, dan menghapus titik akhir menggunakan Amazon Pinpoint API. Jika Anda ingin menambahkan titik akhir secara otomatis dari Android, iOS, atau JavaScript klien Anda, lihat [Mendaftarkan titik akhir dalam aplikasi Anda](#) sebagai gantinya.

Topik

- [Menambahkan titik akhir ke Amazon Pinpoint](#)
- [Mengaitkan pengguna dengan titik akhir Amazon Pinpoint](#)
- [Menambahkan sejumlah titik akhir ke Amazon Pinpoint](#)

- [Mengimpor titik akhir ke Amazon Pinpoint](#)
- [Menghapus titik akhir dari Amazon Pinpoint](#)
- [Mengelola jumlah endpoint maksimum anggota audiens](#)

Menambahkan titik akhir ke Amazon Pinpoint

Titik akhir mewakili tujuan yang dapat Anda kirim pesan—seperti perangkat seluler, nomor telepon, atau alamat email. Sebelum Anda dapat mengirim pesan kepada anggota audiens Anda, Anda harus menentukan satu atau lebih titik akhir untuk individu tersebut.

Saat Anda menentukan titik akhir, Anda menentukan saluran dan alamat. Saluran adalah jenis platform yang Anda gunakan untuk mengirim pesan ke titik akhir. Contoh saluran termasuk layanan pemberitahuan push, SMS, atau email. Alamat menentukan tempat mengirim pesan ke titik akhir, seperti token perangkat, nomor telepon, atau alamat email.

Untuk menambahkan detail lebih lanjut tentang audiens Anda, Anda dapat memperkaya titik akhir Anda dengan atribut khusus dan standar. Atribut ini mungkin termasuk data tentang pengguna Anda, preferensi mereka, perangkat mereka, versi klien yang mereka gunakan, atau lokasi mereka. Saat menambahkan jenis data ini ke titik akhir, Anda dapat:

- Lihat bagan tentang audiens Anda di konsol Amazon Pinpoint.
- Segmentasikan audiens Anda berdasarkan atribut endpoint sehingga Anda dapat mengirim pesan ke audiens target yang tepat.
- Personalisasi pesan Anda dengan memasukkan variabel pesan yang diganti dengan nilai atribut endpoint.

Aplikasi seluler atau JavaScript klien mendaftarkan titik akhir secara otomatis jika Anda mengintegrasikan Amazon Pinpoint menggunakan SDK Seluler atau AWS pustaka AWS Amplify. JavaScript Klien mendaftarkan titik akhir untuk setiap pengguna baru, dan memperbarui titik akhir untuk pengguna yang kembali. Untuk mendaftarkan titik akhir dari ponsel atau JavaScript klien, lihat [Mendaftarkan titik akhir dalam aplikasi Anda](#).

Contoh

Contoh berikut menunjukkan cara menambahkan titik akhir ke proyek Amazon Pinpoint. Titik akhir mewakili anggota audiens yang tinggal di Seattle dan menggunakan iPhone. Orang ini dapat dikirim

melalui layanan Pemberitahuan Push Apple (APN). Alamat titik akhir adalah token perangkat yang disediakan oleh APN.

AWS CLI

Anda dapat menggunakan Amazon Pinpoint dengan menjalankan perintah dengan file. AWS CLI

Example Perbarui perintah titik akhir

Untuk menambah atau memperbarui titik akhir, gunakan perintah [update-endpoint](#):

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

Di mana:

- `application-id` adalah ID proyek Amazon Pinpoint tempat Anda menambahkan atau memperbarui titik akhir.
- `example-endpoint` adalah ID yang Anda tetapkan ke titik akhir baru, atau ID dari titik akhir yang ada yang Anda perbarui.
- `endpoint-request-file.json` adalah path file ke file JSON lokal yang berisi input untuk parameter. `--endpoint-request`

Example File permintaan titik akhir

`update-endpoint` Perintah contoh menggunakan file JSON sebagai argumen untuk `--endpoint-request` parameter. File ini berisi definisi titik akhir seperti berikut:

```
{  
  "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  
  }  
}
```

Untuk atribut yang dapat Anda gunakan untuk menentukan titik akhir, lihat [EndpointRequest](#) skema di Referensi API Amazon Pinpoint.

AWS SDK for Java

Anda dapat menggunakan Amazon Pinpoint API di aplikasi Java Anda dengan menggunakan klien yang disediakan oleh AWS SDK for Java

Example Code

Untuk menambahkan titik akhir, inisialisasi [EndpointRequest](#) objek, dan meneruskannya ke [updateEndpoint](#) metode klien: `AmazonPinpoint`

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



```
public static void main(String[] args) {

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

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

    String applicationId = args[0];

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

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

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

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

Anda dapat menggunakan Amazon Pinpoint dengan membuat permintaan HTTP langsung ke REST API.

Example Permintaan titik akhir PUT

Untuk menambahkan titik akhir, keluarkan PUT permintaan ke sumber daya [Endpoint](#) di URI berikut:

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

Di mana:

- application-id adalah ID proyek Amazon Pinpoint tempat Anda menambahkan atau memperbarui titik akhir.
- endpoint-id adalah ID yang Anda tetapkan ke titik akhir baru, atau ID dari titik akhir yang ada yang Anda perbarui.

Dalam permintaan Anda, sertakan header yang diperlukan, dan berikan [EndpointRequestJSON](#) sebagai isi:

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180428T004705Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180428/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-type;host;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
```

```
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

Jika permintaan Anda berhasil, Anda menerima tanggapan seperti berikut:

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

Informasi terkait

Untuk informasi selengkapnya tentang sumber daya Endpoint di Amazon Pinpoint API, termasuk metode HTTP yang didukung dan parameter permintaan, [lihat](#) Endpoint di Referensi API Amazon Pinpoint.

Untuk informasi selengkapnya tentang mempersonalisasi pesan dengan variabel, lihat [Variabel pesan](#) di Panduan Pengguna Amazon Pinpoint.

Untuk informasi tentang kuota yang berlaku untuk titik akhir, seperti jumlah atribut yang dapat Anda tetapkan, lihat [the section called "Kuota titik akhir"](#)

Mengaitkan pengguna dengan titik akhir Amazon Pinpoint

Endpoint dapat mencakup atribut yang mendefinisikan pengguna, yang mewakili seseorang di audiens Anda. Misalnya, pengguna mungkin mewakili seseorang yang menginstal aplikasi seluler Anda, atau seseorang yang memiliki akun di situs web Anda.

Anda menentukan pengguna dengan menentukan ID pengguna unik dan, secara opsional, atribut pengguna kustom. Jika seseorang menggunakan aplikasi Anda di beberapa perangkat, atau jika orang tersebut dapat mengirim pesan di beberapa alamat, Anda dapat menetapkan ID pengguna

yang sama ke beberapa titik akhir. Dalam hal ini, Amazon Pinpoint menyinkronkan atribut pengguna di seluruh titik akhir. Jadi, jika Anda menambahkan atribut pengguna ke satu titik akhir, Amazon Pinpoint menambahkan atribut tersebut ke setiap titik akhir yang menyertakan ID pengguna yang sama.

Anda dapat menambahkan atribut pengguna untuk melacak data yang berlaku untuk individu dan tidak bervariasi berdasarkan perangkat yang digunakan orang tersebut. Misalnya, Anda dapat menambahkan atribut untuk nama, usia, atau status akun seseorang.

Tip

Jika aplikasi Anda menggunakan kumpulan pengguna Amazon Cognito untuk menangani autentikasi pengguna, Amazon Cognito dapat menambahkan ID pengguna dan atribut ke titik akhir Anda secara otomatis. Untuk nilai ID pengguna titik akhir, Amazon Cognito menetapkan nilai sub yang ditetapkan ke pengguna di kumpulan pengguna. Untuk mempelajari cara menambahkan pengguna dengan Amazon Cognito, lihat [Menggunakan analitik pinpoint amazon dengan kumpulan pengguna amazon cognito](#) di Panduan Pengembang Amazon Cognito.

Setelah Anda menambahkan definisi pengguna ke titik akhir Anda, Anda memiliki lebih banyak opsi untuk cara Anda mengelompokkan audiens Anda. Anda dapat menentukan segmen berdasarkan atribut pengguna, atau Anda dapat menentukan segmen dengan mengimpor daftar ID pengguna. Saat Anda mengirim pesan ke segmen yang didasarkan pada pengguna, tujuan potensial mencakup setiap titik akhir yang terkait dengan setiap pengguna di segmen tersebut.

Anda juga memiliki lebih banyak opsi untuk bagaimana Anda mengirim pesan kepada audiens Anda. Anda dapat menggunakan kampanye untuk mengirim pesan ke segmen pengguna, atau Anda dapat mengirim pesan langsung ke daftar ID pengguna. Untuk mempersonalisasi pesan, Anda dapat menyertakan variabel pesan yang diganti dengan nilai atribut pengguna.

Contoh

Contoh berikut menunjukkan cara menambahkan definisi pengguna ke titik akhir.

AWS CLI

Anda dapat menggunakan Amazon Pinpoint dengan menjalankan perintah dengan file. AWS CLI

Example Perbarui perintah titik akhir

Untuk menambahkan pengguna ke titik akhir, gunakan perintah [update-endpoint](#). Untuk `--endpoint-request` parameter, Anda dapat menentukan titik akhir baru, yang dapat menyertakan pengguna. Atau, untuk memperbarui titik akhir yang ada, Anda hanya dapat memberikan atribut yang ingin Anda ubah. Contoh berikut menambahkan pengguna ke titik akhir yang ada dengan hanya menyediakan atribut pengguna:

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint tempat Anda menambahkan atau memperbarui titik akhir.
- *endpoint-id* adalah ID yang Anda tetapkan ke titik akhir baru, atau ID dari titik akhir yang ada yang Anda perbarui.
- *endpoint-request-file.json* adalah path file ke file JSON lokal yang berisi input untuk parameter. `--endpoint-request`

Example File permintaan titik akhir

`update-endpoint` Perintah contoh menggunakan file JSON sebagai argumen untuk `--endpoint-request` parameter. File ini berisi definisi pengguna seperti berikut:

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

Untuk atribut yang dapat Anda gunakan untuk mendefinisikan pengguna, lihat User objek dalam [EndpointRequest](#) skema di Referensi API Amazon Pinpoint.

AWS SDK for Java

Anda dapat menggunakan Amazon Pinpoint API di aplikasi Java Anda dengan menggunakan klien yang disediakan oleh AWS SDK for Java

Example Code

Untuk menambahkan pengguna ke titik akhir, inisialisasi [EndpointRequest](#) objek, dan meneruskannya ke [updateEndpoint](#) metode klien. AmazonPinpoint Anda dapat menggunakan objek ini untuk menentukan titik akhir baru, yang dapat menyertakan pengguna. Atau, untuk memperbarui titik akhir yang ada, Anda dapat memperbarui hanya properti yang ingin Anda ubah. Contoh berikut menambahkan pengguna ke endpoint yang ada dengan menambahkan [EndpointUser](#) objek ke EndpointRequest objek:

```
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);
    }
}
```

Untuk contoh SDK lengkap, lihat [AddExampleUser.java](#) on. [GitHub](#)

HTTP

Anda dapat menggunakan Amazon Pinpoint dengan membuat permintaan HTTP langsung ke REST API.

Example Masukkan permintaan titik akhir dengan definisi pengguna

Untuk menambahkan pengguna ke titik akhir, keluarkan PUT permintaan ke sumber daya [Endpoint](#) di URI berikut:

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint tempat Anda menambahkan atau memperbarui titik akhir.
- *endpoint-id* adalah ID yang Anda tetapkan ke titik akhir baru, atau ID dari titik akhir yang ada yang Anda perbarui.

Dalam permintaan Anda, sertakan header yang diperlukan, dan berikan [EndpointRequestJSON](#) sebagai badan. Badan permintaan dapat menentukan titik akhir baru, yang dapat menyertakan pengguna. Atau, untuk memperbarui titik akhir yang ada, Anda hanya dapat memberikan atribut yang ingin Anda ubah. Contoh berikut menambahkan pengguna ke titik akhir yang ada dengan hanya menyediakan atribut pengguna:

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-type;host;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

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

```
}
```

Jika permintaan berhasil, Anda menerima tanggapan seperti berikut:

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

Informasi terkait

Untuk informasi selengkapnya tentang sumber daya Endpoint di Amazon Pinpoint API, termasuk metode HTTP yang didukung dan parameter permintaan, [lihat](#) Endpoint di Referensi API Amazon Pinpoint.

Untuk informasi selengkapnya tentang mempersonalisasi pesan dengan variabel, lihat [Variabel pesan](#) di Panduan Pengguna Amazon Pinpoint.

Untuk mempelajari cara menentukan segmen dengan mengimpor daftar ID pengguna, lihat [Mengimpor segmen](#) di Panduan Pengguna Amazon Pinpoint.

Untuk informasi tentang mengirim pesan langsung ke hingga 100 ID pengguna, lihat [Pesan pengguna di Referensi](#) API Amazon Pinpoint.

Untuk informasi tentang kuota yang berlaku untuk titik akhir, termasuk jumlah atribut pengguna yang dapat Anda tetapkan, lihat [the section called "Kuota titik akhir"](#)

Menambahkan sejumlah titik akhir ke Amazon Pinpoint

Anda dapat menambahkan atau memperbarui beberapa titik akhir dalam satu operasi dengan menyediakan titik akhir dalam batch. Setiap permintaan batch dapat mencakup hingga 100 definisi titik akhir.

Jika Anda ingin menambahkan atau memperbarui lebih dari 100 titik akhir dalam satu operasi, lihat [Mengimpor titik akhir ke Amazon Pinpoint](#) sebagai gantinya.

Contoh

Contoh berikut menunjukkan cara menambahkan dua titik akhir sekaligus dengan menyertakan titik akhir dalam permintaan batch.

AWS CLI

Anda dapat menggunakan Amazon Pinpoint dengan menjalankan perintah dengan file. AWS CLI

Example Perbarui perintah batch titik akhir

Untuk mengirimkan permintaan batch endpoint, gunakan [update-endpoints-batch](#) perintah:

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint tempat Anda menambahkan atau memperbarui titik akhir.
- *endpoint_batch_request_file.json* adalah path file ke file JSON lokal yang berisi input untuk parameter. --endpoint-batch-request

Example File permintaan batch titik akhir

update-endpoints-batchPerintah contoh menggunakan file JSON sebagai argumen untuk --endpoint-request parameter. File ini berisi sekumpulan definisi titik akhir seperti berikut:

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

Untuk atribut yang dapat Anda gunakan untuk menentukan kumpulan titik akhir, lihat [EndpointBatchRequest](#) skema di Referensi API Amazon Pinpoint.

AWS SDK for Java

Anda dapat menggunakan Amazon Pinpoint API di aplikasi Java Anda dengan menggunakan klien yang disediakan oleh AWS SDK for Java

Example Code

Untuk mengirimkan permintaan batch endpoint, inisialisasi [EndpointBatchRequest](#) objek, dan meneruskannya ke [updateEndpointsBatch](#) metode klien. AmazonPinpoint Contoh berikut mengisi [EndpointBatchRequest](#) objek dengan dua [EndpointBatchItem](#) objek:

```
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);
        }
    }
}

```

Untuk contoh SDK lengkap, lihat [AddExampleEndpoints.java](#) on [GitHub](#)

HTTP

Anda dapat menggunakan Amazon Pinpoint dengan membuat permintaan HTTP langsung ke REST API.

Example Masukkan permintaan titik akhir

Untuk mengirimkan permintaan batch titik akhir, terbitkan PUT permintaan ke sumber daya [Endpoints](#) di URI berikut:

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

Di mana *application-id* adalah ID proyek Amazon Pinpoint tempat Anda menambahkan atau memperbarui titik akhir.

Dalam permintaan Anda, sertakan header yang diperlukan, dan berikan [EndpointBatchRequest](#) JSON sebagai isi:

```
PUT /v1/apps/application_id/endpoints HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180501T184948Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-type;host;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

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

```

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

```

Jika permintaan Anda berhasil, Anda menerima tanggapan seperti berikut:

```

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

```

Informasi terkait

Untuk informasi selengkapnya tentang sumber daya Endpoint di Amazon Pinpoint API, termasuk metode HTTP yang didukung dan parameter permintaan, [lihat](#) Endpoint di Referensi API Amazon Pinpoint.

Mengimpor titik akhir ke Amazon Pinpoint

Anda dapat menambahkan atau memperbarui titik akhir dalam jumlah besar dengan mengimpornya dari bucket Amazon S3. Mengimpor titik akhir berguna jika Anda memiliki catatan tentang audiens

di luar Amazon Pinpoint, dan Anda ingin menambahkan informasi ini ke proyek Amazon Pinpoint. Dalam hal ini, Anda akan:

1. Buat definisi titik akhir yang didasarkan pada data audiens Anda sendiri.
2. Simpan definisi titik akhir ini dalam satu atau beberapa file, dan unggah file ke bucket Amazon S3.
3. Tambahkan titik akhir ke project Amazon Pinpoint Anda dengan mengimpornya dari bucket.

Setiap pekerjaan impor dapat mentransfer hingga 1 GB data. Dalam pekerjaan biasa, di mana setiap titik akhir adalah 4 KB atau kurang, Anda dapat mengimpor sekitar 250.000 titik akhir. Anda dapat menjalankan hingga dua pekerjaan impor bersamaan per AWS akun. Jika Anda membutuhkan lebih banyak bandwidth untuk pekerjaan impor Anda, Anda dapat mengirimkan permintaan peningkatan kuota layanan keAWS Support. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#).

Sebelum kamu memulai

Sebelum Anda dapat mengimpor titik akhir, Anda memerlukan sumber daya berikut di AWS akun Anda:

- Ember Amazon S3. Untuk membuat bucket, lihat [Membuat bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Peran AWS Identity and Access Management (IAM) yang memberikan izin baca Amazon Pinpoint untuk bucket Amazon S3 Anda. Untuk membuat peran, lihat [Peran IAM untuk mengimpor titik akhir atau segmen](#).

Contoh

Contoh berikut menunjukkan cara menambahkan definisi titik akhir ke bucket Amazon S3 Anda, lalu mengimpor titik akhir tersebut ke dalam project Amazon Pinpoint.

File dengan definisi titik akhir

File yang Anda tambahkan ke bucket Amazon S3 dapat berisi definisi titik akhir dalam format CSV atau JSON yang dibatasi baris baru. Untuk atribut yang dapat Anda gunakan untuk menentukan titik akhir, lihat skema [EndpointRequest](#)JSON di Referensi API Amazon Pinpoint.

CSV

Anda dapat mengimpor titik akhir yang ditentukan dalam file CSV, seperti pada contoh berikut:

```
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
```

Baris pertama adalah header, yang berisi atribut endpoint. Tentukan atribut bersarang dengan menggunakan notasi titik, seperti pada. `Location.Country`

Baris berikutnya menentukan titik akhir dengan memberikan nilai untuk masing-masing atribut di header.

Untuk memasukkan koma atau tanda kutip ganda dalam suatu nilai, lampirkan nilainya dalam tanda kutip ganda, seperti dalam. `"aaa,bbb"`

Jeda baris tidak didukung dalam nilai di CSV.

JSON

Anda dapat mengimpor titik akhir yang ditentukan dalam file JSON yang dibatasi baris baru, seperti pada contoh berikut:

```
{"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"}}
```

Dalam format ini, setiap baris adalah objek JSON lengkap yang berisi definisi titik akhir individu.

Impor permintaan pekerjaan

Contoh berikut menunjukkan cara menambahkan definisi titik akhir ke Amazon S3 dengan mengunggah file lokal ke bucket. Kemudian, contoh mengimpor definisi titik akhir ke dalam proyek Amazon Pinpoint.

AWS CLI

Anda dapat menggunakan Amazon Pinpoint dengan menjalankan perintah dengan file. AWS CLI

Example Perintah S3 CP

Untuk mengunggah file lokal ke bucket Amazon S3, gunakan perintah Amazon [cpS3](#):

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

Di mana:

- *./endpoints-file* adalah path file ke file lokal yang berisi definisi endpoint.
- *bucket-name/prefix/* adalah nama bucket Amazon S3 Anda dan, secara opsional, awalan yang membantu Anda mengatur objek di bucket secara hierarkis. Misalnya, awalan yang berguna mungkin *pinpoint/imports/endpoints/*.

Example Buat perintah pekerjaan impor

Untuk mengimpor definisi titik akhir dari bucket Amazon S3, gunakan [create-import-job](#) perintah:

```
$ 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
```

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint tempat Anda mengimpor titik akhir.
- *bucket-name/prefix/key* adalah lokasi di Amazon S3 yang berisi satu atau lebih objek untuk diimpor. Lokasi dapat diakhiri dengan kunci untuk objek individual, atau dapat diakhiri dengan awalan yang memenuhi syarat beberapa objek.
- *iam-import-role-arn* adalah Nama Sumber Daya Amazon (ARN) dari peran IAM yang memberikan akses baca Amazon Pinpoint ke bucket.
- *format* dapat berupa JSON atau CSV, tergantung pada format yang Anda gunakan untuk menentukan titik akhir Anda. Jika lokasi Amazon S3 menyertakan beberapa objek format campuran, Amazon Pinpoint hanya mengimpor objek yang cocok dengan format yang ditentukan.

- RegisterEndpoints bisa salah satu true atau false. Ketika disetel ke true, pekerjaan impor akan mendaftarkan titik akhir dengan Amazon Pinpoint, saat definisi titik akhir diimpor.

RegisterEndpoints dan DefineSegments kombinasi

RegisterEndpoints	DefineSegments	Deskripsi
betul	betul	Amazon Pinpoint akan mengimpor titik akhir dan membuat segmen yang berisi titik akhir.
betul	SALAH	Amazon Pinpoint akan mengimpor titik akhir dan tidak membuat segmen.
SALAH	betul	Amazon Pinpoint akan mengimpor titik akhir dan membuat segmen yang berisi titik akhir. Titik akhir tidak akan disimpan dan tidak akan menimpa titik akhir yang ada.
salah	salah	Amazon Pinpoint akan menolak permintaan ini.

Tanggapan tersebut mencakup detail tentang pekerjaan impor:

```
{
  "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"
  }
}
```

Respons memberikan ID pekerjaan dengan `Id` atribut. Anda dapat menggunakan ID ini untuk memeriksa status pekerjaan impor saat ini.

Example Dapatkan perintah pekerjaan impor

Untuk memeriksa status pekerjaan impor saat ini, gunakan `get-import-job` perintah:

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

Di mana:

- `application-id` adalah ID proyek Amazon Pinpoint tempat pekerjaan impor dimulai.
- `job-id` adalah ID dari pekerjaan impor yang Anda periksa.

Respons terhadap perintah ini memberikan status pekerjaan impor saat ini:

```
{
  "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"  
    }  
}
```

Respons memberikan status pekerjaan dengan `JobStatus` atribut.

AWS SDK for Java

Anda dapat menggunakan Amazon Pinpoint API di aplikasi Java Anda dengan menggunakan klien yang disediakan oleh AWS SDK for Java

Example Code

Untuk mengunggah file dengan definisi titik akhir ke Amazon S3, gunakan [putObject](#) metode klien. `AmazonS3`

Untuk mengimpor titik akhir ke proyek Amazon Pinpoint, inialisasi [CreateImportJobRequest](#) objek. Kemudian, berikan objek ini ke [createImportJob](#) metode `AmazonPinpoint` klien.

```
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

Anda dapat menggunakan Amazon Pinpoint dengan membuat permintaan HTTP langsung ke REST API.

Example Permintaan objek S3 PUT

Untuk menambahkan definisi titik akhir Anda ke bucket, gunakan operasi objek [PUT Amazon S3](#), dan berikan definisi titik akhir sebagai isi:

```

PUT /prefix/key HTTP/1.1
Content-Type: text/plain
Accept: application/json
Host: bucket-name.s3.amazonaws.com
X-Amz-Content-Sha256:
    c430dc094b0cec2905bc88d96314914d058534b14e2bc6107faa9daa12fdff2d
X-Amz-Date: 20180605T184132Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/s3/aws4_request, SignedHeaders=accept;cache-control;content-

```

```
length;content-type;host;postman-token;x-amz-content-sha256;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache
```

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

Di mana:

- /prefix/key adalah awalan dan nama kunci untuk objek yang akan berisi definisi endpoint setelah upload. Anda dapat menggunakan awalan untuk mengatur objek Anda secara hierarkis. Misalnya, awalan yang berguna mungkin pinpoint/imports/endpoints/.
- bucket-name adalah nama bucket Amazon S3 tempat Anda menambahkan definisi titik akhir.

Example Permintaan pekerjaan impor POST

Untuk mengimpor definisi titik akhir dari bucket Amazon S3, keluarkan permintaan POST ke sumber daya [Impor pekerjaan](#). Dalam permintaan Anda, sertakan header yang diperlukan dan berikan [ImportJobRequest](#) JSON sebagai isi:

```
POST /v1/apps/application_id/jobs/import HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T214912Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3Url": "s3://bucket-name/prefix/key",
  "RoleArn": "iam-import-role-arn",
```

```
"Format": "format",
"RegisterEndpoints": true
}
```

Di mana:

- application-id adalah ID proyek Amazon Pinpoint tempat Anda mengimpor titik akhir.
- bucket-name/prefix/key adalah lokasi di Amazon S3 yang berisi satu atau lebih objek untuk diimpor. Lokasi dapat diakhiri dengan kunci untuk objek individual, atau dapat diakhiri dengan awalan yang memenuhi syarat beberapa objek.
- iam-import-role-arn adalah Nama Sumber Daya Amazon (ARN) dari peran IAM yang memberikan akses baca Amazon Pinpoint ke bucket.
- format dapat berupa JSON atau CSV, tergantung pada format yang Anda gunakan untuk menentukan titik akhir Anda. Jika lokasi Amazon S3 menyertakan beberapa file format campuran, Amazon Pinpoint hanya mengimpor file yang cocok dengan format yang ditentukan.

Jika permintaan Anda berhasil, Anda menerima tanggapan seperti berikut:

```
{
  "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
  }
}
```

Respons memberikan ID pekerjaan dengan Id atribut. Anda dapat menggunakan ID ini untuk memeriksa status pekerjaan impor saat ini.

Example DAPATKAN permintaan pekerjaan impor

Untuk memeriksa status pekerjaan impor saat ini, terbitkan GET permintaan ke sumber daya [pekerjaan Impor](#):

```
GET /v1/apps/application_id/jobs/import/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T220744Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-type;host;postman-token;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache
```

Di mana:

- *application_id* adalah ID proyek Amazon Pinpoint yang tugas impornya dimulai.
- *job_id* adalah ID dari pekerjaan impor yang Anda periksa.

Jika permintaan Anda berhasil, Anda menerima tanggapan seperti berikut:

```
{
  "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
  }
}
```

Respons memberikan status pekerjaan dengan `JobStatus` atribut.

Informasi terkait

Untuk informasi selengkapnya tentang sumber daya Impor Pekerjaan di Amazon Pinpoint API, termasuk metode HTTP yang didukung dan parameter permintaan, lihat [Mengimpor lowongan kerja](#) di Referensi Amazon Pinpoint API.

Menghapus titik akhir dari Amazon Pinpoint

Anda dapat menghapus titik akhir ketika Anda tidak lagi ingin mengirim pesan ke tujuan tertentu—seperti ketika tujuan menjadi tidak dapat dijangkau, atau ketika pelanggan menutup akun.

Contoh

Contoh berikut menunjukkan cara menghapus titik akhir.

AWS CLI

Anda dapat menggunakan Amazon Pinpoint dengan menjalankan perintah dengan file. AWS CLI

Example Hapus perintah titik akhir

Untuk menghapus titik akhir, gunakan [delete-endpoint](#) perintah:

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint yang berisi endpoint.
- *endpoint-id* adalah ID dari titik akhir yang Anda hapus.

Respons terhadap perintah ini adalah definisi JSON dari titik akhir yang Anda hapus.

AWS SDK for Java

Anda dapat menggunakan Amazon Pinpoint API di aplikasi Java Anda dengan menggunakan klien yang disediakan oleh AWS SDK for Java

Example Code

Untuk menghapus titik akhir, gunakan [deleteEndpoint](#) metode AmazonPinpoint klien. Berikan [DeleteEndpointRequest](#) objek sebagai argumen metode:

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

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

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

            Usage:  <appName> <endpointId >

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

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

        String appId = args[0];
        String endpointId = args[1];
```

```
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

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

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

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

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

Untuk contoh SDK lengkap, lihat [DeleteEndpoint.java](#) on [GitHub](#)

HTTP

Anda dapat menggunakan Amazon Pinpoint dengan membuat permintaan HTTP langsung ke REST API.

Example HAPUS permintaan titik akhir

Untuk menghapus titik akhir, keluarkan DELETE permintaan ke sumber daya [Endpoint](#):

```
DELETE /v1/apps/application-id/endpoints/endpoint-id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
```

```
Cache-Control: no-cache
```

Di mana:

- application-id adalah ID proyek Amazon Pinpoint yang berisi endpoint.
- endpoint-id adalah ID dari titik akhir yang Anda hapus.

Respons terhadap permintaan ini adalah definisi JSON dari titik akhir yang Anda hapus.

Mengelola jumlah endpoint maksimum anggota audiens

Setiap anggota audiens Anda dapat memiliki maksimal 15 titik akhir yang terkait dengan mereka UserId, lihat [Kuota titik akhir](#). Jika Anda mencoba untuk menambahkan endpoint ke-16 kemudian, tergantung pada ChannelType, Anda akan mendapatkan BadRequestException atau itu akan berhasil dengan menghapus endpoint dengan yang tertua EffectiveDate.

Menambahkan endpoint ke-16

- Jika jenis saluran baru untuk titik akhir adalah SMS, PUSH, VOICE, EMAIL, CUSTOM atau IN_APP maka BadRequestException dikembalikan karena anggota audiens berada pada jumlah titik akhir maksimum mereka. Anda perlu menghapus titik akhir yang terkait dengan anggota audiens dan coba lagi, lihat [Menghapus titik akhir dari Amazon Pinpoint](#).
- Jika jenis saluran baru untuk titik akhir adalah ADM, GCM, APNS_VOIP, APNS_VOIP_SANDBOX atau BAIDU:
 - Pastikan setidaknya satu titik akhir yang saat ini terkait dengan anggota audiens memiliki ChannelType ADM, GCM, APNS_VOICE, APNS_VOICE, APNS_VOIP_SANDBOX atau BAIDU. Jika tidak BadRequestException akan dikembalikan dan endpoint perlu dihapus sebelum Anda mencoba lagi, lihat [Menghapus titik akhir dari Amazon Pinpoint](#).
 - Jika tidak, titik akhir dengan yang tertua EffectiveDate diatur ke INACTIVE tempat ADM, GCM, APNS_VOIP, APNS_VOIP_SANDBOX atau BAIDU. ChannelType
 - UserId dari endpoint lama dihapus.
 - Endpoint baru dikaitkan dengan anggota audiens dan mereka masih memiliki jumlah titik akhir maksimum.

Endpoint dapat diaktifkan kembali dengan mengatur Status ke ACTIVE dan menambahkan UserId kembali ke titik akhir.

Mengakses data pemirsa di Amazon Pinpoint

Saat Anda menambahkan titik akhir ke Amazon Pinpoint, titik ini tumbuh sebagai repositori data audiens. Data ini terdiri dari:

- Titik akhir yang Anda tambahkan atau perbarui dengan menggunakan Amazon Pinpoint API.
- Titik akhir yang ditambahkan atau diperbarui kode klien Anda saat pengguna datang ke aplikasi Anda.

Seiring pertumbuhan dan perubahan audiens Anda, begitu juga data endpoint Anda. Untuk melihat informasi terbaru yang dimiliki Amazon Pinpoint tentang audiens Anda, Anda dapat mencari titik akhir secara individual, atau Anda dapat mengekspor semua titik akhir untuk proyek Amazon Pinpoint. Dengan melihat data endpoint, Anda dapat mempelajari karakteristik audiens yang Anda rekam di titik akhir, seperti:

- Perangkat dan platform pengguna Anda.
- Zona waktu pengguna Anda.
- Versi aplikasi Anda yang diinstal pada perangkat pengguna.
- Lokasi pengguna Anda, seperti kota atau negara mereka.
- Atribut atau metrik kustom apa pun yang Anda rekam.

Konsol Amazon Pinpoint juga menyediakan analitik untuk demografi dan atribut kustom yang ditangkap di titik akhir Anda.

Sebelum Anda dapat mencari titik akhir, Anda harus menemukannya ke proyek Amazon Pinpoint Anda. Untuk menambahkan titik akhir, lihat [Mendefinisikan audiens Anda ke Amazon Pinpoint](#).

Gunakan topik di bagian ini untuk mencari atau mengekspor titik akhir dengan menggunakan API Amazon Pinpoint.

Topik

- [Mencari titik akhir dengan Amazon Pinpoint](#)
- [Mengekspor titik akhir dari Amazon Pinpoint](#)
- [Daftar ID titik akhir dengan Amazon Pinpoint](#)

Mencari titik akhir dengan Amazon Pinpoint

Anda dapat mencari detail untuk setiap titik akhir individu yang ditambahkan ke proyek Amazon Pinpoint. Detail ini dapat mencakup alamat tujuan untuk pesan Anda, saluran pesan, data tentang perangkat pengguna, data tentang lokasi pengguna, dan atribut kustom apa pun yang Anda rekam di titik akhir.

Untuk mencari titik akhir, Anda memerlukan ID titik akhir. Jika Anda tidak tahu ID, Anda bisa mendapatkan data endpoint dengan mengekspor sebagai gantinya. Untuk mengekspor titik akhir, lihat [the section called "Mengekspor titik akhir"](#).

Contoh

Contoh berikut menunjukkan kepada Anda cara mencari titik akhir individu dengan menentukan ID-nya.

AWS CLI

Anda dapat menggunakan Amazon Pinpoint dengan menjalankan perintah dengan file. AWS CLI

Example Dapatkan perintah endpoint

Untuk mencari titik akhir, gunakan [get-endpoint](#) perintah:

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint yang berisi titik akhir.
- *endpoint-id* adalah ID dari titik akhir yang Anda cari.

Respons terhadap perintah ini adalah definisi JSON dari titik akhir, seperti pada contoh berikut:

```
{  
  "EndpointResponse": {  
    "Address":  
    "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",  
    "ApplicationId": "application-id",  
    "Attributes": {
```

```
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "ChannelType": "APNS",
  "CohortId": "63",
  "CreationDate": "2018-05-01T17:31:01.046Z",
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "EffectiveDate": "2018-05-07T19:03:29.963Z",
  "EndpointStatus": "ACTIVE",
  "Id": "example_endpoint",
  "Location": {
    "City": "Seattle",
    "Country": "US",
    "Latitude": 47.6,
    "Longitude": -122.3,
    "PostalCode": "98121"
  },
  "Metrics": {
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0,
    "technology_interest_level": 9.0
  },
  "OptOut": "ALL",
  "RequestId": "7f546cac-6858-11e8-adcd-2b5a07aab338",
  "User": {
    "UserAttributes": {
      "Gender": "Female",
      "FirstName": "Wang",
      "LastName": "Xiulan",
      "Age": "39"
    },
    "UserId": "example_user"
  }
}
```

```
}  
}
```

AWS SDK for Java

Anda dapat menggunakan Amazon Pinpoint API di aplikasi Java Anda dengan menggunakan klien yang disediakan oleh AWS SDK for Java

Example Code

Untuk mencari titik akhir, inialisasi objek [GetEndpointRequest](#) Kemudian, berikan objek ini ke [getEndpoint](#) metode AmazonPinpoint klien:

```
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");
}
}
```

Untuk mencetak data titik akhir dalam format yang dapat dibaca, contoh ini menggunakan pustaka Google GSON untuk mengonversi `EndpointResponse` objek menjadi string JSON.

HTTP

Anda dapat menggunakan Amazon Pinpoint dengan membuat permintaan HTTP langsung ke REST API.

Example DAPATKAN permintaan titik akhir

Untuk mencari titik akhir, keluarkan GET permintaan ke sumber daya [Endpoint](#):

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint yang berisi titik akhir.
- *endpoint-id* adalah ID dari titik akhir yang Anda cari.

Respons terhadap permintaan ini adalah definisi JSON dari titik akhir, seperti pada contoh berikut:

```
{
  "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"
}
```

Informasi terkait

Untuk informasi selengkapnya tentang sumber daya Endpoint di Amazon Pinpoint API, [lihat](#) Endpoint di Referensi API Amazon Pinpoint.

Mengekspor titik akhir dari Amazon Pinpoint

Untuk mendapatkan semua informasi yang dimiliki Amazon Pinpoint tentang audiens Anda, Anda dapat mengekspor definisi titik akhir yang termasuk dalam proyek. Saat Anda mengekspor, Amazon Pinpoint menempatkan definisi titik akhir dalam bucket Amazon S3 yang Anda tentukan. Mengekspor titik akhir berguna saat Anda ingin:

- Lihat data terbaru tentang titik akhir baru dan yang sudah ada yang didaftarkan aplikasi klien Anda di Amazon Pinpoint.
- Sinkronkan data endpoint di Amazon Pinpoint dengan sistem Customer Relationship Management (CRM) Anda sendiri.
- Buat laporan tentang atau analisis data pelanggan Anda.

Sebelum Anda memulai

Sebelum Anda dapat mengekspor titik akhir, Anda memerlukan sumber daya berikut di AWS akun Anda:

- Ember Amazon S3. Untuk membuat bucket, lihat [Membuat bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Peran AWS Identity and Access Management (IAM) yang memberikan izin menulis Amazon Pinpoint untuk bucket Amazon S3 Anda. Untuk membuat peran, lihat [Peran IAM untuk mengekspor titik akhir atau segmen](#).

Contoh

Contoh berikut menunjukkan cara mengekspor titik akhir dari proyek Amazon Pinpoint, lalu mengunduh titik akhir tersebut dari bucket Amazon S3 Anda.

AWS CLI

Anda dapat menggunakan Amazon Pinpoint dengan menjalankan perintah dengan file. AWS CLI

Example Buat perintah pekerjaan ekspor

Untuk mengekspor titik akhir dalam proyek Amazon Pinpoint Anda, gunakan [create-export-job](#) perintah:

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint yang berisi titik akhir.
- *bucket-name/prefix/* adalah nama bucket Amazon S3 Anda dan, secara opsional, awalan yang membantu Anda mengatur objek di bucket secara hierarkis. Misalnya, awalan yang berguna mungkin `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* adalah Nama Sumber Daya Amazon (ARN) dari peran IAM yang memberikan akses tulis Amazon Pinpoint ke bucket.

Tanggapan terhadap perintah ini memberikan rincian tentang pekerjaan ekspor:

```
{
  "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"
  }
}
```

Respons memberikan ID pekerjaan dengan `Id` atribut. Anda dapat menggunakan ID ini untuk memeriksa status pekerjaan ekspor saat ini.

Example Dapatkan perintah pekerjaan ekspor

Untuk memeriksa status pekerjaan ekspor saat ini, gunakan [get-export-job](#) perintah:

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

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint tempat Anda mengekspor titik akhir.
- *job-id* adalah ID pekerjaan yang Anda periksa.

Tanggapan terhadap perintah ini memberikan status pekerjaan ekspor saat ini:

```
{
  "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"
  }
}
```

Respons memberikan status pekerjaan dengan JobStatus atribut. Saat nilai status pekerjaan COMPLETED, Anda bisa mendapatkan titik akhir yang diekspor dari bucket Amazon S3.

Example Perintah S3 CP

Untuk mengunduh titik akhir yang diekspor, gunakan perintah Amazon [cpS3](#):

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

Di mana:

- *bucket-name/prefix/key* adalah lokasi file.gz yang ditambahkan Amazon Pinpoint ke bucket Anda saat Anda mengekspor titik akhir. File ini berisi definisi titik akhir yang diekspor. Misalnya, di URL `https://PINPOINT-EXAMPLE-BUCKET.s3.us-`

west-2.amazonaws.com/Exports/example.csv, PINPOINT-EXAMPLE-BUCKET adalah nama bucket dan Exports/example.csv merupakan kuncinya. Untuk informasi selengkapnya tentang Kunci, lihat [Kunci](#) di Panduan Pengguna Amazon S3.

- */local/directory/* adalah jalur file ke direktori lokal yang ingin Anda unduh titik akhir.

AWS SDK for Java

Anda dapat menggunakan Amazon Pinpoint API di aplikasi Java Anda dengan menggunakan klien yang disediakan oleh AWS SDK for Java

Example Code

Untuk mengeksport titik akhir dari proyek Amazon Pinpoint, inisialisasi [CreateExportJobRequest](#) objek. Kemudian, berikan objek ini ke [createExportJob](#) metode AmazonPinpoint klien.

Untuk mengunduh titik akhir yang diekspor dari Amazon Pinpoint, gunakan metode [getObject](#) klien. 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);
}
}
}
}

```

Untuk contoh SDK lengkap, lihat [ExportEndpoints.java](#) on [GitHub](#)

HTTP

Anda dapat menggunakan Amazon Pinpoint dengan membuat permintaan HTTP langsung ke REST API.

Example Permintaan pekerjaan ekspor POST

Untuk mengekspor titik akhir dalam proyek Amazon Pinpoint Anda, keluarkan permintaan ke sumber POST daya [Ekspor pekerjaan](#):

```

POST /v1/apps/application_id/jobs/export HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T001238Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-

```

```
control;content-length;content-type;host;postman-token;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache

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

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint yang berisi titik akhir.
- *bucket-name/prefix* adalah nama bucket Amazon S3 Anda dan, secara opsional, awalan yang membantu Anda mengatur objek di bucket secara hierarkis. Misalnya, awalan yang berguna mungkin `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* adalah Nama Sumber Daya Amazon (ARN) dari peran IAM yang memberikan akses tulis Amazon Pinpoint ke bucket.

Tanggapan atas permintaan ini memberikan rincian tentang pekerjaan ekspor:

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

Respons memberikan ID pekerjaan dengan `Id` atribut. Anda dapat menggunakan ID ini untuk memeriksa status pekerjaan ekspor saat ini.

Example DAPATKAN permintaan pekerjaan ekspor

Untuk memeriksa status pekerjaan ekspor saat ini, terbitkan GET permintaan ke sumber daya [pekerjaan Ekspor](#):

```
GET /v1/apps/application_id/jobs/export/job_id HTTP/1.1
```

```
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T002443Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-type;host;postman-token;x-amz-date,
  Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875caced95d1e68d91b4170
Cache-Control: no-cache
```

Di mana:

- *application-id* adalah ID proyek Amazon Pinpoint tempat Anda mengekspor titik akhir.
- *job-id* adalah ID pekerjaan yang Anda periksa.

Tanggapan terhadap permintaan ini memberikan status pekerjaan ekspor saat ini:

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

Respons memberikan status pekerjaan dengan JobStatus atribut. Saat nilai status pekerjaan COMPLETED, Anda bisa mendapatkan titik akhir yang diekspor dari bucket Amazon S3.

Informasi terkait

Untuk informasi selengkapnya tentang sumber daya Ekspor Pekerjaan di Amazon Pinpoint API, termasuk metode HTTP dan parameter permintaan yang didukung, lihat [Mengekspor lowongan kerja](#) di Referensi API Amazon Pinpoint.

Daftar ID titik akhir dengan Amazon Pinpoint

Untuk memperbarui atau menghapus titik akhir, Anda memerlukan ID titik akhir. Jadi, jika Anda ingin melakukan operasi ini di semua titik akhir dalam proyek Amazon Pinpoint, langkah pertama adalah mencantumkan semua ID titik akhir yang termasuk dalam proyek itu. Kemudian, Anda dapat mengulangi ID ini untuk, misalnya, menambahkan atribut secara global atau menghapus semua titik akhir dalam proyek Anda.

Contoh berikut menggunakan AWS SDK for Java dan melakukan hal berikut:

1. Memanggil `exportEndpointsToS3` metode contoh dari kode contoh di [Mengekspor titik akhir dari Amazon Pinpoint](#). Metode ini mengekspor definisi titik akhir dari proyek Amazon Pinpoint. Definisi titik akhir ditambahkan sebagai file gzip ke bucket Amazon S3.
2. Mengunduh file gzip yang diekspor.
3. Membaca file gzip dan memperoleh ID titik akhir dari definisi JSON setiap titik akhir.
4. Mencetak ID titik akhir ke konsol.
5. Membersihkan dengan menghapus file yang Amazon Pinpoint ditambahkan ke 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);
    }
}
```

Untuk contoh SDK lengkap, lihat [ListEndpoints.java](#) on [GitHub](#)

Membuat segmen

Pengguna segmen mewakili subset pengguna berdasarkan karakteristik bersama, seperti bagaimana baru-baru ini pengguna menggunakan aplikasi Anda atau platform perangkat mana yang mereka gunakan. Segmen menunjuk pengguna mana yang menerima pesan yang dikirim oleh kampanye. Tentukan segmen sehingga Anda dapat menjangkau audiens yang tepat saat Anda ingin mengundang pengguna kembali ke aplikasi, membuat penawaran khusus, atau meningkatkan keterlibatan dan pembelian pengguna.

Setelah membuat segmen, Anda dapat menggunakannya dalam satu atau beberapa kampanye. Kampanye mengirimkan pesan yang disesuaikan kepada pengguna di segmen tersebut.

Untuk informasi selengkapnya, lihat [Segmen](#).

Topik

- [Membangun segmen](#)
- [Mengimpor segmen](#)
- [Menyesuaikan segmen dengan AWS Lambda](#)

Membangun segmen

Untuk menjangkau audiens yang dituju untuk kampanye, buat segmen berdasarkan data yang dilaporkan oleh aplikasi Anda. Misalnya, untuk menjangkau pengguna yang belum menggunakan aplikasi Anda baru-baru ini, Anda dapat menentukan segmen untuk pengguna yang belum menggunakan aplikasi Anda dalam 30 hari terakhir.

Membangun segmen dengan AWS SDK for Java

Contoh berikut menunjukkan bagaimana membangun segmen dengan AWS SDK for Java

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
```

```
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

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

```
/**
```

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

```
 *
```

```
 * For more information, see the following documentation topic:
```

```
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
```

```
 */
```

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

```
                Usage:  <appId>
```

```
                Where:
```

```
                appId - The application ID to create a segment for.
```

```
        """;

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

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

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

public static SegmentResponse createSegment(PinpointClient client, String
appId) {
    try {
        Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();

        segmentAttributes.put("Team", AttributeDimension.builder()
            .attributeType(AttributeType.INCLUSIVE)
            .values("Lakers")
            .build());

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

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

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

        SegmentLocation segmentLocation = SegmentLocation
            .builder()
```

```
        .build();

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

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

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

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

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

Saat Anda menjalankan contoh ini, berikut ini dicetak ke jendela konsol IDE Anda:

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

Untuk contoh SDK lengkap, lihat [CreateSegment.java](#) on [GitHub](#)

Mengimpor segmen

Dengan Amazon Pinpoint, Anda dapat menentukan segmen pengguna dengan mengimpor informasi tentang titik akhir yang termasuk dalam segmen tersebut. Endpoint adalah tujuan pengiriman pesan tunggal, seperti token perangkat push seluler, nomor ponsel, atau alamat email.

Mengimpor segmen berguna jika Anda telah membuat segmen pengguna di luar Amazon Pinpoint tetapi Anda ingin melibatkan pengguna Anda dengan kampanye Amazon Pinpoint.

Saat Anda mengimpor segmen, Amazon Pinpoint mendapatkan titik akhir segmen dari Amazon Simple Storage Service (Amazon S3). Sebelum mengimpor, Anda menambahkan titik akhir ke Amazon S3, dan Anda membuat peran IAM yang memberikan akses Amazon Pinpoint ke Amazon S3. Kemudian, Anda memberikan Amazon Pinpoint lokasi Amazon S3 tempat titik akhir disimpan, dan Amazon Pinpoint menambahkan setiap titik akhir ke segmen.

Untuk membuat peran IAM, lihat [Peran IAM untuk mengimpor titik akhir atau segmen](#). Untuk informasi tentang mengimpor segmen menggunakan konsol Amazon Pinpoint, [lihat Mengimpor](#) segmen di Panduan Pengguna Amazon Pinpoint.

Mengimpor segmen

Contoh berikut menunjukkan bagaimana untuk mengimpor segmen dengan menggunakan AWS SDK for Java

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

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

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

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

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

            """;

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

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

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

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

```
        System.out.println("See application " + response.jobStatus() + " for import job
status.");
        pinpoint.close();
    }

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

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

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

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

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

Untuk contoh SDK lengkap, lihat [ImportingSegments.java](#) on [GitHub](#)

Menyesuaikan segmen dengan AWS Lambda

Ini adalah dokumentasi prarilis untuk fitur dalam rilis beta publik. Dokumentasi ini dapat mengalami perubahan.

Anda dapat menggunakan AWS Lambda untuk menyesuaikan bagaimana kampanye Amazon Pinpoint melibatkan audiens target Anda. Dengan AWS Lambda, Anda dapat mengubah segmen kampanye saat Amazon Pinpoint mengirim pesan kampanye.

AWS Lambda adalah layanan komputasi yang dapat Anda gunakan untuk menjalankan kode tanpa perlu menyediakan atau mengelola server. Anda mengemas kode Anda dan mengunggahnya ke Lambda sebagai fungsi Lambda. Lambda menjalankan fungsi saat fungsi dipanggil, yang mungkin dilakukan secara manual oleh Anda atau secara otomatis sebagai respons terhadap peristiwa. Untuk informasi selengkapnya, lihat [Panduan Developer AWS Lambda](#).

Untuk menetapkan fungsi Lambda ke kampanye, Anda menentukan CampaignHook setelah kampanye dengan menggunakan sumber daya Kampanye [di](#) Amazon Pinpoint API. Pengaturan ini termasuk nama fungsi Lambda. Mereka juga menyertakan CampaignHook mode, yang menentukan apakah Amazon Pinpoint menerima nilai kembali dari fungsi tersebut.

Fungsi Lambda yang Anda tetapkan ke kampanye disebut sebagai ekstensi Amazon Pinpoint.

Dengan CampaignHook pengaturan yang ditentukan, Amazon Pinpoint secara otomatis memanggil fungsi Lambda saat menjalankan kampanye, sebelum mengirimkan pesan kampanye. Saat Amazon Pinpoint memanggil fungsi, Amazon Pinpoint menyediakan data peristiwa tentang pengiriman pesan. Data ini mencakup segmen kampanye, yang merupakan daftar titik akhir yang dikirimkan oleh Amazon Pinpoint.

Jika CampaignHook mode diatur ke FILTER, Amazon Pinpoint memungkinkan fungsi untuk memodifikasi dan mengembalikan segmen sebelum mengirim pesan. Misalnya, fungsi mungkin memperbarui definisi titik akhir dengan atribut yang berisi data dari sumber yang berada di luar Amazon Pinpoint. Atau, fungsi tersebut mungkin memfilter segmen dengan menghapus titik akhir tertentu, berdasarkan kondisi dalam kode fungsi Anda. Setelah Amazon Pinpoint menerima segmen yang dimodifikasi dari fungsi Anda, Amazon Pinpoint akan mengirimkan pesan ke setiap titik akhir segmen menggunakan saluran pengiriman kampanye.

Dengan memproses segmen Anda AWS Lambda, Anda memiliki kontrol lebih besar atas siapa Anda mengirim pesan dan apa isi pesan tersebut. Anda dapat menyesuaikan kampanye Anda secara real time, pada saat pesan kampanye dikirim. Memfilter segmen memungkinkan Anda untuk melibatkan subset segmen yang didefinisikan secara lebih sempit. Menambahkan atau memperbarui atribut endpoint juga memungkinkan Anda membuat data baru tersedia untuk variabel pesan.

Note

Anda juga dapat menggunakan CampaignHook pengaturan untuk menetapkan fungsi Lambda yang menangani pengiriman pesan. Jenis fungsi ini berguna untuk mengirimkan pesan melalui saluran khusus yang tidak didukung Amazon Pinpoint, seperti platform media sosial. Untuk informasi selengkapnya, lihat [Membuat saluran khusus di Amazon Pinpoint](#). Saat menjalankan hook Lambda menggunakan Amazon Pinpoint, fungsi Lambda juga harus berada di wilayah yang sama dengan proyek Amazon Pinpoint.

Untuk mengubah segmen kampanye dengan AWS Lambda, pertama-tama buat fungsi yang memproses data peristiwa yang dikirim oleh Amazon Pinpoint dan mengembalikan segmen yang dimodifikasi. Kemudian, otorisasi Amazon Pinpoint untuk menjalankan fungsi dengan menetapkan kebijakan fungsi Lambda. Terakhir, tetapkan fungsi ke satu atau beberapa kampanye dengan menentukan pengaturan CampaignHook

Data peristiwa

Saat Amazon Pinpoint memanggil fungsi Lambda Anda, fungsi ini menyediakan payload berikut sebagai data peristiwa:

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

AWS Lambda meneruskan data acara ke kode fungsi Anda. Data acara menyediakan atribut berikut:

- **MessageConfiguration**— Memiliki struktur yang sama dengan **DirectMessageConfiguration** objek sumber daya [Pesan](#) di Amazon Pinpoint API.
- **ApplicationId**— ID proyek Amazon Pinpoint yang menjadi milik kampanye tersebut.

- **CampaignId**— ID kampanye Amazon Pinpoint tempat fungsi dipanggil.
- **TreatmentId**— ID variasi kampanye yang digunakan untuk pengujian A/B.
- **ActivityId**— ID dari aktivitas yang sedang dilakukan oleh kampanye.
- **ScheduledTimeTanggal dan waktu**, dalam format ISO 8601, saat pesan kampanye akan dikirimkan.
- **Endpoints**— Peta yang mengaitkan ID titik akhir dengan definisi titik akhir. Setiap payload data peristiwa berisi hingga 50 titik akhir. Jika segmen kampanye berisi lebih dari 50 titik akhir, Amazon Pinpoint akan memanggil fungsi berulang kali, dengan hingga 50 titik akhir sekaligus, hingga semua titik akhir diproses.

Membuat fungsi Lambda

Untuk mempelajari cara membuat fungsi Lambda, lihat [Memulai di Panduan AWS Lambda Pengembang](#). Saat Anda membuat fungsi, ingatlah bahwa pengiriman pesan gagal dalam kondisi berikut:

- Fungsi Lambda membutuhkan waktu lebih dari 15 detik untuk mengembalikan segmen yang dimodifikasi.
- Amazon Pinpoint tidak dapat memecahkan kode nilai pengembalian fungsi.
- Fungsi ini membutuhkan lebih dari 3 upaya dari Amazon Pinpoint untuk berhasil memanggilmnya.

Amazon Pinpoint hanya menerima definisi titik akhir dalam nilai pengembalian fungsi. Fungsi tidak dapat memodifikasi elemen lain dalam data peristiwa.

Contoh fungsi Lambda

Fungsi Lambda Anda memproses data peristiwa yang dikirim oleh Amazon Pinpoint, dan mengembalikan titik akhir yang dimodifikasi, seperti yang ditunjukkan oleh contoh handler berikut, yang ditulis dalam 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 meneruskan data peristiwa ke handler sebagai parameter. `event`

Dalam contoh ini, handler iterasi melalui setiap titik akhir dalam `event.Endpoints` objek, dan menambahkan atribut baru, `CreditScore`, ke titik akhir. Nilai `CreditScore` atribut hanyalah angka acak.

`console.log()` Pernyataan mencatat peristiwa di CloudWatch Log.

`callback()` Pernyataan tersebut mengembalikan titik akhir yang dimodifikasi ke Amazon Pinpoint. Biasanya, `callback` parameter opsional dalam fungsi Lambda Node.js, tetapi diperlukan dalam konteks ini karena fungsi harus mengembalikan titik akhir yang diperbarui ke Amazon Pinpoint.

Fungsi Anda harus mengembalikan titik akhir dalam format yang sama yang disediakan oleh data peristiwa, yaitu peta yang mengaitkan ID titik akhir dengan definisi titik akhir, seperti pada contoh berikut:

```

{
  "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": {}
  }
}
```

Fungsi contoh memodifikasi dan mengembalikan event. Endpoints objek yang diterima dalam data peristiwa.

Secara opsional, Anda dapat menyertakan `BodyOverride` atribut `TitleOverride` dan dalam definisi titik akhir yang Anda kembalikan.

Note

Saat Anda menggunakan solusi ini untuk mengirim pesan, Amazon Pinpoint menghormati `BodyOverride` atribut `TitleOverride` dan hanya untuk titik akhir yang nilai atributnya adalah salah satu dari yang berikut: ADM,,,,,,,,,APNS, APNS_SANDBOX APNS_VOIPAPNS_VOIP_SANDBOX, BAIDU atau. `ChannelType` GCM SMS Amazon Pinpoint tidak menghormati atribut ini untuk titik akhir di mana nilai atributnya berada. `ChannelType` EMAIL

Menetapkan kebijakan fungsi Lambda

Sebelum Anda dapat menggunakan fungsi Lambda untuk memproses titik akhir Anda, Anda harus mengotorisasi Amazon Pinpoint untuk menjalankan fungsi Lambda Anda. Untuk memberikan izin pemanggilan, tetapkan kebijakan fungsi Lambda ke fungsi tersebut. Kebijakan fungsi Lambda adalah kebijakan izin berbasis sumber daya yang menetapkan entitas mana yang dapat menggunakan fungsi Anda dan tindakan apa yang dapat dilakukan entitas tersebut.

Untuk informasi selengkapnya, lihat [Menggunakan kebijakan berbasis sumber daya AWS Lambda](#) di Panduan Pengembang. AWS Lambda

Contoh kebijakan fungsi

Kebijakan berikut memberikan izin kepada prinsipal layanan Amazon Pinpoint untuk menggunakan tindakan `lambda:InvokeFunction` tersebut untuk kampanye tertentu (`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"
    }
  }
}
```

Kebijakan fungsi Anda memerlukan Condition blok yang menyertakan AWS:SourceArn kunci. Kode ini menyatakan kampanye Amazon Pinpoint mana yang diizinkan untuk menjalankan fungsi tersebut. Dalam contoh ini, kebijakan hanya memberikan izin untuk satu kampanye. ConditionBlok juga harus menyertakan AWS:SourceAccount kunci, yang mengontrol AWS akun mana yang dapat menjalankan tindakan.

Untuk menulis kebijakan yang lebih umum, gunakan wildcard pencocokan multikarakter (*). Misalnya, Anda dapat menggunakan Condition blok berikut untuk mengizinkan kampanye apa pun dalam proyek Amazon Pinpoint tertentu (*application-id*) untuk menjalankan fungsi:

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

Jika Anda ingin fungsi Lambda menjadi fungsi default yang digunakan oleh semua kampanye untuk proyek, sebaiknya Anda mengonfigurasi Condition blok untuk kebijakan tersebut dengan cara sebelumnya. Untuk informasi tentang menyetel fungsi Lambda sebagai default untuk semua kampanye dalam proyek, lihat. [Menetapkan fungsi Lambda ke kampanye](#)

Memberikan izin pemanggilan Amazon Pinpoint

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk menambahkan izin ke kebijakan fungsi Lambda yang ditetapkan ke fungsi Lambda Anda. Untuk mengizinkan Amazon Pinpoint memanggil fungsi untuk kampanye tertentu, [add-permission](#) gunakan perintah Lambda, seperti yang ditunjukkan pada contoh berikut:

```
$ 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
```

Anda dapat mencari ID kampanye Anda dengan menggunakan perintah [get-campaign](#) di AWS CLI. Anda juga dapat mencari ID aplikasi Anda dengan menggunakan perintah [get-apps](#).

Ketika Anda menjalankan `add-permission` perintah Lambda, Lambda mengembalikan output berikut:

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

```
}
```

Nilai `Statement` adalah versi string JSON dari pernyataan yang ditambahkan ke kebijakan fungsi Lambda.

Menetapkan fungsi Lambda ke kampanye

Anda dapat menetapkan fungsi Lambda ke kampanye Amazon Pinpoint individual. Atau, Anda dapat mengatur fungsi Lambda sebagai default yang digunakan oleh semua kampanye untuk proyek, kecuali untuk kampanye yang Anda tetapkan fungsi satu per satu.

Untuk menetapkan fungsi Lambda ke kampanye individual, gunakan Amazon Pinpoint API untuk membuat atau memperbarui objek, dan menentukan [Campaign](#) atributnya. `CampaignHook` Untuk menetapkan fungsi Lambda sebagai default untuk semua kampanye dalam proyek, buat atau perbarui [Settings](#) sumber daya untuk proyek tersebut, dan tentukan objeknya `CampaignHook`.

Dalam kedua kasus, atur `CampaignHook` atribut berikut:

- `LambdaFunctionName`— Nama atau ARN dari fungsi Lambda yang dipanggil Amazon Pinpoint sebelum mengirim pesan untuk kampanye.
- `Mode` – Atur ke `FILTER`. Dengan mode ini, Amazon Pinpoint memanggil fungsi dan menunggu untuk mengembalikan titik akhir yang dimodifikasi. Setelah menerimanya, Amazon Pinpoint mengirim pesan. Amazon Pinpoint menunggu hingga 15 detik sebelum gagal pengiriman pesan.

Dengan `CampaignHook` pengaturan yang ditentukan untuk kampanye, Amazon Pinpoint memanggil fungsi Lambda yang ditentukan sebelum mengirim pesan kampanye. Amazon Pinpoint menunggu untuk menerima titik akhir yang dimodifikasi dari fungsi tersebut. Jika Amazon Pinpoint menerima titik akhir yang diperbarui, maka akan dilanjutkan dengan pengiriman pesan, menggunakan data titik akhir yang diperbarui.

Membuat kampanye

Untuk membantu meningkatkan interaksi antara aplikasi Anda dan penggunanya, gunakan Amazon Pinpoint untuk membuat dan mengelola kampanye pemberitahuan push yang menjangkau segmen pengguna tertentu.

Misalnya, kampanye Anda mungkin mengundang pengguna kembali ke aplikasi Anda yang belum menjalankannya baru-baru ini atau menawarkan promosi khusus kepada pengguna yang belum membeli baru-baru ini.

Kampanye mengirimkan pesan yang disesuaikan ke segmen pengguna yang Anda tentukan. Kampanye dapat mengirim pesan ke semua pengguna di segmen, atau Anda dapat mengalokasikan penahanan, yang merupakan persentase pengguna yang tidak menerima pesan.

Anda dapat mengatur jadwal kampanye untuk mengirim pesan sekali atau pada frekuensi berulang, seperti seminggu sekali. Untuk mencegah pengguna menerima pesan pada waktu yang tidak nyaman, jadwal dapat mencakup waktu tenang di mana tidak ada pesan yang dikirim.

Untuk bereksperimen dengan strategi kampanye alternatif, siapkan kampanye Anda sebagai tes A/B. Tes A/B mencakup dua atau lebih perawatan pesan atau jadwal. Perawatan adalah variasi dari pesan atau jadwal Anda. Saat pengguna merespons kampanye, Anda dapat melihat analitik kampanye untuk membandingkan efektivitas setiap perlakuan.

Untuk informasi selengkapnya, lihat [Kampanye](#).

Membuat kampanye standar

Kampanye standar mengirimkan pemberitahuan push kustom ke segmen tertentu sesuai dengan jadwal yang Anda tentukan.

Membuat kampanye dengan AWS SDK for Java

Contoh berikut menunjukkan cara membuat kampanye dengan AWS SDK for Java

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
```

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

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

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

        final String usage = ""

            Usage:    <appId> <segmentId>

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

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

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

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

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

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

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

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

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

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

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

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

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

    return null;
}
}
```

Saat Anda menjalankan contoh ini, berikut ini dicetak ke jendela konsol IDE Anda:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Untuk contoh SDK lengkap, lihat [CreateCampaign.java](#) on [GitHub](#)

Membuat kampanye uji A/B

Tes A/B berperilaku seperti kampanye standar, tetapi memungkinkan Anda menentukan perlakuan yang berbeda untuk pesan atau jadwal kampanye.

Membuat kampanye pengujian A/B dengan AWS SDK for Java

Contoh berikut menunjukkan cara membuat kampanye uji A/B dengan AWS SDK for Java

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

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

public class PinpointCampaignSample {

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

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

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

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

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

        WriteCampaignRequest request = new WriteCampaignRequest()
            .withDescription("My description.")
            .withSchedule(schedule)
            .withSegmentId(segmentId)
            .withName("MyCampaign")
            .withMessageConfiguration(messageConfiguration)
            .withAdditionalTreatments(additionalTreatments)
            .withHoldoutPercent(10); // Hold out of A/B test

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

```
        CreateCampaignResult result = client.createCampaign(createCampaignRequest);  
  
        System.out.println("Campaign ID: " + result.getCampaignResponse().getId());  
  
        return result.getCampaignResponse();  
    }  
}
```

Saat Anda menjalankan contoh ini, berikut ini dicetak ke jendela konsol IDE Anda:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Menggunakan Amazon Pinpoint SMS dan Voice API, versi 2

Note

Amazon Pinpoint telah memperbarui dokumentasi panduan pengguna mereka. Untuk mendapatkan informasi terbaru mengenai cara membuat, mengonfigurasi, dan mengelola sumber daya SMS dan suara Amazon Pinpoint Anda, lihat Panduan Pengguna SMS [Amazon Pinpoint](#) yang baru.

Topik berikut telah dipindahkan ke [Panduan Pengguna SMS Amazon Pinpoint](#) yang baru.

- [Mengelola nomor telepon](#)
- [Mengelola ID pengirim](#)
- [Mengelola kolam](#)
- [Mengelola daftar opt-out](#)
- [Mengelola set konfigurasi](#)
- [Mengelola kata kunci](#)
- [Mengelola tujuan acara](#)
- [Mengirim pesan](#)


Amazon Pinpoint menyertakan API (disebut SMS dan Voice API, versi 2) yang dirancang untuk mengirim SMS dan pesan suara. Sementara Amazon Pinpoint API berfokus pada pengiriman pesan melalui kampanye dan perjalanan yang dijadwalkan dan digerakkan oleh peristiwa, SMS dan Voice API menyediakan fitur dan kemampuan baru untuk mengirim SMS dan pesan suara langsung ke penerima individu. Anda dapat menggunakan SMS dan Voice API secara independen dari kampanye Amazon Pinpoint dan fitur perjalanan, atau Anda dapat menggunakan keduanya secara bersamaan untuk mengakomodasi kasus penggunaan yang berbeda. Jika Anda sudah menggunakan Amazon Pinpoint untuk mengirim SMS atau pesan suara, akun Anda sudah dikonfigurasi untuk menggunakan API ini.

API ini adalah solusi yang baik untuk pengguna yang memiliki arsitektur multi-tenant, seperti Independent Software Vendors (ISV). API ini memudahkan untuk memastikan bahwa data peristiwa, nomor telepon originasi, dan daftar opt-out dipisahkan untuk penyewa yang berbeda.

Saat Anda menggunakan SMS dan Voice API, kami sarankan Anda menyiapkan set konfigurasi dan tujuan acara. SMS dan Voice API tidak secara otomatis memancarkan data peristiwa untuk pesan

yang Anda kirim. Menyiapkan tujuan acara memastikan bahwa Anda menangkap data peristiwa penting, seperti pengiriman pesan dan peristiwa kegagalan.

Versi 2 dari API ini didahului oleh Versi 1. Jika saat ini Anda menggunakan Versi 1 dari API ini, itu akan terus tersedia, dan Anda dapat terus menggunakannya. Namun, jika Anda bermigrasi ke Versi 2, Anda akan mendapatkan fitur tambahan, seperti kemampuan untuk membuat kumpulan nomor telepon, meminta nomor telepon baru secara terprogram, dan mengaktifkan atau menonaktifkan kemampuan nomor telepon tertentu.

 Note

Ada beberapa tugas yang saat ini hanya dapat diselesaikan dengan menggunakan konsol Amazon Pinpoint. Misalnya, jika Anda ingin [memverifikasi nomor telepon untuk digunakan saat akun Anda berada di kotak pasir SMS](#), atau jika Anda ingin [mendaftar untuk menggunakan 10DLC](#), Anda harus menggunakan konsol Amazon Pinpoint.

Bagian ini memberikan informasi tentang API ini, dan menyertakan contoh cara menggunakannya. Anda juga dapat menemukan dokumentasi referensi di [SMS dan Suara, Referensi API versi 2](#).

Mengirim dan memvalidasi Kata Sandi Satu Kali (OTP) dengan Amazon Pinpoint

Amazon Pinpoint menyertakan fitur manajemen One-Time Password (OTP). Anda dapat menggunakan fitur ini untuk menghasilkan kata sandi satu kali baru dan mengirimkannya ke penerima sebagai pesan SMS. Aplikasi Anda kemudian dapat memanggil Amazon Pinpoint API untuk memverifikasi kata sandi ini.

Important

Untuk menggunakan fitur ini, akun Anda harus memiliki akses produksi dan identitas originasi aktif. Untuk informasi selengkapnya, lihat [Tentang kotak pasir SMS Amazon Pinpoint](#) dan [Minta nomor telepon di Panduan Pengguna SMS Amazon Pinpoint](#).

Di beberapa negara dan wilayah, Anda harus mendapatkan nomor telepon khusus atau ID originasi sebelum Anda dapat mengirim pesan SMS. Misalnya, ketika Anda mengirim pesan ke penerima di Amerika Serikat, Anda harus memiliki nomor bebas pulsa khusus, nomor 10DLC, atau kode pendek. Ketika Anda mengirim pesan ke penerima di India, Anda harus memiliki ID pengirim terdaftar, yang mencakup ID Entitas Utama (PEID) dan ID Templat. Persyaratan ini masih berlaku saat Anda menggunakan fitur OTP.

Untuk menggunakan fitur ini, Anda memerlukan izin untuk mengirim dan memverifikasi pesan OTP, lihat [Kata sandi satu kali](#). Jika Anda memerlukan bantuan untuk menentukan izin, lihat [Memecahkan masalah Amazon Pinpoint identitas dan manajemen akses](#).

Mengirim pesan OTP

Anda dapat menggunakan `SendOtpMessages` operasi di Amazon Pinpoint API untuk mengirim kode OTP ke pengguna aplikasi Anda. Saat Anda menggunakan API ini, Amazon Pinpoint menghasilkan kode acak dan mengirimkannya ke pengguna Anda sebagai pesan SMS. Permintaan Anda dapat mencakup parameter berikut:

- `Channel`— Saluran komunikasi yang dikirim melalui kode OTP. Saat ini, hanya pesan SMS yang didukung, jadi satu-satunya nilai yang dapat diterima adalah SMS.
- `BrandName`— Nama merek, perusahaan, atau produk yang terkait dengan kode OTP. Nama ini dapat berisi hingga 20 karakter.

Note

Saat Amazon Pinpoint mengirim pesan OTP, nama merek secara otomatis dimasukkan ke dalam templat pesan berikut:


```
This is your One Time Password: {{otp}} from {{brand}}
```

Jadi, jika Anda menentukan ExampleCorp sebagai nama merek Anda, dan Amazon Pinpoint menghasilkan kata sandi satu kali 123456, itu mengirimkan pesan berikut ke pengguna Anda:

```
This is your One Time Password: 123456 from ExampleCorp
```

- **CodeLength**— Jumlah digit yang akan ada dalam kode OTP yang dikirim ke penerima. Kode OTP dapat berisi antara 5 dan 8 digit, inklusif.
- **ValidityPeriod**— Jumlah waktu, dalam hitungan menit, bahwa kode OTP akan valid. Masa berlaku bisa antara 5 dan 60 menit, inklusif.
- **AllowedAttempts**— Berapa kali penerima tidak berhasil mencoba memverifikasi OTP. Jika jumlah upaya melebihi nilai ini, OTP secara otomatis menjadi tidak valid. Jumlah maksimum upaya yang diizinkan adalah 5.
- **Language**— Bahasa, dalam format IETF BCP-47, untuk digunakan saat mengirim pesan. Nilai yang dapat diterima adalah:
 - **de-DE**— Jerman
 - **en-GB**— Inggris (UK)
 - **en-US**— Inggris (AS)
 - **es-419**— Spanyol (Amerika Latin)
 - **es-ES**— Spanyol
 - **fr-CA**— Prancis (Kanada)
 - **fr-FR**— Prancis
 - **it-IT**— Italia
 - **ja-JP**— Jepang
 - **ko-KR**— Korea
 - **pt-BR**— Portugis (Brasil)

- zh-CN— Bahasa Mandarin (Sederhana)
- zh-TW— Tionghoa (Tradisional)
- `OriginationIdentity`— Identitas asal (seperti kode panjang, kode pendek, atau ID pengirim) yang digunakan untuk mengirim kode OTP. Jika Anda menggunakan kode panjang atau nomor bebas pulsa untuk mengirim OTP, nomor telepon harus dalam format E.164.
- `DestinationIdentity`— Nomor telepon, dalam format E.164, tempat kode OTP dikirim.
- `ReferenceId`— ID referensi unik untuk permintaan tersebut. ID referensi sama persis dengan ID referensi yang Anda berikan saat Anda memverifikasi OTP. ID referensi dapat berisi antara 1 dan 48 karakter, inklusif.
- `EntityId`— ID Entitas yang terdaftar di badan pengatur. Parameter ini saat ini hanya digunakan saat mengirim pesan ke penerima di India. Jika Anda tidak mengirim ke penerima di India, Anda dapat menghilangkan parameter ini.
- `TemplateId`— ID Template yang terdaftar di badan pengatur. Parameter ini saat ini hanya digunakan saat mengirim pesan ke penerima di India. Jika Anda tidak mengirim ke penerima di India, Anda dapat menghilangkan parameter ini.

 Note

Untuk informasi selengkapnya tentang persyaratan pengiriman pesan ke penerima di India, lihat [Persyaratan khusus untuk mengirim pesan SMS ke penerima di India di Panduan Pengguna Amazon Pinpoint](#).

Untuk memastikan bahwa akun Amazon Pinpoint Anda dikonfigurasi dengan benar untuk mengirim pesan OTP, Anda dapat menggunakan AWS CLI untuk mengirim pesan pengujian. Untuk informasi selengkapnya tentang AWS CLI, lihat [Panduan AWS Command Line Interface Pengguna](#).

Untuk mengirim pesan OTP pengujian menggunakan AWS CLI, jalankan [send-otp-message](#) perintah di 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
```

Pada perintah sebelumnya, lakukan hal berikut:

- Ganti *7353f53e6885409fa32d07cedexample* dengan id aplikasi Anda.

- Ganti *ExampleCorp* dengan nama perusahaan Anda.
- Ganti *5* in `CodeLength` dengan jumlah digit yang akan ada dalam kode OTP yang dikirim ke penerima.
- Ganti *20* in `ValidityPeriod` dengan jumlah waktu, dalam menit, bahwa kode OTP akan valid.
- Ganti *5* in `AllowedAttempts` dengan berapa kali penerima tidak berhasil mencoba memverifikasi OTP.
- Ganti *+18555550142* `OriginationIdentity` dengan identitas asal yang digunakan untuk mengirim kode OTP.
- Ganti *+12065550007* in `DestinationIdentity` dengan nomor telepon untuk mengirim kode OTP ke.
- Ganti *SampleReferenceId* `ReferenceId` dengan ID referensi unik untuk permintaan tersebut.

SendOtpMessage Respons

Ketika Anda berhasil mengirim pesan OTP, Anda menerima respons yang menyerupai contoh berikut:

```
{
  "MessageResponse": {
    "ApplicationId": "7353f53e6885409fa32d07cedexample",
    "RequestId": "255d15ea-75fe-4040-b919-096f2example",
    "Result": {
      "+12065550007": {
        "DeliveryStatus": "SUCCESSFUL",
        "MessageId": "nvrimgq9kq4en96qgp0tlqli3og1at6aexample",
        "StatusCode": 200,
        "StatusMessage": "MessageId: nvrimgq9kq4en96qgp0tlqli3og1at6aexample"
      }
    }
  }
}
```

Memvalidasi pesan OTP

Untuk memverifikasi kode OTP, panggil `VerifyOtpMessages` API. Permintaan Anda harus menyertakan parameter berikut:

- **DestinationIdentity**— Nomor telepon, dalam format E.164, tempat kode OTP dikirim.
- **ReferenceId**— ID referensi yang Anda gunakan saat Anda mengirim kode OTP ke penerima. ID referensi harus sama persis.
- **Otp**— Kode OTP yang Anda validasi.

Anda dapat menggunakan AWS CLI untuk menguji proses validasi. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat [Panduan AWS Command Line Interface Pengguna](#).

Untuk memverifikasi OTP menggunakan AWS CLI, jalankan [verify-otp-message](#) perintah di terminal:

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

Pada perintah sebelumnya, lakukan hal berikut:

- Ganti `7353f53e6885409fa32d07cedexample` dengan id aplikasi Anda.
- Ganti `+12065550007` **DestinationIdentity** dengan nomor telepon tempat kode OTP dikirim.
- Ganti `SampleReferenceId` **ReferenceId** dengan ID referensi unik untuk permintaan tersebut. Nilai ini harus sesuai dengan **ReferenceID** yang digunakan untuk mengirim permintaan.
- Ganti `01234` in **Otp** dengan OTP yang dikirim ke. **DestinationIdentity**

VerifyOtpMessage Respons

Ketika Anda mengirim permintaan ke **VerifyOTPMessage** API, ia mengembalikan **VerificationResponse** objek, yang berisi properti tunggal, **Valid**. Jika ID referensi, nomor telepon, dan OTP semuanya cocok dengan nilai yang diharapkan Amazon Pinpoint, dan jika OTP belum kedaluwarsa, nilainya `true` adalah; jika tidak, itu **Valid** benar. `false` Berikut ini adalah contoh tanggapan untuk verifikasi OTP yang berhasil:

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

Contoh kode

Bagian ini berisi contoh kode yang menunjukkan cara menggunakan SDK for Python (Boto3) untuk mengirim dan memverifikasi kode OTP.

Menghasilkan ID referensi

Fungsi berikut menghasilkan ID referensi unik untuk setiap penerima, berdasarkan nomor telepon penerima, produk atau merek tempat penerima menerima OTP, dan sumber permintaan (yang bisa berupa nama halaman di situs atau aplikasi, misalnya). Ketika Anda memverifikasi kode OTP, Anda harus melewati ID referensi yang identik agar validasi berhasil. Contoh kode pengiriman dan validasi menggunakan fungsi utilitas ini.

Fungsi ini tidak diperlukan, tetapi ini adalah cara yang berguna untuk mencakup proses pengiriman dan verifikasi OTP ke transaksi tertentu dengan cara yang dapat dengan mudah dikirimkan kembali selama langkah verifikasi. Anda dapat menggunakan ID referensi apa pun yang Anda inginkan—ini hanyalah contoh dasar. Namun, contoh kode lain di bagian ini bergantung pada fungsi ini.

```
# 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()
```

Mengirim kode OTP

Contoh kode berikut menunjukkan cara menggunakan SDK for Python (Boto3) untuk mengirim kode 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")
```

Memvalidasi kode OTP

Contoh kode berikut menunjukkan cara menggunakan SDK for Python (Boto3) untuk memverifikasi kode OTP yang sudah Anda kirim. Agar langkah validasi berhasil, permintaan Anda harus menyertakan ID referensi yang sama persis dengan ID referensi yang digunakan untuk mengirim pesan.

```
# 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")
```

Mengirim dan mengambil pesan dalam aplikasi di Amazon Pinpoint

Anda dapat menggunakan pesan dalam aplikasi untuk mengirim pesan yang ditargetkan ke pengguna aplikasi Anda. Pesan dalam aplikasi sangat dapat disesuaikan. Mereka dapat menyertakan tombol yang membuka situs web atau membawa pengguna ke bagian tertentu dari aplikasi Anda. Anda dapat mengonfigurasi warna latar belakang dan teks, memposisikan teks, dan menambahkan tombol dan gambar ke notifikasi. Anda dapat mengirim satu pesan, atau membuat korsel yang berisi hingga lima pesan unik. Untuk ringkasan pesan dalam aplikasi, termasuk petunjuk untuk membuat template pesan dalam aplikasi, lihat [Membuat template dalam aplikasi](#) di Panduan Pengguna Amazon Pinpoint.

Anda dapat menggunakannya AWS Amplify untuk mengintegrasikan kemampuan perpesanan dalam aplikasi Amazon Pinpoint dengan mulus ke dalam aplikasi Anda. Amplify dapat secara otomatis menangani proses pengambilan pesan, merender pesan, dan mengirim data analitik ke Amazon Pinpoint. Integrasi ini saat ini didukung untuk aplikasi React Native. Untuk informasi selengkapnya, lihat [Pesan Dalam Aplikasi](#) di Dokumentasi Kerangka Amplify.

Bagian ini menyediakan informasi tentang meminta pesan dalam aplikasi untuk titik akhir di aplikasi Anda, dan untuk menafsirkan hasil permintaan tersebut.

Mengambil pesan dalam aplikasi untuk titik akhir

Aplikasi Anda dapat memanggil [GetInAppMessages](#) API untuk mengambil semua pesan dalam aplikasi yang menjadi hak endpoint tertentu. Saat Anda memanggil `GetInAppMessages` API, Anda memberikan parameter berikut:

- `ApplicationId`— ID unik proyek Amazon Pinpoint yang dikaitkan dengan kampanye pesan dalam aplikasi.
- `EndpointId`- ID unik dari titik akhir tempat Anda mengambil pesan.

Ketika Anda memanggil API dengan nilai-nilai ini, ia mengembalikan daftar pesan. Untuk informasi lebih lanjut tentang respon yang dihasilkan oleh operasi ini, lihat [Memahami respons `GetInAppMessages` API](#).

Anda dapat menggunakan AWS SDK untuk memanggil `GetInAppMessages` operasi. Contoh kode berikut mencakup fungsi yang mengambil pesan dalam aplikasi.

JavaScript

Buat klien dalam modul terpisah dan ekspor:

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

Mengambil pesan dalam aplikasi untuk titik akhir:

```
// 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()
```

Memahami respons `GetInAppMessages` API

Ketika Anda memanggil operasi [GetInAppMessages](#) API, ia mengembalikan daftar pesan yang endpoint ditentukan berhak. Aplikasi Anda kemudian dapat merender pesan berdasarkan nilai dalam respons.

Berikut ini adalah contoh objek JSON yang dikembalikan saat Anda memanggil `GetInAppMessages` API:

```
{
  "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"
  }
]
}
}

```

Bagian berikut menyediakan informasi lebih lanjut tentang komponen respons ini.

Objek **InAppMessageCampaigns**

InAppMessageCampaignsObjek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
CampaignId	String yang berisi nama dan ID kampanye unik kampanye Amazon Pinpoint tempat pesan dikirim. Nama mendahului ID kampanye.	Dibuat secara otomatis oleh Amazon Pinpoint saat Anda membuat kampanye.

Atribut	Deskripsi	PengGUNANNYA
	Kedua nilai dipisahkan dengan tanda hubung (-).	
TreatmentId	Integer yang mewakili ID perlakuan kampanye untuk pesan ini. Jika kampanye hanya memiliki satu perlakuan, nilainya adalah 0.	
Priority	Prioritas pesan dalam aplikasi, dinyatakan sebagai integer antara 1 dan 5, inklusif, di mana 1 menunjukkan prioritas tertinggi, dan 5 menunjukkan prioritas terendah.	Langkah 1 dari proses pembuatan kampanye.
InAppMessage	Sebuah Objek InAppMessage yang berisi informasi tentang bagaimana pesan diberikan.	Berdasarkan konten dalam template pesan dalam aplikasi yang ditentukan untuk kampanye.
Schedule	Objek Jadwal yang berisi informasi tentang kapan pesan dikirim.	Langkah 4 dari proses pembuatan kampanye (jika kampanye dibuat di konsol) atau <code>Schedule</code> objek (jika kampanye dibuat menggunakan API atau SDK).
DailyCap	Berapa kali, ditampilkan sebagai integer, bahwa pesan dalam aplikasi dapat ditampilkan kepada pengguna selama periode 24 jam.	Diwarisi dari pengaturan tingkat proyek . Jika kampanye menyertakan setelan yang menimpa setelan proyek, maka kampanye tersebut akan digunakan sebagai gantinya.

Atribut	Deskripsi	PengGUNANNYA
SessionCap	Berapa kali, dinyatakan sebagai integer, bahwa pesan dalam aplikasi dapat ditampilkan kepada pengguna selama sesi aplikasi.	
TotalCap	Jumlah total waktu, dinyatakan sebagai integer, bahwa setiap pesan dalam aplikasi dapat ditampilkan ke titik akhir per kampanye.	

Objek InAppMessage

InAppMessageObjek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
Content	Array yang berisi InAppMessageContent objek, yang menggambarkan isi pesan.	Berdasarkan konten dalam template pesan dalam aplikasi yang ditentukan untuk kampanye.
Layout	String yang menjelaskan bagaimana pesan dalam aplikasi akan muncul di perangkat penerima. Nilai yang mungkin adalah: <ul style="list-style-type: none"> BOTTOM_BANNER - pesan yang muncul sebagai spanduk di bagian bawah halaman. TOP_BANNER - pesan yang muncul sebagai spanduk di bagian atas halaman. 	

Atribut	Deskripsi	PengGUNANNYA
	<ul style="list-style-type: none"> • OVERLAYS- pesan yang mencakup seluruh layar. • MOBILE_FEED — pesan yang muncul di jendela di depan halaman. • MIDDLE_BANNER - pesan yang muncul sebagai spanduk di tengah halaman. • CAROUSEL- Tata letak yang dapat digulir hingga lima pesan unik. 	

Objek HeaderConfig

HeaderConfigObjek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
Alignment	Sebuah string yang menentukan keselarasan teks teks header. Nilai yang mungkin adalah LEFT, CENTER, dan RIGHT.	Berdasarkan konten dalam template pesan dalam aplikasi yang ditentukan untuk kampanye.
Header	Teks header pesan.	
TextColor	Warna teks header, dinyatakan sebagai string yang menggambarkan kode warna hex (seperti "#000000" untuk hitam).	

Objek **BodyConfig**

BodyConfigObjek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
Alignment	String yang menentukan keselarasan teks dari badan pesan. Nilai yang mungkin adalah LEFT, CENTER, dan RIGHT.	Berdasarkan konten dalam template pesan dalam aplikasi yang ditentukan untuk kampanye.
Body	Teks badan utama pesan.	
TextColor	Warna teks tubuh, dinyatakan sebagai string yang berisi kode warna hex (seperti "#000000" untuk hitam).	

Objek **InAppMessageContent**

InAppMessageContentObjek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
BackgroundColor	Warna latar belakang pesan dalam aplikasi, dinyatakan sebagai string yang berisi kode warna hex (seperti "#000000" untuk hitam).	Berdasarkan konten dalam template pesan dalam aplikasi yang ditentukan untuk kampanye.
BodyConfig	Sebuah BodyConfig objek, yang berisi informasi yang terkait dengan isi tubuh utama pesan.	
HeaderConfig	Sebuah HeaderConfig objek, yang berisi informasi yang	

Atribut	Deskripsi	PengGUNANNYA
	terkait dengan header atau judul pesan.	
ImageUrl	URL gambar yang muncul dalam pesan.	
PrimaryBtn	InAppMessageButton Objek yang berisi informasi tentang tombol utama dalam pesan.	
SecondaryBtn	InAppMessageButton Objek yang berisi informasi tentang tombol sekunder dalam pesan. Tidak ada jika template pesan dalam aplikasi tidak menentukan tombol sekunder.	

Objek **Schedule**

ScheduleObjek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
EndDate	Waktu yang dijadwalkan, dalam format ISO 8601, kapan kampanye akan berakhir.	Langkah 4 dari proses pembuatan kampanye (jika kampanye dibuat di konsol) atauSchedule objek (jika kampanye dibuat menggunakan API atau SDK).
EventFilter	Informasi tentang peristiwa yang menyebabkan pesan dalam aplikasi ditampilkan. Saat Anda membuat acara yang cocok dengan kampanye dalam aplikasi Amazon Pinpoint, pesan akan ditampilkan.	

Objek InAppMessageButton

InAppMessageButton objek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
DefaultConfig	DefaultButtonConfig objek yang berisi informasi tentang pengaturan default untuk tombol dalam pesan dalam aplikasi.	Berdasarkan konten dalam template pesan dalam aplikasi yang ditentukan untuk kampanye.
Android	OverrideButtonConfig objek yang menentukan cara tombol berperilaku pada perangkat Android. Ini menerima konfigurasi tombol default rinci dalam DefaultConfig objek.	
IOS	OverrideButtonConfig objek yang menentukan cara tombol berperilaku pada perangkat iOS. Ini menerima konfigurasi tombol default rinci dalam DefaultConfig objek.	
Web	OverrideButtonConfig objek yang menentukan cara tombol berperilaku dalam aplikasi web. Ini menerima konfigurasi tombol default rinci dalam DefaultConfig objek.	

Objek `DefaultButtonConfig`

`DefaultButtonConfig` objek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
<code>BackgroundColor</code>	Warna latar belakang tombol, dinyatakan sebagai string yang berisi kode warna hex (seperti "#000000" untuk hitam).	Berdasarkan konten dalam template pesan dalam aplikasi yang ditentukan untuk kampanye.
<code>BorderRadius</code>	Jari-jari batas tombol dalam piksel, dinyatakan sebagai bilangan bulat. Jumlah yang lebih besar menghasilkan sudut yang lebih bulat.	
<code>ButtonAction</code>	String yang menjelaskan tindakan yang terjadi saat penerima memilih tombol dalam pesan dalam aplikasi. Nilai yang mungkin adalah: <ul style="list-style-type: none">• <code>LINK</code>— Sebuah link ke tujuan web.• <code>DEEP_LINK</code> — Sebuah link ke halaman tertentu dalam aplikasi.• <code>CLOSE</code>- Menolak pesan.	
<code>Link</code>	URL tujuan untuk tombol. Tidak hadir untuk tombol di mana <code>ButtonAction</code> adalah <code>CLOSE</code> .	
<code>Text</code>	Teks yang muncul pada tombol.	

Atribut	Deskripsi	PengGUNANNYA
TextColor	Warna teks pada tombol, dinyatakan sebagai string yang berisi kode warna hex (seperti "#000000" untuk hitam).	

Objek **OverrideButtonConfig**

OverrideButtonConfigObjek hanya ada jika template pesan dalam aplikasi menggunakan tombol ganti. Tombol override adalah tombol yang memiliki konfigurasi khusus untuk jenis perangkat tertentu, seperti perangkat iOS, perangkat Android, atau browser web.

OverrideButtonConfigObjek berisi atribut berikut:

Atribut	Deskripsi	PengGUNANNYA
ButtonAction	Tindakan yang terjadi saat penerima memilih tombol dalam pesan dalam aplikasi. Nilai yang mungkin adalah: <ul style="list-style-type: none"> LINK— Sebuah link ke tujuan web. DEEP_LINK — Sebuah link ke halaman tertentu dalam aplikasi. CLOSE- Menolak pesan. 	Berdasarkan konten dalam template pesan dalam aplikasi yang ditentukan untuk kampanye.
Link	URL tujuan untuk tombol. Tidak hadir untuk tombol di manaButtonAction adalahCLOSE.	
Text	Teks yang muncul pada tombol.	

Atribut	Deskripsi	PengGUNANNYA
TextColor	Warna teks pada tombol, dinyatakan sebagai string yang berisi kode warna hex (seperti "#000000" untuk hitam).	

Memvalidasi nomor telepon di Amazon Pinpoint

Amazon Pinpoint menyertakan layanan validasi nomor telepon yang dapat Anda gunakan untuk menentukan apakah nomor telepon valid, dan untuk mendapatkan informasi tambahan tentang nomor telepon itu sendiri. Misalnya, ketika Anda menggunakan layanan validasi nomor telepon, ia mengembalikan informasi berikut:

- Nomor telepon dalam format E.164.
- Jenis nomor telepon (seperti ponsel, telepon rumah, atau VoIP).
- Kota dan negara tempat nomor telepon berada.
- Penyedia layanan yang terkait dengan nomor telepon.

Ada biaya tambahan untuk menggunakan layanan validasi nomor telepon. Untuk informasi selengkapnya, lihat [harga Amazon Pinpoint](#).

Important

Untuk originasi nomor telepon di Amerika Serikat dan Kanada, API yang memvalidasi nomor telepon tidak akan lagi mengembalikan data untuk City, County, Timezone dan ZipCode

Kasus penggunaan validasi nomor telepon

Anda dapat menggunakan layanan validasi nomor telepon untuk mengaktifkan beberapa kasus penggunaan, termasuk yang berikut:

- Memverifikasi nomor telepon yang disediakan pada formulir web — Jika Anda menggunakan formulir berbasis web untuk mengumpulkan informasi kontak bagi pelanggan Anda, Anda memvalidasi nomor telepon yang diberikan pelanggan sebelum mengirimkan formulir. Gunakan backend situs web Anda untuk memvalidasi nomor dengan menggunakan Amazon Pinpoint API. Respons API menyatakan apakah nomor tersebut tidak valid—misalnya, jika nomor telepon salah diformat. Jika Anda menentukan bahwa nomor telepon yang diberikan pelanggan tidak valid, formulir web Anda dapat meminta pelanggan untuk memberikan nomor yang berbeda.
- Membersihkan basis data kontak yang ada — Jika Anda memiliki database nomor telepon pelanggan, Anda dapat memvalidasi setiap nomor telepon, dan kemudian memperbarui database

Anda berdasarkan temuan Anda. Misalnya, jika Anda menemukan titik akhir dengan nomor telepon yang tidak mampu menerima pesan SMS, Anda dapat mengubah ChannelType properti untuk titik akhir dari SMS ke. VOICE Anda dapat memvalidasi nomor telepon terlebih dahulu dan kemudian memperbarui ChannelType properti untuk titik akhir baru atau yang sudah ada dengan mengikuti petunjuk [Menambahkan titik akhir ke Amazon Pinpoint](#) untuk satu titik akhir atau [Menambahkan sejumlah titik akhir ke Amazon Pinpoint](#) beberapa titik akhir.

- Memilih saluran yang tepat sebelum mengirim pesan — Jika Anda berniat mengirim pesan SMS tetapi Anda menentukan bahwa nomor tujuan tidak valid, Anda dapat mengirim pesan ke penerima melalui saluran yang berbeda. Misalnya, jika titik akhir tidak dapat menerima pesan SMS, Anda dapat mengirim pesan suara sebagai gantinya.

Menggunakan layanan validasi nomor telepon

Contoh berikut menunjukkan cara memvalidasi nomor telepon menggunakan AWS CLI Untuk informasi selengkapnya, lihat [phone-number-validated](#) di Referensi AWS CLI Perintah. Misalnya tanggapan validasi, lihat [Tanggapan validasi nomor telepon](#). Untuk informasi selengkapnya tentang mengonfigurasi AWS CLI, lihat [Mengkonfigurasi AWS CLI](#) di [Panduan AWS Command Line Interface Pengguna](#).

Untuk menggunakan layanan validasi nomor telepon dengan menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

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

Pada perintah sebelumnya, ganti **+442079460881** dengan nomor telepon yang ingin Anda validasi dan **GB** dengan kode negara atau wilayah ISO dua digit.

Note

Saat Anda memberikan nomor telepon ke layanan validasi nomor telepon, Anda harus selalu menyertakan kode negara. Jika Anda tidak menyertakan kode negara, layanan mungkin mengembalikan informasi untuk nomor telepon di negara lain. Anda dapat memiliki tanda hubung di nomor telepon, misalnya **+44-207-946-0881**.

Tanggapan validasi nomor telepon

Informasi yang disediakan oleh layanan validasi nomor telepon sedikit berbeda berdasarkan data yang tersedia untuk nomor telepon yang Anda berikan. Bagian ini berisi contoh tanggapan yang dikembalikan oleh layanan validasi nomor telepon.

Note

Data yang disediakan oleh layanan validasi nomor telepon didasarkan pada informasi yang diberikan oleh penyedia telekomunikasi dan entitas lain di seluruh dunia. Penyedia di beberapa negara mungkin memperbarui informasi ini lebih jarang daripada penyedia di negara lain. Misalnya, jika Anda mengeluarkan permintaan untuk memvalidasi nomor ponsel, dan nomor yang Anda berikan telah di-porting dari satu operator seluler ke operator seluler lainnya, respons dari layanan validasi nomor telepon mungkin menyertakan nama operator asli, sebagai lawan dari yang sekarang.

Nomor ponsel yang valid

Ketika Anda mengirim permintaan ke layanan validasi nomor telepon, dan nomor telepon adalah nomor ponsel yang valid, ia mengembalikan informasi yang menyerupai contoh berikut:

```
{
  "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"
  }
}
```

Nomor telepon rumah yang valid

Jika permintaan Anda berisi nomor telepon darat yang valid, layanan validasi nomor telepon mengembalikan informasi yang menyerupai contoh berikut:

```
{
  "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"
}
```

Nomor telepon VoIP yang valid

Jika permintaan Anda berisi nomor telepon Voice over Internet Protocol (VoIP) yang valid, layanan validasi nomor telepon akan menampilkan informasi yang menyerupai contoh berikut:

```
{
  "NumberValidateResponse": {
    "Carrier": "ExampleCorp",
    "City": "Countrywide",
    "CleansedPhoneNumberE164": "+441514960001",
    "CleansedPhoneNumberNational": "1514960001",
    "Country": "United Kingdom",
    "CountryCodeIso2": "GB",
    "CountryCodeNumeric": "44",
    "OriginalPhoneNumber": "+441514960001",
    "PhoneType": "VOIP",
    "PhoneTypeCode": 2
  }
}
```

Nomor telepon tidak valid

Jika permintaan Anda berisi nomor telepon yang tidak valid, layanan validasi nomor telepon akan menampilkan informasi yang menyerupai contoh berikut:

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

Perhatikan bahwa PhoneType properti dalam respons ini menunjukkan bahwa nomor telepon ini INVALID, dan tidak menyertakan informasi tentang operator atau lokasi yang terkait dengan nomor telepon. Anda harus menghindari mengirim SMS atau pesan suara ke nomor telepon di mana PhoneType berada INVALID, karena angka-angka ini tidak mungkin milik penerima yang sebenarnya.

Nomor telepon lainnya

Kadang-kadang, respons dari layanan validasi nomor telepon mencakup PhoneType nilai. OTHER Layanan mungkin mengembalikan respons semacam ini dalam situasi berikut:

- Nomor telepon adalah nomor bebas pulsa (freephone).
- Nomor telepon dicadangkan untuk digunakan dalam acara TV dan film, seperti nomor telepon Amerika Utara yang dimulai dengan 555.
- Nomor telepon termasuk kode area yang saat ini tidak digunakan, seperti kode area 999 di Amerika Utara.
- Nomor telepon dicadangkan untuk beberapa tujuan lain.

Contoh berikut menunjukkan respons yang diberikan oleh layanan validasi nomor telepon ketika permintaan Anda menyertakan nomor telepon fiktif Amerika Utara:

```
{
  "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"
}
}
```

Nomor telepon prabayar

Jika permintaan Anda berisi nomor telepon prabayar yang valid, layanan validasi nomor telepon mengembalikan informasi yang menyerupai contoh berikut:

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

Untuk informasi selengkapnya tentang informasi yang terkandung dalam tanggapan ini, lihat [Nomor telepon memvalidasi di Referensi](#) API Amazon Pinpoint.

Mengirim pesan transaksional dari aplikasi Anda

Anda dapat menggunakan API Amazon Pinpoint dan AWS SDK untuk mengirim pesan transaksional langsung dari aplikasi Anda. Pesan transaksional adalah pesan yang Anda kirim ke penerima tertentu, dibandingkan dengan pesan yang Anda kirim ke segmen. Ada beberapa alasan bahwa Anda mungkin ingin mengirim pesan transaksional daripada pesan berbasis kampanye. Misalnya, Anda dapat mengirim konfirmasi pemesanan melalui email saat pelanggan melakukan pemesanan. Anda juga dapat mengirim kata sandi satu kali melalui SMS atau suara yang dapat digunakan pelanggan untuk menyelesaikan proses pembuatan akun untuk layanan Anda.

Bagian ini mencakup contoh kode dalam beberapa bahasa pemrograman yang dapat Anda gunakan untuk mulai mengirim email transaksional, pesan SMS, dan pesan suara.

Topik di bagian ini:

- [Kirim pesan email transaksional](#)
- [Kirim pesan SMS](#)
- [Kirim pesan suara](#)
- [Kirim notifikasi push](#)

Kirim pesan email transaksional

Bagian ini menyediakan contoh kode lengkap yang dapat Anda gunakan untuk mengirim pesan email transaksional melalui Amazon Pinpoint:

- [Dengan menggunakan SendMessages operasi di Amazon Pinpoint API](#): Anda dapat menggunakan SendMessages operasi di Amazon Pinpoint API untuk mengirim pesan di semua saluran yang didukung Amazon Pinpoint, termasuk notifikasi push, SMS, suara, dan saluran email.

Keuntungan menggunakan operasi ini adalah bahwa sintaks permintaan untuk mengirim pesan sangat mirip di semua saluran. Ini membuatnya lebih mudah untuk menggunakan kembali kode yang ada. SendMessages Operasi ini juga memungkinkan Anda untuk mengganti konten dalam pesan email Anda, dan memungkinkan Anda mengirim email ke ID titik akhir Amazon Pinpoint daripada ke alamat email tertentu.

Bagian ini mencakup contoh kode dalam beberapa bahasa pemrograman yang dapat Anda gunakan untuk mulai mengirim email transaksional.

Topik di bagian ini:

- [Memilih metode pengiriman email](#)
- [Memilih antara Amazon Pinpoint dan Amazon Simple Email Service \(SES\)](#)
- [Kirim email dengan menggunakan Amazon Pinpoint API](#)
- [Mengirim email dengan header berhenti berlangganan](#)

Memilih metode pengiriman email

Metode terbaik yang digunakan untuk mengirim email transaksional tergantung pada kasus penggunaan Anda. Misalnya, jika Anda perlu mengirim email menggunakan aplikasi pihak ketiga, atau jika tidak ada AWS SDK yang tersedia untuk bahasa pemrograman Anda, Anda mungkin harus menggunakan antarmuka SMTP. Jika Anda ingin mengirim pesan di saluran lain yang didukung Amazon Pinpoint, dan Anda ingin menggunakan kode yang konsisten untuk membuat permintaan tersebut, Anda harus menggunakan SendMessages operasi di Amazon Pinpoint API.

Memilih antara Amazon Pinpoint dan Amazon Simple Email Service (SES)

Jika Anda mengirim sejumlah besar email transaksional, seperti konfirmasi pembelian atau pesan pengaturan ulang kata sandi, pertimbangkan untuk menggunakan Amazon SES. Amazon SES memiliki API dan antarmuka SMTP, yang keduanya cocok untuk mengirim email dari aplikasi atau layanan Anda. Ini juga menawarkan fitur email tambahan, termasuk kemampuan menerima email, set konfigurasi, dan kemampuan otorisasi pengiriman.

Amazon SES juga menyertakan antarmuka SMTP yang dapat Anda integrasikan dengan aplikasi pihak ketiga yang ada, termasuk layanan manajemen hubungan pelanggan (CRM) seperti Salesforce. Untuk informasi selengkapnya tentang mengirim email menggunakan Amazon SES, [Panduan Pengembang Layanan Email Sederhana Amazon](#) untuk informasi lebih lanjut.

Kirim email dengan menggunakan Amazon Pinpoint API

Bagian ini berisi contoh kode lengkap yang dapat Anda gunakan untuk mengirim email melalui Amazon Pinpoint API dengan menggunakan SDK. AWS

C#

Gunakan contoh ini untuk mengirim email dengan menggunakan file [AWS SDK for .NET](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi file. AWS SDK for .NET

Untuk informasi selengkapnya, lihat [Memulai AWS SDK for .NET](#) di Panduan AWS SDK for .NET Pengembang.

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Mengonfigurasi AWS kredensial di Panduan](#) Pengembang.AWS SDK for .NET

Contoh kode ini diuji menggunakan AWS SDK for .NET versi 3.3.29.13 dan runtime .NET Core versi 2.1.2.

```
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

Gunakan contoh ini untuk mengirim email dengan menggunakan file [AWS SDK for Java](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi file. AWS SDK for Java 2.x Untuk informasi lebih lanjut, lihat [Memulai](#) di Panduan Developer.AWS SDK for Java 2.x

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Menetapkan kredensi default dan Wilayah](#) di Panduan AWS SDK for Java Pengembang.

Contoh kode ini diuji menggunakan AWS SDK for Java versi 2.3.1 dan OpenJDK versi 11.0.1.

```

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);
    }
}
```

Untuk contoh SDK lengkap, lihat [SendEmailMessage.java](#) on. [GitHub](#)

JavaScript (Node.js)

Gunakan contoh ini untuk mengirim email menggunakan [AWS SDK for JavaScript di Node.js](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengonfigurasi SDK untuk JavaScript di Node.js. Untuk informasi selengkapnya, lihat [Memulai AWS](#) SDK untuk JavaScript di Panduan Pengembang Node.js.

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Menyetel kredensial](#) di AWS SDK untuk JavaScript di Panduan Pengembang Node.js.

Contoh kode ini diuji menggunakan SDK untuk JavaScript di Node.js versi 2.388.0 dan Node.js versi 11.7.0.

```
"use strict";

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

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

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

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

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

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)`
```

```
-----  
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in  
Node.js.  
For more information, see https://aws.amazon.com/sdk-for-node-js/;  
  
// The body of the email for recipients whose email clients support HTML content.  
var body_html = `  
<head></head>  
<body>  
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>  
  <p>This email was sent with  
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using the  
    <a href='https://aws.amazon.com/sdk-for-node-js/'>  
      AWS SDK for JavaScript in Node.js</a>.</p>  
</body>  
</html>`;  
  
// The character encoding the you want to use for the subject line and  
// message body of the email.  
var charset = "UTF-8";  
  
// Specify that you're using a shared credentials file.  
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });  
AWS.config.credentials = credentials;  
  
// Specify the region.  
AWS.config.update({ region: aws_region });  
  
//Create a new Pinpoint object.  
var pinpoint = new AWS.Pinpoint();  
  
// Specify the parameters to pass to the API.  
var params = {  
  ApplicationId: appId,  
  MessageRequest: {  
    Addresses: {  
      [toAddress]: {  
        ChannelType: "EMAIL",  
      },  
    },  
    MessageConfiguration: {  
      EmailMessage: {  
        FromAddress: senderAddress,  
        SimpleEmail: {
```



```
        Subject: {
            Charset: charset,
            Data: subject,
        },
        HtmlPart: {
            Charset: charset,
            Data: body_html,
        },
        TextPart: {
            Charset: charset,
            Data: body_text,
        },
    },
},
},
},
};

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

Python

Gunakan contoh ini untuk mengirim email dengan menggunakan file [AWS SDK for Python \(Boto3\)](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi SDK for Python (Boto3). Untuk informasi selengkapnya, lihat [Mulai Cepat](#) di Referensi AWS API SDK for 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()
```

Anda juga dapat menggunakan template pesan untuk mengirim pesan email, seperti yang ditunjukkan pada contoh berikut:

```
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()
```

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Kredensial di Referensi](#) API AWS SDK for Python (Boto3).

Mengirim email dengan header berhenti berlangganan

Note

Sebelum Anda dapat menggunakan header email, Anda harus mengatur peran pengiriman orkestrasi email jika Anda mengirim email dari kampanye atau perjalanan. Untuk mengirim email langsung, Anda harus memiliki izin untuk `ses:SendEmail` dan `ses:SendRawEmail`. Untuk informasi selengkapnya, lihat [Membuat peran pengiriman orkestrasi email di Panduan Pengguna Amazon Pinpoint](#).

Menyertakan tautan berhenti berlangganan di email Anda adalah praktik terbaik, dan di beberapa negara itu merupakan persyaratan hukum. Untuk menambahkan tautan berhenti berlangganan sekali klik, tambahkan header berikut:

1. Atur Nama header ke `List-Unsubscribe` dan atur Nilai ke tautan berhenti berlangganan Anda. Tautan harus mendukung permintaan HTTP POST untuk memproses permintaan berhenti berlangganan penerima.
2. Atur Nama header ke `List-Unsubscribe-Post` dan atur Nilai ke `List-Unsubscribe=One-Click`.

Anda dapat menambahkan hingga 15 header ke pesan email. Untuk daftar header yang didukung, lihat [bidang header Amazon SES](#) di [Panduan Pengembang Layanan Email Sederhana Amazon](#).

Contoh berikut menunjukkan cara mengirim pesan email dengan header berhenti berlangganan menggunakan AWS Command Line Interface Untuk informasi selengkapnya tentang mengonfigurasi AWS CLI, lihat [Mengkonfigurasi AWS CLI](#) dalam [Panduan AWS Command Line Interface Pengguna](#).

Dalam perintah berikut, lakukan hal berikut:

- Ganti *AppId* dengan id aplikasi Anda.
- Ganti *richard_roe@example.com* dengan alamat email penerima.
- Ganti *https://example.com/unsub* dengan tautan berhenti berlangganan Anda.
- Ganti *example123456* dengan pengenal unik untuk penerima.

```
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"
}
]
```

Kirim pesan SMS

Anda dapat menggunakan Amazon Pinpoint API untuk mengirim pesan SMS (pesan teks) ke nomor telepon atau ID titik akhir tertentu. Bagian ini berisi contoh kode lengkap yang dapat Anda gunakan untuk mengirim pesan SMS melalui Amazon Pinpoint API dengan menggunakan SDK. AWS

C#

Gunakan contoh ini untuk mengirim pesan SMS dengan menggunakan file [AWS SDK for .NET](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi file. AWS SDK for .NET Untuk informasi selengkapnya, lihat [Memulai](#) di Panduan AWS SDK for .NET Pengembang.

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna IAM yang ada. Untuk informasi selengkapnya, lihat [Mengonfigurasi AWS kredensial di Panduan](#) Pengembang. AWS SDK for .NET

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

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
```



```
.SetBasePath(Directory.GetCurrentDirectory())
.AddJsonFile("settings.json") // Load test settings from .json file.
.AddJsonFile("settings.local.json",
    true) // Optionally load local settings.
.Build();

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

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

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

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

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

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

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

try
{
    var response = await SendSmsMessage(region, appId, destinationNumber,
        originationNumber, registeredKeyword, senderId, messageType);
}
```

```

        Console.WriteLine($"Message sent to
{response.MessageResponse.Result.Count} recipient(s).");
        foreach (var messageResultValue in
            response.MessageResponse.Result.Select(r => r.Value))
        {
            Console.WriteLine($"{messageResultValue.MessageId} Status:
{messageResultValue.DeliveryStatus}");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
    }
}

public static async Task<SendMessagesResponse> SendSmsMessage(
    string region, string appId, string destinationNumber, string
originationNumber,
    string? keyword, string? senderId, MessageType messageType)
{
    // The content of the SMS message.
    string message = "This message was sent through Amazon Pinpoint using" +
        " the AWS SDK for .NET. Reply STOP to opt out.";

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

    SendMessagesRequest sendRequest = new SendMessagesRequest
    {
        ApplicationId = appId,
        MessageRequest = new MessageRequest
        {
            Addresses =
                new Dictionary<string, AddressConfiguration>
                {
                    {
                        destinationNumber,
                        new AddressConfiguration { ChannelType =
ChannelType.SMS }
                    }
                },
    }
}

```

```
        MessageConfiguration = new DirectMessageConfiguration
        {
            SMSMessage = new SMSMessage
            {
                Body = message,
                MessageType = MessageType.TRANSACTIONAL,
                OriginationNumber = originationNumber,
                SenderId = senderId,
                Keyword = keyword
            }
        }
    };
    SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
    return response;
}
```

Java

Gunakan contoh ini untuk mengirim pesan SMS dengan menggunakan file [AWS SDK for Java](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi SDK for Java. Untuk informasi lebih lanjut, lihat [Memulai](#) di Panduan Developer.AWS SDK for Java

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna IAM yang ada. Untuk informasi selengkapnya, lihat [Menetapkan kredensi default dan Wilayah](#) di Panduan AWS SDK for Java Pengembang.

```
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();

SendMessageResponse response =
pinpoint.sendMessage(request);
MessageResponse msg1 = response.getMessageResponse();
Map map1 = msg1.getResult();

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

} catch (PinpointException e) {
    System.err.println(e.getAwsErrorDetails().getErrorMessage());
    System.exit(1);
}
}
```

Untuk contoh SDK lengkap, lihat [SendMessage.java](#) on [GitHub](#)

JavaScript (Node.js)

Gunakan contoh ini untuk mengirim pesan SMS dengan menggunakan [AWS SDK untuk JavaScript di Node.js](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengonfigurasi SDK untuk JavaScript di Node.js. Untuk informasi selengkapnya, lihat [Memulai AWS SDK untuk JavaScript di Panduan Pengembang Node.js](#).

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna IAM yang ada. Untuk informasi selengkapnya, lihat [Menyetel kredensial](#) di AWS SDK untuk JavaScript di Panduan Pengembang Node.js.

```
"use strict";

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

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

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

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

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

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

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

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

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

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

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

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

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

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};
```



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

Python

Gunakan contoh ini untuk mengirim pesan SMS dengan menggunakan file [AWS SDK for Python \(Boto3\)](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi SDK untuk Python. Untuk informasi selengkapnya, lihat [Mulai cepat](#) di AWS SDK for Python (Boto3) Memulai.

```
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()
```

Anda juga dapat menggunakan template pesan untuk mengirim pesan SMS, seperti yang ditunjukkan pada contoh berikut:

```
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()
```

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna IAM yang ada. Untuk informasi selengkapnya, lihat [Kredensial di Referensi](#) API AWS SDK for Python (Boto3).

Kirim pesan suara

Anda dapat menggunakan Amazon Pinpoint API untuk mengirim pesan suara ke nomor telepon tertentu. Bagian ini berisi contoh kode lengkap yang dapat Anda gunakan untuk mengirim pesan suara melalui Amazon Pinpoint SMS dan Voice API dengan menggunakan SDK. AWS

Java

Gunakan contoh ini untuk mengirim pesan suara menggunakan [AWS SDK for Java](#). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi SDK for Java. Untuk informasi selengkapnya, lihat [Memulai](#) di Panduan AWS SDK for Java Pengembang.

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Menyiapkan AWS kredensial dan Wilayah untuk pengembangan](#) di Panduan AWS SDK for Java Pengembang.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;  
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;  
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;  
import  
    software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;  
import  
    software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
```

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

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

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

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

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

    // The content of the message. This example uses SSML to customize and
    control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
```

```

static final String ssm1Message = "<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);
    }
}
}
```

Untuk contoh SDK lengkap, lihat [SendVoiceMessage.java](#) on [GitHub](#)

JavaScript (Node.js)

Gunakan contoh ini untuk mengirim pesan suara menggunakan AWS SDK for JavaScript di Node.js. Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengonfigurasi SDK untuk JavaScript di Node.js.

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Menyetel kredensial](#) di AWSSDK untuk JavaScript di Panduan Pengembang 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

Gunakan contoh ini untuk mengirim pesan suara menggunakan AWS SDK for Python (Boto3). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi SDK for Python (Boto3).

Contoh ini mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Kredensial di Referensi](#) API AWSSDK for 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,
    ssm_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,
        ssm_l_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")

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

Kirim notifikasi push

Amazon Pinpoint API dapat mengirim notifikasi push transaksional ke pengenal perangkat tertentu. Bagian ini berisi contoh kode lengkap yang dapat Anda gunakan untuk mengirim notifikasi push melalui Amazon Pinpoint API dengan menggunakan AWS SDK.

Anda dapat menggunakan contoh ini untuk mengirim notifikasi push melalui layanan notifikasi push apa pun yang didukung Amazon Pinpoint. Saat ini, Amazon Pinpoint mendukung saluran berikut: Firebase Cloud Messaging (FCM), Apple Push Notification Service (APN), Baidu Cloud Push, dan Amazon Device Messaging (ADM).

Note

Saat Anda mengirim notifikasi push melalui layanan Firebase Cloud Messaging (FCM), gunakan nama layanan GCM dalam panggilan Anda ke API Amazon Pinpoint. Layanan Google Cloud Messaging (GCM) dihentikan oleh Google pada 10 April 2018. Namun, Amazon Pinpoint API menggunakan nama GCM layanan untuk pesan yang dikirimkannya

melalui layanan FCM untuk menjaga kompatibilitas dengan kode API yang ditulis sebelum penghentian layanan GCM.

JavaScript (Node.js)

Gunakan contoh ini untuk mengirim notifikasi push menggunakan AWS SDK untuk JavaScript di Node.js. Contoh ini menganggap bahwa Anda telah menginstal dan mengonfigurasi SDK for JavaScript di Node.js.

Contoh ini juga mengasumsikan bahwa Anda menggunakan file kredensi bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Menyetel kredensi](#) di AWSSDK for JavaScript in Panduan Pengembang Node.js.

```
'use strict';

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

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

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

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

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

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

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

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

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

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

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

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

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



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

return messageRequest
}

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

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

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

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

    //Create a new Pinpoint object.
```

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

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

SendMessage()
```

Python

Gunakan contoh ini untuk mengirim pemberitahuan push dengan menggunakan AWS SDK for Python (Boto3). Contoh ini menganggap bahwa Anda telah menginstal dan mengonfigurasi SDK for Python (Boto3).

Contoh ini juga mengasumsikan bahwa Anda menggunakan file kredensi bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi lebih lanjut, lihat [Kredensial dalam Referensi](#) API AWSSDK for Python (Boto3).

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

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

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

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

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

```
# that you choose.
application_id = "ce796be37f32f178af652b26eexample"

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

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

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

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

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

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

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

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

    if service == "GCM":
        message_request = {
```

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

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

return message_request

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

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

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

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

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

send_message()
```

Membuat saluran khusus di Amazon Pinpoint

Amazon Pinpoint menyertakan dukungan bawaan untuk mengirim pesan melalui notifikasi push, email, SMS, dan saluran suara. Anda juga dapat mengonfigurasi Amazon Pinpoint untuk mengirim pesan melalui saluran lain dengan membuat saluran khusus. Saluran khusus di Amazon Pinpoint memungkinkan Anda mengirim pesan melalui layanan apa pun yang memiliki API, termasuk layanan pihak ketiga. Anda dapat berinteraksi dengan API dengan menggunakan webhook, atau dengan memanggil AWS Lambda fungsi.

Segmen tempat Anda mengirim kampanye saluran khusus dapat berisi titik akhir dari semua jenis (yaitu, titik akhir di mana nilai `ChannelType` atribut adalah EMAIL, SUARA, SMS, KUSTOM, atau salah satu dari berbagai jenis titik akhir pemberitahuan push).

Membuat kampanye yang mengirim pesan melalui saluran khusus

Untuk menetapkan fungsi Lambda atau webhook ke kampanye individual, gunakan Amazon Pinpoint API untuk membuat atau memperbarui [Kampanye](#) objek.

`KlasterMessageConfiguration` objek dalam kampanye juga harus berisi `CustomMessage` objek. Objek ini memiliki satu anggota, `Data`. Nilai dari `Data` adalah string JSON yang berisi payload pesan yang ingin Anda kirim ke saluran khusus.

Kampanye harus berisi `aCustomDeliveryConfiguration` objek.

Dalam `CustomDeliveryConfiguration` objek, tentukan hal berikut:

- `EndpointTypes`- Array yang berisi semua jenis titik akhir yang harus dikirimkan kampanye saluran khusus. Ini dapat berisi salah satu atau semua jenis saluran berikut:
 - ADM
 - APNS
 - APNS_SANDBOX
 - APNS_VOIP
 - APNS_VOIP_SANDBOX
 - BAIDU
 - CUSTOM
 - EMAIL
 - GCM

- SMS
- VOICE
- `DeliveryUri`— Tujuan yang endpoint dikirim ke. Anda hanya dapat menentukan salah satu dari yang berikut ini:
 - Nama Sumber Daya Amazon (ARN) dari fungsi Lambda yang ingin Anda jalankan ketika kampanye berjalan.
 - URL webhook yang ingin Anda kirimkan data titik akhir saat kampanye berjalan.

Note

KlasterCampaignobjek juga dapat berisiHookobjek. Objek ini hanya digunakan untuk membuat segmen yang disesuaikan dengan fungsi Lambda saat kampanye dijalankan. Untuk informasi selengkapnya, lihat [Menyesuaikan segmen dengan AWS Lambda](#).

Memahami data peristiwa yang dikirim Amazon Pinpoint ke saluran khusus

Sebelum membuat fungsi Lambda yang mengirim pesan melalui saluran khusus, Anda harus membiasakan diri dengan data yang dipancarkan Amazon Pinpoint. Saat kampanye Amazon Pinpoint mengirim pesan melalui saluran khusus, kampanye Amazon Pinpoint akan mengirimkan payload ke fungsi Lambda target yang menyerupai contoh berikut:

```
{
  "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"
  }
}
```

Data peristiwa menyediakan atribut berikut:

- **ApplicationId**— ID proyek Amazon Pinpoint yang dimiliki kampanye.
- **CampaignId**- ID kampanye Amazon Pinpoint yang memanggil fungsi Lambda.
- **TreatmentId**— ID varian kampanye. Jika Anda membuat kampanye standar, nilai ini selalu 0. Jika Anda membuat kampanye pengujian A/B, nilai ini adalah bilangan bulat antara 0 dan 4.
- **ActivityId**— ID kegiatan yang dilakukan oleh kampanye.
- **ScheduledTime**- Waktu ketika Amazon Pinpoint mengeksekusi kampanye, ditampilkan dalam format ISO 8601.
- **Endpoints**- Daftar titik akhir yang ditargetkan oleh kampanye. Setiap payload dapat berisi hingga 50 titik akhir. Jika segmen tempat kampanye dikirimkan berisi lebih dari 50 titik akhir, Amazon Pinpoint memanggil fungsi berulang kali, hingga 50 titik akhir sekaligus, hingga semua titik akhir diproses.

Anda dapat menggunakan data sampel ini ketika membuat dan menguji fungsi Lambda saluran kustom Anda.

Mengonfigurasi webhooks

Jika Anda menggunakan webhook untuk mengirim pesan saluran khusus, URL webhook harus dimulai dengan "https://". URL webhook hanya boleh berisi karakter alfanumerik, dan simbol berikut: tanda hubung (-), titik (.), garis bawah (_), garis bawah (*), tanda tambah (?), tanda tambah (?), tanda tambah (*), tanda tambah (*), tanda tambah (. , slash atau solidus (/), pound atau tanda hash (#), dan titik koma (:). URL harus mematuhi [RFC3986](#).

Saat Anda membuat kampanye yang menentukan URL webhook, Amazon Pinpoint mengeluarkan HTTPHEAD ke URL tersebut. Tanggapan terhadapHEAD permintaan harus berisi header yang disebut X-Amz-Pinpoint-AccountId. Nilai header ini harus sama dengan AndaAWSID akun.

Mengonfigurasi fungsi Lambda

Bagian ini memberikan ikhtisar langkah-langkah yang perlu Anda ambil saat membuat fungsi Lambda yang mengirim pesan melalui saluran khusus. Pertama, Anda membuat fungsi. Setelah itu, Anda menambahkan kebijakan eksekusi ke fungsi. Kebijakan ini memungkinkan Amazon Pinpoint untuk menjalankan kebijakan saat kampanye berjalan.

Untuk pengenalan untuk membuat fungsi Lambda, lihat [Membangun fungsi Lambda](#) di dalam AWS Lambda Panduan Pengembang.

Contoh fungsi Lambda

Contoh kode berikut memproses payload dan mencatat jumlah titik akhir dari setiap jenis endpoint di 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)
```

Saat kampanye Amazon Pinpoint menjalankan fungsi Lambda ini, Amazon Pinpoint mengirimkan fungsi daftar anggota segmen. Fungsi menghitung jumlah titik akhir masing-masingChannelType. Kemudian mengirimkan data itu ke Amazon CloudWatch. Anda dapat melihat metrik ini diMetrikbagian dari CloudWatch konsol. Metrik tersebut tersedia diPinpointCustomChannelExecutionnamespace.

Anda dapat memodifikasi contoh kode ini sehingga juga terhubung ke API layanan eksternal untuk mengirim pesan melalui layanan tersebut.

Format respons fungsi Lambda untuk Amazon Pinpoint

Jika Anda ingin menggunakan perjalanan multivariat atau yes/no split untuk menentukan jalur endpoint setelah aktivitas saluran kustom, Anda harus menyusun respons fungsi Lambda Anda ke dalam format yang dapat dipahami Amazon Pinpoint, dan kemudian mengirim titik akhir ke jalur yang benar.

Struktur respon harus dalam format berikut:

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

Ini kemudian akan memungkinkan Anda memilih kunci dan nilai yang ingin Anda tentukan jalur endpoint.

Memberikan izin Amazon Pinpoint untuk mengaktifkan fungsi Lambda

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk menambahkan izin ke kebijakan fungsi Lambda yang ditetapkan ke fungsi Lambda Anda. Untuk mengizinkan Amazon Pinpoint menjalankan fungsi, gunakan Lambda [izin tambahan](#) perintah, seperti yang ditunjukkan oleh contoh berikut:

```
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
```

Pada perintah sebelumnya, lakukan hal berikut:

- Ganti *MyFunction* dengan nama fungsi Lambda.
- Ganti *us-east-1* dengan AWS Wilayah tempat Anda menggunakan Amazon Pinpoint.
- Ganti *111122223333* dengan ID akun AWS Anda.

Ketika Anda menjalankan `add-permission` perintah, Lambda mengembalikan output berikut:


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

Klaster `Statement` nilai adalah versi string JSON dari pernyataan yang ditambahkan ke kebijakan fungsi Lambda.

Lebih lanjut membatasi kebijakan eksekusi

Anda dapat memodifikasi kebijakan eksekusi dengan membatasinya ke proyek Amazon Pinpoint tertentu. Untuk melakukan ini, ganti *** dalam contoh sebelumnya dengan ID unik proyek. Anda dapat membatasi kebijakan lebih lanjut dengan membatasi kebijakan tersebut pada kampanye tertentu. Misalnya, membatasi kebijakan hanya mengizinkan kampanye dengan ID kampanye `95fee4cd1d7f5cd67987c1436example` dalam proyek dengan ID proyek `dbaf6ec2226f0a9a8615e3ea5example`, gunakan nilai berikut untuk `source-arn` atribut:

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

 **Note**

Jika Anda membatasi eksekusi fungsi Lambda ke kampanye tertentu, pertama-tama Anda harus membuat fungsi dengan kebijakan yang tidak terlalu ketat. Selanjutnya, Anda harus membuat kampanye di Amazon Pinpoint dan memilih fungsinya. Terakhir, Anda harus memperbarui kebijakan eksekusi untuk merujuk ke kampanye yang ditentukan.

Streaming acara Amazon Pinpoint ke Kinesis

Di Amazon Pinpoint, peristiwa adalah tindakan yang terjadi saat pengguna berinteraksi dengan salah satu aplikasi Anda, saat Anda mengirim pesan dari kampanye atau perjalanan, atau saat Anda mengirim SMS transaksional atau pesan email. Misalnya, jika Anda mengirim pesan email, beberapa peristiwa terjadi:

- Saat Anda mengirim pesan, peristiwa kirim terjadi.
- Saat pesan mencapai kotak masuk penerima, peristiwa terkirim akan terjadi.
- Ketika penerima membuka pesan, peristiwa terbuka terjadi.

Anda dapat mengonfigurasi Amazon Pinpoint untuk mengirim informasi tentang peristiwa ke Amazon Kinesis. Platform Kinesis menawarkan layanan yang dapat Anda gunakan untuk mengumpulkan, memproses, dan menganalisis data dari AWS layanan secara real time. Amazon Pinpoint dapat mengirim data peristiwa ke Firehose, yang mengalirkan data ini ke penyimpanan AWS data seperti Amazon S3 atau Amazon Redshift. Amazon Pinpoint juga dapat mengalirkan data ke Kinesis Data Streams, yang menyerap dan menyimpan beberapa aliran data untuk diproses oleh aplikasi analitik.

Aliran peristiwa Amazon Pinpoint mencakup informasi tentang interaksi pengguna dengan aplikasi (aplikasi) yang Anda sambungkan ke Amazon Pinpoint. Ini juga mencakup informasi tentang semua pesan yang Anda kirim dari kampanye, melalui saluran apa pun, dan dari perjalanan. Ini juga dapat mencakup peristiwa kustom apa pun yang telah Anda tentukan. Terakhir, ini mencakup informasi tentang semua email transaksional dan pesan SMS yang Anda kirim.

Note

Amazon Pinpoint tidak mengalirkan informasi tentang pemberitahuan push transaksional atau pesan suara.

Bab ini memberikan informasi tentang pengaturan Amazon Pinpoint untuk mengalirkan data peristiwa ke Kinesis. Ini juga berisi contoh data peristiwa yang dialirkan Amazon Pinpoint.

Topik

- [Menyiapkan streaming acara](#)
- [Acara aplikasi](#)

- [Acara kampanye](#)
- [Acara perjalanan](#)
- [Peristiwa email](#)
- [Acara SMS](#)

Menyiapkan streaming acara

Anda dapat mengatur Amazon Pinpoint untuk mengirim data peristiwa ke aliran Amazon Kinesis atau aliran pengiriman Amazon Data Firehose. Amazon Pinpoint dapat mengirim data acara untuk kampanye, perjalanan, dan email transaksional dan pesan SMS.

Bagian ini mencakup informasi tentang pengaturan streaming acara secara terprogram. Anda juga dapat menggunakan konsol Amazon Pinpoint untuk mengatur streaming acara. Untuk informasi tentang mengatur streaming acara menggunakan konsol Amazon Pinpoint, lihat [Pengaturan aliran peristiwa di Panduan Pengguna Amazon Pinpoint](#).

Prasyarat

Contoh di bagian ini memerlukan masukan berikut:

- ID aplikasi aplikasi yang terintegrasi dengan Amazon Pinpoint dan melaporkan peristiwa. Untuk informasi tentang cara mengintegrasikan, lihat [Mengintegrasikan Amazon Pinpoint dengan aplikasi Anda](#).
- Nama Sumber Daya Amazon (ARN) dari aliran Kinesis atau aliran pengiriman Firehose di akun Anda. AWS Untuk informasi tentang cara membuat sumber daya ini, lihat [Membuat dan Mengelola Aliran](#) di Panduan Pengembang Amazon Kinesis Data Streams atau [Membuat aliran pengiriman Amazon Data Firehose](#) di Panduan Pengembang [Amazon Data Firehose](#).
- ARN dari peran AWS Identity and Access Management (IAM) yang mengizinkan Amazon Pinpoint untuk mengirim data ke aliran. Untuk informasi tentang membuat peran, lihat [Peran IAM untuk streaming acara ke Kinesis](#).

AWS CLI

AWS CLI Contoh berikut menggunakan [put-event-stream](#) perintah. Perintah ini mengonfigurasi Amazon Pinpoint untuk mengirim peristiwa ke aliran Kinesis:

```
aws pinpoint put-event-stream \
```

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

AWS SDK for Java

Contoh Java berikut mengonfigurasi Amazon Pinpoint untuk mengirim peristiwa ke aliran 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);  
}
```

Contoh ini membangun [WriteEventStream](#) objek yang menyimpan ARN dari aliran Kinesis dan peran IAM. [WriteEventStream](#) objek diteruskan ke [PutEventStreamRequest](#) objek untuk mengonfigurasi Amazon Pinpoint untuk mengalirkan peristiwa untuk aplikasi tertentu. [PutEventStreamRequest](#) objek diteruskan ke [putEventStream](#) metode klien Amazon Pinpoint.

Anda dapat menetapkan aliran Kinesis ke beberapa aplikasi. Jika Anda melakukan ini, Amazon Pinpoint mengirimkan data peristiwa yang dikodekan dalam base64 dari setiap aplikasi ke aliran, yang memungkinkan Anda menganalisis data sebagai koleksi. Metode contoh berikut menerima daftar ID aplikasi (aplikasi), dan menggunakan metode contoh sebelumnya `createEventStream`, untuk menetapkan aliran ke setiap aplikasi:

```
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());  
}
```

Meskipun Anda dapat menetapkan satu aliran ke beberapa aplikasi, Anda tidak dapat menetapkan beberapa aliran ke satu aplikasi.

Menonaktifkan streaming acara

Jika Anda menetapkan aliran Kinesis ke aplikasi, Anda dapat menonaktifkan streaming acara untuk aplikasi itu. Amazon Pinpoint menghentikan streaming peristiwa ke Kinesis, tetapi Anda dapat melihat analisis peristiwa dengan menggunakan konsol Amazon Pinpoint.

AWS CLI

Gunakan [delete-event-stream](#) perintah:

```
aws pinpoint delete-event-stream --application-id application-id
```

AWS SDK for Java

Gunakan [deleteEventStream](#) metode klien Amazon Pinpoint:

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

Acara aplikasi

Setelah Anda mengintegrasikan aplikasi (aplikasi) dengan Amazon Pinpoint, Amazon Pinpoint dapat melakukan streaming data peristiwa tentang aktivitas pengguna, acara khusus, dan pengiriman pesan untuk aplikasi.

Contoh

Objek JSON untuk kejadian aplikasi berisi data yang ditampilkan dalam contoh berikut.


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

Atribut peristiwa

Bagian ini mendefinisikan atribut yang disertakan dalam aliran peristiwa aplikasi.

Atribut	Deskripsi
event_type	Tipe peristiwa. Nilai yang mungkin adalah: <ul style="list-style-type: none">_session.start— Endpoint memulai sesi baru.

Atribut	Deskripsi
	<ul style="list-style-type: none">• <code>session.stop</code>— Endpoint mengakhiri sesi.• <code>_userauth.sign_in</code>— Endpoint masuk ke aplikasi.• <code>_userauth.sign_up</code>— Endpoint baru menyelesaikan proses pendaftaran di aplikasi Anda.• <code>_userauth.auth_fail</code>— Endpoint berusaha untuk masuk ke aplikasi Anda, tetapi tidak dapat menyelesaikan proses.• <code>_monetisasi.pembelian</code>— Endpoint melakukan pembelian di aplikasi Anda.• <code>_session.jeda</code>— Endpoint menghentikan sesi. Sesi yang dijeda dapat dilanjutkan sehingga Anda dapat terus mengumpulkan metrik tanpa memulai sesi yang sama sekali baru.• <code>_session.resume</code>— Endpoint dilanjutkan sesi.
<code>event_timestamp</code>	Waktu ketika acara dilaporkan, ditampilkan sebagai waktu Unix dalam milidetik.
<code>arrival_timestamp</code>	Waktu ketika acara diterima oleh Amazon Pinpoint, ditampilkan sebagai waktu Unix dalam milidetik.
<code>event_version</code>	Versi acara JSON skema. <div data-bbox="829 1444 1507 1759" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Tip</p><p>Periksa versi ini di aplikasi pemrosesan acara Anda sehingga Anda tahu kapan harus memperbarui aplikasi sebagai respons terhadap pembaruan skema.</p></div>

Atribut	Deskripsi
<code>application</code>	Informasi tentang proyek Amazon Pinpoint yang terkait dengan acara tersebut. Untuk informasi selengkapnya, lihat Aplikasi tabel.
<code>client</code>	Informasi tentang titik akhir yang melaporkan peristiwa tersebut. Untuk informasi selengkapnya, lihat Klien tabel.
<code>device</code>	Informasi tentang perangkat yang melaporkan peristiwa tersebut. Untuk informasi selengkapnya, lihat Perangkat tabel.
<code>session</code>	Informasi tentang sesi yang dihasilkan acara. Untuk informasi selengkapnya, lihat Sesi tabel.
<code>attributes</code>	Atribut yang terkait dengan kejadian. Untuk peristiwa yang dilaporkan oleh aplikasi Anda, objek ini menyertakan atribut kustom yang Anda tetapkan.
<code>metrics</code>	Metrik yang terkait dengan acara tersebut. Anda dapat mengonfigurasi aplikasi secara opsional untuk mengirim metrik khusus ke Amazon Pinpoint.

Aplikasi

Termasuk informasi tentang proyek Amazon Pinpoint yang terkait dengan acara tersebut.

Atribut	Deskripsi
<code>app_id</code>	ID unik dari proyek Amazon Pinpoint yang melaporkan acara tersebut.
<code>cognito_identity_pool_id</code>	ID dari Amazon Cognito Identity Pool yang terkait dengan titik akhir.

Atribut	Deskripsi
<code>package_name</code>	Nama paket aplikasi, seperti <code>com.example.my_app</code> .
<code>sdk</code>	Informasi tentang SDK yang digunakan untuk melaporkan acara tersebut. Untuk informasi selengkapnya, lihat SDK tabel.
<code>title</code>	Nama aplikasi.
<code>version_name</code>	Nama versi aplikasi, seperti <code>V2.5</code> .
<code>version_code</code>	Nomor versi aplikasi, seperti <code>3</code> .

SDK

Termasuk informasi tentang SDK yang digunakan untuk melaporkan acara tersebut.

Atribut	Deskripsi
<code>name</code>	Nama SDK yang digunakan untuk melaporkan acara tersebut.
<code>version</code>	Versi SDK.

Klien

Termasuk informasi tentang titik akhir yang menghasilkan acara.

Atribut	Deskripsi
<code>client_id</code>	ID dari titik akhir.
<code>cognito_id</code>	Token ID Amazon Cognito yang terkait dengan titik akhir.

Perangkat

Termasuk informasi tentang perangkat titik akhir yang menghasilkan acara.

Atribut	Deskripsi
<code>locale</code>	Informasi tentang pengaturan bahasa dan wilayah untuk perangkat endpoint. Untuk informasi selengkapnya, lihat Lokal tabel.
<code>make</code>	Produsen perangkat endpoint.
<code>model</code>	Model pengenalan perangkat endpoint.
<code>platform</code>	Informasi tentang sistem operasi pada perangkat endpoint. Untuk informasi selengkapnya, lihat Platform tabel.

Lokal

Termasuk informasi tentang pengaturan bahasa dan wilayah untuk perangkat endpoint.

Atribut	Deskripsi
<code>code</code>	Locale identifier yang terkait dengan perangkat.
<code>country</code>	Negara atau wilayah yang terkait dengan lokal perangkat.
<code>language</code>	Bahasa yang terkait dengan lokal perangkat.

Platform

Termasuk informasi tentang sistem operasi pada perangkat endpoint.

Atribut	Deskripsi
<code>name</code>	Nama sistem operasi pada perangkat.

Atribut	Deskripsi
version	Versi sistem operasi pada perangkat.

Sesi

Termasuk informasi tentang sesi yang dihasilkan acara.

Atribut	Deskripsi
session_id	ID unik yang mengidentifikasi sesi.
start_timestamp	Tanggal dan waktu saat sesi dimulai, ditampilkan sebagai waktu Unix dalam milidetik.
stop_timestamp	Tanggal dan waktu saat sesi berakhir, ditampilkan sebagai waktu Unix dalam milidetik.

Acara kampanye

Jika Anda menggunakan Amazon Pinpoint untuk mengirim kampanye melalui saluran apa pun, Amazon Pinpoint dapat melakukan streaming data peristiwa tentang kampanye tersebut. Ini termasuk data peristiwa untuk setiap email atau pesan SMS yang Anda kirim dari kampanye. Untuk informasi terperinci tentang data yang dialirkan Amazon Pinpoint untuk jenis pesan tersebut, lihat [the section called “Peristiwa email”](#) dan [the section called “Acara SMS”](#).

Contoh acara

Objek JSON untuk acara kampanye berisi data yang ditampilkan dalam contoh berikut.

```
{
  "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-f540ccexample"
},
"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"
}


```

Atribut acara kampanye

Bagian ini mendefinisikan atribut yang disertakan dalam aliran peristiwa kampanye.

Atribut	Deskripsi
event_type	<p>Tipe peristiwa. Nilai yang mungkin adalah:</p> <ul style="list-style-type: none"> _campaign.send — Amazon Pinpoint mengeksekusi kampanye. _campaign.opened_notification - Untuk kampanye notifikasi push, jenis peristiwa ini menunjukkan bahwa penerima mengetuk notifikasi untuk membukanya. _campaign.received_foreground - Untuk kampanye notifikasi push, jenis peristiwa ini

Atribut	Deskripsi
	<p>menunjukkan bahwa penerima menerima pesan sebagai pemberitahuan latar depan.</p> <ul style="list-style-type: none"> • <code>_campaign.received_background</code> - Untuk kampanye notifikasi push, jenis peristiwa ini menunjukkan bahwa penerima menerima pesan sebagai pemberitahuan latar belakang. <div data-bbox="862 583 1511 1234" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p><code>_campaign.opened_notification</code>, <code>_campaign.received_foreground</code>, dan <code>_campaign.received_background</code> dikembalikan hanya jika Anda menggunakannya AWS Amplify. Untuk informasi selengkapnya tentang mengintegrasikan aplikasi Anda AWS Amplify. Lihat Menghubungkan aplikasi frontend Anda ke Amazon Pinpoint dengan Amplify AWS.</p> </div>
<code>event_timestamp</code>	Waktu ketika acara dilaporkan, ditampilkan sebagai waktu Unix dalam milidetik.
<code>arrival_timestamp</code>	Waktu ketika acara diterima oleh Amazon Pinpoint, ditampilkan sebagai waktu Unix dalam milidetik.

Atribut	Deskripsi
event_version	Versi acara JSON skema. <div data-bbox="829 302 1507 617" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Tip Periksa versi ini di aplikasi pemrosesan acara Anda sehingga Anda tahu kapan harus memperbarui aplikasi sebagai respons terhadap pembaruan skema.</p></div>
application	Informasi tentang proyek Amazon Pinpoint yang terkait dengan peristiwa tersebut. Untuk informasi lebih lanjut, lihat tabel Application .
client	Informasi tentang titik akhir yang dikaitkan dengan peristiwa tersebut. Untuk informasi lebih lanjut, lihat tabel Client .
device	Informasi tentang perangkat yang melaporkan peristiwa tersebut. Untuk pesan kampanye dan transaksional, objek ini kosong.
session	Informasi tentang sesi yang dihasilkan acara. Untuk kampanye, objek ini kosong.

Atribut	Deskripsi
<code>attributes</code>	<p>Atribut yang terkait dengan peristiwa tersebut. Untuk peristiwa yang dilaporkan oleh salah satu aplikasi Anda, objek ini dapat menyertakan atribut khusus yang ditentukan oleh aplikasi. Untuk peristiwa yang dibuat saat Anda mengirim kampanye, objek ini berisi atribut yang terkait dengan kampanye. Untuk peristiwa yang dihasilkan saat Anda mengirim pesan transaksional, objek ini berisi informasi yang terkait dengan pesan itu sendiri.</p> <p>Untuk informasi lebih lanjut, lihat tabel Attributes.</p>
<code>client_context</code>	<p>Basic custom objek, yang berisi endpoint properti. endpointProperti berisi isi catatan titik akhir untuk titik akhir yang dikirimkan kampanye.</p>
<code>awsAccountId</code>	<p>ID AWS akun yang digunakan untuk mengirim pesan.</p>

Aplikasi

Menyertakan informasi tentang proyek Amazon Pinpoint yang terkait dengan peristiwa tersebut.

Atribut	Deskripsi
<code>app_id</code>	<p>ID unik dari proyek Amazon Pinpoint yang melaporkan acara tersebut.</p>
<code>sdk</code>	<p>SDK yang digunakan untuk melaporkan acara tersebut.</p>

Atribut

Ini termasuk informasi tentang kampanye yang menghasilkan peristiwa tersebut.

Atribut	Deskripsi
<code>treatment_id</code>	Jika pesan dikirim menggunakan kampanye pengujian A/B, nilai ini menunjukkan nomor perlakuan pesan. Untuk kampanye standar, nilai ini adalah 0.
<code>campaign_activity_id</code>	ID unik yang dihasilkan Amazon Pinpoint saat peristiwa itu terjadi.
<code>delivery_type</code>	<p>Metode pengiriman untuk kampanye. Jangan bingung atribut ini dengan <code>ChannelType</code> bidang yang ditentukan di bawah <code>endpoint.properties.client_context.ChannelType</code>. Bidang ini biasanya didasarkan pada titik akhir bahwa pesan sedang dikirim ke.</p> <p>Untuk saluran yang hanya mendukung satu jenis titik akhir, <code>delivery_type</code> dan <code>ChannelType</code> bidang memiliki nilai yang sama. Misalnya, untuk saluran email, <code>delivery_type</code> dan <code>ChannelType</code> bidang memiliki nilai EMAIL yang sama.</p> <p>Namun, kondisi ini tidak selalu berlaku untuk saluran yang mendukung berbagai jenis titik akhir, seperti saluran khusus. Anda dapat menggunakan saluran khusus untuk titik akhir yang berbeda, seperti EMAIL, SMS, CUSTOM, dan sebagainya. Dalam hal ini, <code>delivery_type</code> mengidentifikasi peristiwa pengiriman kustom, KUSTOM, dan <code>ChannelType</code> menentukan jenis titik akhir yang dikirimkan kampanye, seperti EMAIL, SMS, KUSTOM,</p>

Atribut	Deskripsi
	<p>dan sebagainya. Untuk informasi selengkapnya tentang cara membuat saluran khusus, lihat Membuat saluran khusus.</p> <p>Nilai yang mungkin adalah:</p> <ul style="list-style-type: none">• SUREL• SMS• ADM• APN• APNS_SANDBOX• APNS_VOIP• APNS_VOIP_SANDBOX• SUARA• GCM• BAIDU• MENDORONG• KUSTOM
campaign_id	ID unik dari kampanye asal pesan dikirim.

Atribut	Deskripsi
<code>campaign_send_status</code>	<p>Menunjukkan status kampanye untuk titik akhir target. Nilai yang mungkin termasuk:</p> <ul style="list-style-type: none">• SUKSES - Kampanye berhasil dikirim ke titik akhir.• KEGAGALAN - Kampanye tidak dikirim ke titik akhir.• DAILY_CAP — Kampanye tidak dikirim ke titik akhir karena jumlah maksimum pesan harian telah dikirim ke titik akhir.• KEDALUWARSA — Kampanye tidak dikirim ke titik akhir karena pengirimannya akan melebihi durasi maksimum atau pengaturan tarif pengiriman untuk kampanye.• QUIET_TIME - Kampanye tidak dikirim ke titik akhir karena pembatasan waktu tenang.• HOLDOUT - Kampanye tidak dikirim ke titik akhir karena titik akhir adalah anggota grup holdout.• DUPLICATE_ADDRESS - Ada duplikat alamat endpoint di segmen. Kampanye dikirim sekali ke alamat endpoint.• QUIET_TIME - Kampanye tidak dikirim ke titik akhir karena pembatasan waktu tenang.• CAMPAIGN_CAP — Kampanye tidak dikirim ke titik akhir karena jumlah maksimum pesan telah dikirim ke titik akhir dari kampanye ini.• FAILURE_PERMANENT - Kegagalan permanen terjadi saat mengirim ke titik akhir.• TRANSIENT_FAILURE - Kegagalan sementara terjadi saat mengirim ke titik akhir.• THROTTLED - Pengiriman dicekik.

Atribut	Deskripsi
	<ul style="list-style-type: none"> • TIDAK DIKETAHUI - Kegagalan tidak diketahui. • HOOK_FAILURE - Kait kampanye gagal. • CUSTOM_DELIVERY_FAILURE - Pengiriman kustom gagal. • REKOMENDASI_FAILURE - Recommender gagal. • UNSUPPORTED_CHANNEL - Saluran tidak didukung.

Klien

Termasuk informasi tentang titik akhir yang ditargetkan oleh kampanye.

Atribut	Deskripsi
client_id	ID titik akhir yang dikirimkan kampanye.

Acara perjalanan

Jika Anda mempublikasikan perjalanan, Amazon Pinpoint dapat mengalirkan data acara tentang perjalanan. Ini termasuk data acara untuk setiap email, SMS, push, atau pesan khusus yang Anda kirim dari perjalanan.

Lihat berikut ini untuk informasi tentang data yang dialirkan Amazon Pinpoint:

- Untuk pesan email, lihat [the section called “Peristiwa email”](#).
- Untuk pesan SMS, lihat [acara SMS](#).

Contoh acara

Objek JSON untuk acara perjalanan berisi data yang ditunjukkan dalam sampel berikut.

```
{
```

```

"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"
}

```

Atribut acara perjalanan

Bagian ini mendefinisikan atribut yang disertakan dalam data aliran peristiwa yang dihasilkan Amazon Pinpoint untuk perjalanan.

Atribut	Deskripsi
<code>event_type</code>	Jenis peristiwa. Untuk acara perjalanan, nilai untuk atribut ini selalu <code>_journey.send</code> , yang menunjukkan bahwa Amazon Pinpoint mengeksekusi perjalanan.
<code>event_timestamp</code>	Waktu ketika peristiwa dilaporkan, ditampilkan sebagai waktu Unix dalam milidetik.
<code>arrival_timestamp</code>	Waktu ketika acara diterima oleh Amazon Pinpoint, ditampilkan sebagai waktu Unix dalam milidetik.
<code>event_version</code>	Versi skema acara JSON. <div data-bbox="829 856 1507 1171"><p> Tip</p><p>Periksa versi ini di aplikasi pemrosesan acara Anda sehingga Anda tahu kapan harus memperbarui aplikasi sebagai respons terhadap pembaruan skema.</p></div>
<code>application</code>	Informasi tentang proyek Amazon Pinpoint yang terkait dengan acara tersebut. Untuk informasi selengkapnya, lihat tabel Aplikasi .
<code>client</code>	Informasi tentang titik akhir yang terkait dengan acara tersebut. Untuk informasi selengkapnya, lihat tabel Klien .
<code>device</code>	Informasi tentang perangkat yang melaporkan acara tersebut. Untuk perjalanan, objek ini kosong.
<code>session</code>	Informasi tentang sesi yang menghasilkan acara. Untuk perjalanan, objek ini kosong.

Atribut	Deskripsi
<code>attributes</code>	Atribut yang terkait dengan aktivitas perjalanan dan perjalanan yang menghasilkan acara. Untuk informasi selengkapnya, lihat tabel Atribut .
<code>client_context</code>	Berisi custom objek, yang berisi endpoint properti. endpointProperti berisi konten catatan titik akhir untuk titik akhir yang terkait dengan peristiwa tersebut.
<code>awsAccountId</code>	ID AWS akun yang digunakan untuk menjalankan perjalanan.

Aplikasi

Termasuk informasi tentang proyek Amazon Pinpoint yang terkait dengan acara tersebut.

Atribut	Deskripsi
<code>app_id</code>	ID unik dari proyek Amazon Pinpoint yang melaporkan acara tersebut.
<code>sdk</code>	SDK yang digunakan untuk melaporkan peristiwa tersebut.


Klien

Termasuk informasi tentang titik akhir yang terkait dengan acara.


Atribut	Deskripsi
<code>client_id</code>	ID dari titik akhir.

Atribut

Termasuk informasi tentang perjalanan yang menghasilkan acara.

Atribut	Deskripsi
journey_run_id	ID unik dari perjalanan lari yang menghasilkan acara. Amazon Pinpoint menghasilkan dan menetapkan ID ini secara otomatis ke setiap perjalanan baru.
journey_send_status	<p>Menunjukkan status pengiriman pesan yang terkait dengan acara. Nilai yang mungkin termasuk:</p> <ul style="list-style-type: none">• SUKSES — Pesan berhasil dikirim ke titik akhir.• KEGAGALAN - Pesan tidak dikirim ke titik akhir karena terjadi kesalahan.• CUSTOM_DELIVERY_FAILURE - Pengiriman kustom gagal.• FAILURE_PERMANENT - Kegagalan permanen terjadi saat mengirim ke titik akhir. <div data-bbox="862 1262 1507 1873"><p> Tip</p><p>Anda dapat memfilter peristiwa dengan status FAILURE_PERMANENT dan journey_send_status_code menyetel ke 403 untuk menentukan apakah ada kebijakan akses dan pelanggaran peran. Untuk kampanye keluar dengan suara, pengecualian ini biasanya terjadi pada instance saat peran eksekusi kampanye sambungkan yang mengikat</p></div>

Atribut	Deskripsi
	<p>perjalanan Amazon Pinpoint ke kampanye Amazon Connect secara tidak sengaja dihapus untuk eksekusi perjalanan dalam penerbangan.</p> <ul style="list-style-type: none"> • THROTTLED - Pengiriman dibatasi. • UNSUPPORTED_CHANNEL - Saluran tidak didukung. • DAILY_CAP — Pesan tidak dikirim ke titik akhir karena pengiriman pesan akan melebihi jumlah maksimum pesan yang dapat dikirim oleh perjalanan atau proyek ke satu titik akhir selama periode 24 jam. • QUIET_TIME — Pesan tidak dikirim karena pembatasan waktu tenang untuk perjalanan atau proyek. • QUIET_TIME_MISSING_TIMEZONE — Pesan tidak dikirim karena estimasi zona waktu tidak dapat memperkirakan zona waktu untuk titik akhir dan waktu tenang diaktifkan.
journey_id	ID unik dari perjalanan yang menghasilkan acara.
journey_activity_id	ID unik dari aktivitas perjalanan yang menghasilkan acara.

Atribut	Deskripsi
<code>journey_activity_type</code>	Jenis kegiatan perjalanan acara. Ini bisa berupa EMAIL, SMS, PUSH, CONTACT_C ENTER, atau CUSTOM. <div data-bbox="829 401 1508 617"><p> Note VOICE bukan jenis aktivitas perjalanan yang didukung.</p></div>
<code>journey_send_status_message</code>	Deskripsi status acara kirim.
<code>journey_send_status_code</code>	Kode status HTTP dari permintaan.

Peristiwa email

Saat Anda mengirim pesan email, Amazon Pinpoint dapat melakukan streaming data yang menyediakan informasi tambahan tentang jenis peristiwa berikut untuk pesan tersebut:

- Mengirim
- Pengiriman
- Pentalan
- Aduan
- Membuka
- Klik
- Penolakan
- Berhenti berlangganan
- Kegagalan rendering

Jenis acara dalam daftar sebelumnya dijelaskan secara rinci dalam [Atribut peristiwa email](#).

Bergantung pada API dan pengaturan yang Anda gunakan untuk mengirim pesan email, Anda mungkin melihat jenis peristiwa tambahan atau data yang berbeda. Misalnya, jika Anda mengirim pesan menggunakan set konfigurasi yang mempublikasikan data peristiwa ke Amazon Kinesis,

seperti yang disediakan oleh Amazon Simple Email Service (Amazon SES), data tersebut juga dapat menyertakan peristiwa untuk kegagalan rendering template. Untuk informasi tentang data tersebut, lihat [Pemantauan menggunakan penerbitan peristiwa Amazon SES](#) di Panduan Pengembang Layanan Email Amazon Simple.

Contoh peristiwa

Kirim email

Objek JSON untuk email mengirim acara berisi data yang ditunjukkan dalam contoh berikut.

```
{
  "event_type": "_email.send",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622025,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "9a311b17-6f8e-4093-be61-4d0bbexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "received"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rn bmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }
        ]
      }
    }
  }
}
```

```
    }, {
      "name": "To",
      "value": "recipient@example.com"
    }, {
      "name": "Subject",
      "value": "Amazon Pinpoint Test"
    }, {
      "name": "MIME-Version",
      "value": "1.0"
    }, {
      "name": "Content-Type",
      "value": "multipart/alternative; boundary=\"-----=_Part_314159_271828\""
    }
  ]],
  "common_headers": {
    "from": "sender@example.com",
    "to": ["recipient@example.com"],
    "subject": "Amazon Pinpoint Test"
  }
},
"send": {}
}
}
}
```

Email dikirimkan

Objek JSON untuk email disampaikan peristiwa berisi data yang ditunjukkan dalam contoh berikut.

```
{
  "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": "0200000073rnbnmd1-mbvdlg3uo-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
  }
}
}
```

```
}  
}
```

Klik email

Objek JSON untuk acara klik email berisi data yang ditunjukkan dalam contoh berikut.

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

```

Email terbuka

Objek JSON untuk acara terbuka email berisi data yang ditunjukkan dalam contoh berikut.

```

{
    "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": "0200000073rn bmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
        "message_send_timestamp": 1564618621380,
        "from_address": "sender@example.com",
        "destination": ["recipient@example.com"],
        "headers_truncated": false,
        "headers": [{
          "name": "From",
          "value": "sender@example.com"
        }, {
          "name": "To",
          "value": "recipient@example.com"
        }, {
          "name": "Subject",
          "value": "Amazon Pinpoint Test"
        }, {
          "name": "MIME-Version",
          "value": "1.0"
        }, {
          "name": "Content-Type",
          "value": "multipart/alternative; boundary=\"-----_Part_314159_271828\""
        }, {
          "name": "Message-ID",
          "value": "null"
        }
      ]],
        "common_headers": {
          "from": "sender@example.com",
          "to": ["recipient@example.com"],
          "subject": "Amazon Pinpoint Test"
        }
      }
    }
  },
  "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)"
    }
}
}
```

Atribut peristiwa email

Bagian ini mendefinisikan atribut yang disertakan dalam data aliran peristiwa yang dihasilkan Amazon Pinpoint saat Anda mengirim pesan email.

Atribut	Deskripsi
event_type	<p>Tipe peristiwa. Nilai yang mungkin adalah:</p> <ul style="list-style-type: none">• <code>_email.send</code> — Amazon Pinpoint menerima pesan tersebut dan berusaha mengirimkannya ke penerima.• <code>_email.delivered</code> — Pesan dikirim ke penerima.• <code>_email.rejected</code> — Amazon Pinpoint menentukan bahwa pesan tersebut berisi malware dan tidak mencoba mengirimkannya.• <code>_email.hardbounce</code> — Masalah permanen mencegah Amazon Pinpoint mengirimkan pesan. Amazon Pinpoint tidak akan mencoba mengirimkan pesan lagi.• <code>_email.softbounce</code> — Masalah sementara mencegah Amazon Pinpoint mengirimkan pesan. Amazon Pinpoint akan mencoba mengirimkan pesan lagi untuk jangka waktu tertentu. Jika pesan masih tidak dapat dikirimkan, tidak ada lagi percobaan ulang.

Atribut	Deskripsi
	<p>yang akan dicoba. Keadaan akhir email kemudian akan SOFTBOUNCE.</p> <ul style="list-style-type: none">• <code>_email.complaint</code> — Penerima menerima pesan, lalu melaporkan pesan tersebut ke penyedia email mereka sebagai spam (misalnya, dengan menggunakan fitur “Laporkan Spam” dari klien email mereka).• <code>_email.open</code> — Penerima telah menerima pesan dan membukanya.• <code>_email.click</code> — Penerima telah menerima pesan dan mengklik tautan di dalamnya.• <code>_email.unsubscribe</code> — Penerima telah menerima pesan dan mengklik tautan berhenti berlangganan di dalamnya.• <code>_email.rendering_failure</code> - Email tidak dikirim karena kegagalan render. Ini dapat terjadi ketika data templat hilang atau ketika ada ketidakcocokan antara parameter templat dan data.
<code>event_timestamp</code>	Waktu ketika pesan dikirim, ditampilkan sebagai waktu Unix dalam milidetik. Nilai ini biasanya sama untuk semua peristiwa yang dihasilkan untuk pesan.
<code>arrival_timestamp</code>	Waktu ketika acara diterima oleh Amazon Pinpoint, ditampilkan sebagai waktu Unix dalam milidetik.

Atribut	Deskripsi
<code>event_version</code>	<p>Versi acara JSON skema.</p> <div data-bbox="829 302 1507 617" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Tip</p><p>Periksa versi ini di aplikasi pemrosesan acara Anda sehingga Anda tahu kapan harus memperbarui aplikasi sebagai respons terhadap pembaruan skema.</p></div>
<code>application</code>	<p>Informasi tentang proyek Amazon Pinpoint yang terkait dengan peristiwa tersebut. Lihat tabel Aplikasi untuk informasi lebih lanjut.</p>
<code>client</code>	<p>Informasi tentang klien aplikasi yang diinstal pada perangkat yang melaporkan peristiwa tersebut. Untuk informasi lebih lanjut, lihat tabel Klien.</p>
<code>device</code>	<p>Informasi tentang perangkat yang melaporkan kejadian tersebut. Untuk informasi lebih lanjut, lihat tabel Perangkat.</p> <p>Untuk peristiwa email, objek ini kosong.</p>
<code>session</code>	<p>Untuk peristiwa email, objek ini kosong.</p>

Atribut	Deskripsi
<code>attributes</code>	<p>Atribut yang terkait dengan kejadian tersebut. Untuk informasi lebih lanjut, lihat tabel Atribut.</p> <p>Untuk peristiwa yang dilaporkan oleh salah satu aplikasi Anda, objek ini dapat menyertakan atribut khusus yang ditentukan oleh aplikasi. Untuk peristiwa yang dibuat saat Anda mengirim pesan dari kampanye atau perjalanan, objek ini berisi atribut yang terkait dengan kampanye atau perjalanan. Untuk peristiwa yang dihasilkan saat Anda mengirim pesan transaksional, objek ini berisi informasi yang terkait dengan pesan itu sendiri.</p>
<code>client_context</code>	<p>Untuk peristiwa email, objek ini berisicustom objek, yang berisilegacy_identifier atribut. Nilai untuklegacy_identifier atribut adalah ID dari proyek yang pesan dikirim dari.</p>
<code>facets</code>	<p>Informasi tambahan tentang pesan, seperti header email. Lihat tabel Facets untuk informasi lebih lanjut.</p>
<code>awsAccountId</code>	<p>ID dariAWS akun yang digunakan untuk mengirim pesan.</p>

Aplikasi

Menyertakan informasi tentang proyek Amazon Pinpoint yang terkait dengan peristiwa tersebut.

Atribut	Deskripsi
<code>app_id</code>	<p>ID unik dari proyek Amazon Pinpoint yang melaporkan acara tersebut.</p>

Atribut	Deskripsi
sdk	SDK yang digunakan untuk melaporkan acara tersebut. Jika Anda mengirim pesan email transaksional dengan memanggil Amazon Pinpoint API secara langsung atau dengan menggunakan konsol Amazon Pinpoint, objek ini kosong.

Atribut

Termasuk informasi tentang kampanye atau perjalanan yang menghasilkan peristiwa tersebut.

Kampanye

Termasuk informasi tentang kampanye yang menghasilkan peristiwa tersebut.

Atribut	Deskripsi
feedback	Untuk_email.click peristiwa, nilai untuk atribut ini adalah URL tautan yang diklik penerima dalam pesan untuk menghasilkan acara. Untuk acara lainnya, nilai ini mewakili jenis acara, seperti received, opened, atau clicked.
treatment_id	Jika pesan dikirim menggunakan kampanye pengujian A/B, nilai ini menunjukkan nomor perlakuan pesan. Untuk kampanye standar dan pesan email transaksional, nilai ini adalah 0.
campaign_activity_id	ID unik yang dihasilkan Amazon Pinpoint ketika peristiwa tersebut terjadi.
campaign_id	ID unik kampanye yang mengirim pesan.

Perjalanan

Termasuk informasi tentang perjalanan yang menghasilkan peristiwa tersebut.

Atribut	Deskripsi
<code>journey_run_id</code>	ID unik dari perjalanan berjalan yang mengirim pesan. Amazon Pinpoint menghasilkan dan menetapkan ID ini secara otomatis ke setiap perjalanan baru.
<code>feedback</code>	Untuk <code>email.click</code> peristiwa, nilai untuk atribut ini adalah URL tautan yang diklik penerima dalam pesan untuk menghasilkan acara. Untuk acara lainnya, nilai ini mewakili jenis acara, seperti <code>received</code> , <code>delivered</code> , atau <code>opened</code> .
<code>journey_id</code>	ID unik dari perjalanan yang mengirim pesan.
<code>journey_activity_id</code>	ID unik dari aktivitas perjalanan yang mengirim pesan.

Klien

Pengenal unik klien yang ditargetkan oleh kampanye atau perjalanan.

Atribut	Deskripsi
<code>client_id</code>	ID klien. Nilainya adalah Endpoint ID untuk kampanye dan perjalanan, dan untuk pengiriman Transaksional, itu adalah UUID.

Faset


Termasuk informasi tentang pesan dan jenis acara.

Atribut	Deskripsi
<code>email_channel</code>	Berisimail_event objek, yang berisi dua objek:mail, dan objek yang sesuai dengan jenis acara.

Mail

Mencakup informasi tentang konten pesan e-mail, dan metadata tentang pesan.

Atribut	Deskripsi
<code>message_id</code>	ID unik pesan. Amazon Pinpoint secara otomatis menghasilkan ID ini saat menerima pesan.
<code>message_send_timestamp</code>	Tanggal dan waktu ketika pesan dikirim, dalam format yang ditentukan dalam RFC 822 .
<code>from_address</code>	Alamat email tempat pesan dikirim.
<code>destination</code>	Array yang berisi alamat email tempat pesan dikirim.
<code>headers_truncated</code>	Nilai Boolean yang menunjukkan apakah header email dipotong.
<code>headers</code>	Objek yang berisi beberapa pasangan nama-nilai yang sesuai dengan header dalam pesan. Objek ini biasanya berisi informasi tentang header berikut ini: <ul style="list-style-type: none"> • <code>From</code>—Alamat email pengirim. • <code>To</code>—Alamat email penerima. • <code>Subject</code>—Baris subjek email.

Atribut	Deskripsi
common_headers	<div data-bbox="860 210 1510 430" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p> Tip Header subjek tidak disertakan untuk acara kampanye _email.send.</p> </div> <ul style="list-style-type: none"> • MIME-Version - Menunjukkan bahwa pesan dalam format MIME. Jika header ini ada, nilainya selalu 1.0. • Content-Type - Jenis media MIME dari konten pesan. <p>Berisi informasi tentang beberapa header umum untuk pesan email. Informasi dapat mencakup tanggal ketika pesan dikirim, dan baris ke, dari, dan subjek pesan.</p>

Acara SMS

Jika saluran SMS diaktifkan untuk sebuah proyek, Amazon Pinpoint dapat melakukan streaming data peristiwa tentang pengiriman pesan SMS untuk proyek tersebut. Peristiwa SMS yang dihasilkan oleh operator dapat memakan waktu hingga 72 jam untuk diterima dan tidak boleh digunakan untuk menentukan apakah ada keterlambatan pengiriman pesan keluar. Setelah 72 jam, jika Amazon Pinpoint belum menerima peristiwa terakhir dari operator, layanan akan secara otomatis mengembalikan record_status UNKNOWN karena kami tidak tahu apa yang terjadi pada pesan itu.

Contoh

Objek JSON untuk acara SMS berisi data yang ditunjukkan pada contoh berikut.

```
{
  "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": "cbcf3c5e3bd48a8ae2b9cb41example",
    "origination_phone_number": "+12065550142",
    "destination_phone_number": "+14255550199",
    "record_status": "DELIVERED",
    "iso_country_code": "US",
    "treatment_id": "0",
    "number_of_message_parts": "1",
    "message_id": "1111-2222-3333",
    "message_type": "Transactional",
    "campaign_id": "52dc44b35c4742c98c5935269example"
  },
  "metrics": {
    "price_in_millicents_usd": 645.0
  },
  "awsAccountId": "123456789012"
}


```

Atribut acara SMS

Bagian ini mendefinisikan atribut yang disertakan dalam data aliran peristiwa yang dihasilkan Amazon Pinpoint saat Anda mengirim pesan SMS.

Peristiwa

Atribut	Deskripsi
event_type	<p>Tipe peristiwa. Nilai yang mungkin adalah:</p> <ul style="list-style-type: none"> _SMS.BUFFERED - Pesan masih dalam proses dikirim ke penerima.

Atribut	Deskripsi
	<ul style="list-style-type: none"> • <code>_SMS.SUCCESS</code> - Pesan berhasil diterima oleh operator/dikirim ke penerima. • <code>_SMS.FAILURE</code> — Amazon Pinpoint tidak dapat mengirimkan pesan ke penerima. Untuk mempelajari lebih lanjut tentang kesalahan yang mencegah pesan terkirim, lihat <code>attributes.record_status</code>. • <code>_SMS.OPTOUT</code> - Pelanggan menerima pesan dan menjawab dengan mengirimkan kata kunci opt-out (biasanya "STOP").
<code>event_timestamp</code>	Waktu ketika acara dilaporkan, ditampilkan sebagai waktu Unix dalam milidetik.
<code>arrival_timestamp</code>	Waktu ketika acara diterima oleh Amazon Pinpoint, ditampilkan sebagai waktu Unix dalam milidetik.
<code>event_version</code>	<p>Versi acara JSON skema.</p> <div data-bbox="829 1121 1507 1436" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Tip</p> <p>Periksa versi ini di aplikasi pemrosesan acara Anda sehingga Anda tahu kapan harus memperbarui aplikasi sebagai respons terhadap pembaruan skema.</p> </div>
<code>application</code>	Informasi tentang proyek Amazon Pinpoint yang terkait dengan peristiwa tersebut. Untuk informasi selengkapnya, lihat tabel Aplikasi .
<code>client</code>	Informasi tentang klien aplikasi yang diinstal pada perangkat yang melaporkan peristiwa tersebut. Untuk informasi selengkapnya, lihat tabel Klien .

Atribut	Deskripsi
<code>device</code>	<p>Informasi tentang perangkat yang melaporkan kejadian tersebut. Untuk informasi selengkapnya, lihat tabel Perangkat.</p> <p>Untuk acara SMS, objek ini kosong.</p>
<code>session</code>	<p>Untuk acara SMS, objek ini kosong.</p>
<code>attributes</code>	<p>Atribut yang terkait dengan acara tersebut. Untuk peristiwa yang dilaporkan oleh salah satu aplikasi Anda, objek ini dapat menyertakan atribut khusus yang ditentukan oleh aplikasi. Untuk peristiwa yang dibuat saat Anda mengirim kampanye, objek ini berisi atribut yang terkait dengan kampanye. Untuk peristiwa yang dihasilkan saat Anda mengirim pesan transaksional, objek ini berisi informasi yang terkait dengan pesan itu sendiri.</p> <p>Untuk informasi selengkapnya, lihat tabel Atribut.</p>
<code>metrics</code>	<p>Metrik tambahan yang terkait dengan peristiwa tersebut. Untuk informasi selengkapnya, lihat tabel Metrik.</p>
<code>awsAccountId</code>	<p>ID AWS akun yang digunakan untuk mengirim pesan.</p>

Aplikasi

Menyertakan informasi tentang proyek Amazon Pinpoint yang terkait dengan peristiwa tersebut dan, jika berlaku, SDK yang digunakan untuk melaporkan peristiwa tersebut.

Atribut	Deskripsi
app_id	ID unik dari proyek Amazon Pinpoint yang melaporkan acara tersebut.
sdk	SDK yang digunakan untuk melaporkan acara tersebut. Jika Anda mengirim pesan SMS transaksional dengan memanggil Amazon Pinpoint API secara langsung atau dengan menggunakan konsol Amazon Pinpoint, objek ini kosong.

Atribut

Termasuk informasi tentang atribut yang terkait dengan acara tersebut.

Atribut	Deskripsi
sender_request_id	ID unik yang terkait dengan permintaan untuk mengirim pesan SMS.
campaign_activity_id	ID unik dari aktivitas dalam kampanye.
origination_phone_number	Nomor telepon tempat pesan itu dikirim.
destination_phone_number	Nomor telepon yang Anda coba kirimkan pesan.
record_status	Informasi tambahan tentang status pesan. Nilai yang mungkin termasuk: <ul style="list-style-type: none">• SUKSES/DISAMPAIKAN - Pesan berhasil disampaikan.• PENDING — Pesan belum terkirim ke perangkat penerima.• INVALID — Nomor telepon tujuan tidak valid.

Atribut	Deskripsi
	<ul style="list-style-type: none">• UNREACHABLE - Perangkat penerima saat ini tidak dapat dijangkau atau tidak tersedia. Misalnya, perangkat mungkin dimatikan, atau mungkin terputus dari jaringan. Anda dapat mencoba mengirim pesan lagi nanti.• UNKNOWN - Terjadi kesalahan yang mencegah pengiriman pesan. Kesalahan ini biasanya bersifat sementara, dan Anda dapat mencoba mengirim pesan lagi nanti.• DIBLOKIR - Perangkat penerima memblokir pesan SMS dari nomor asal.• CARRIER_UNREACHABLE — Masalah dengan jaringan seluler penerima mencegah pesan terkirim. Kesalahan ini biasanya bersifat sementara, dan Anda dapat mencoba mengirim pesan lagi nanti.• SPAM — Operator seluler penerima mengidentifikasi konten pesan sebagai spam dan pengiriman pesan yang diblokir.• INVALID_MESSAGE — Badan pesan SMS tidak valid dan tidak dapat dikirimkan.• CARRIER_BLOCKED - Operator penerima telah memblokir pengiriman pesan ini. Hal ini sering terjadi ketika operator mengidentifikasi isi pesan sebagai tidak diminta atau berbahaya.• TTL_EXPIRED — Pesan SMS tidak dapat dikirimkan dalam jangka waktu tertentu. Kesalahan ini biasanya bersifat sementara, dan Anda dapat mencoba mengirim pesan lagi nanti.• MAX_PRICE_EXCEEDED — Mengirim pesan akan mengakibatkan biaya yang

Atribut	Deskripsi
	<p>melebihi kuota belanja SMS bulanan untuk akun Anda. Anda dapat meminta kenaikan kuota ini dengan menyelesaikan prosedur dalam Meminta kenaikan kuota belanja SMS bulanan Anda di Panduan Pengguna Amazon Pinpoint.</p> <ul style="list-style-type: none">• OPTED_OUT - Pesan SMS tidak dikirim karena penerima memilih untuk tidak menerima pesan dari Anda.• NO_QUOTA_LEFT_ON_ACCOUNT - Tidak ada cukup kuota pengeluaran yang tersisa di akun Anda untuk mengirim pesan. Anda dapat meminta kenaikan kuota ini dengan menyelesaikan prosedur dalam Meminta kenaikan kuota belanja SMS bulanan Anda di Panduan Pengguna Amazon Pinpoint.• NO_ORIGINATION_IDENTITY_AVAILABLE_TO_SEND — Akun Anda tidak berisi nomor telepon yang dapat digunakan untuk mengirim pesan ke tujuan.• DESTINATION_COUNTRY_NOT_SUPPORTED - Negara tujuan diblokir. Untuk semua negara yang didukung, lihat Negara dan wilayah yang didukung (saluran SMS)• ACCOUNT_IN_SANDBOX - Akun Anda ada di kotak pasir dan hanya dapat mengirim ke nomor tujuan terverifikasi. Anda dapat memverifikasi nomor tujuan di konsol Amazon Pinpoint atau memulai proses untuk memindahkan akun dari kotak pasir, lihat Pindah dari kotak pasir SMS Amazon Pinpoint ke produksi.• RATE_EXCEEDED - Anda mencoba mengirim pesan terlalu cepat dan dicekik.

Atribut	Deskripsi
	<p>Anda perlu memperlambat tingkat panggilan Anda. Untuk detail tentang batasan kami, lihat Batas Bagian Pesan per Detik (MPS).</p> <ul style="list-style-type: none">• INVALID_ORIGINATION_IDENTITY - Identitas originasi yang disediakan tidak valid.• ORIGINATION_IDENTITY_DOES_NOT_EXIST - Identitas originasi yang disediakan tidak ada.• INVALID_DLT_PARAMETERS - Parameter DLT tidak valid (diperlukan untuk tujuan di India) disediakan.• INVALID_PARAMETERS - Parameter tidak valid disediakan.• ACCESS_DENIED — Akun Anda diblokir untuk mengirim pesan. Hubungi dukungan pelanggan untuk mengetahui penyebabnya dan menyelesaikan masalah.• INVALID_KEYWORD - Kata kunci yang disediakan tidak valid. Kata kunci bisa dalam format yang salah atau tidak diatur di akun Anda.• INVALID_SENDER_ID - ID Pengirim yang disediakan tidak valid. ID Pengirim bisa dalam format atau panjang yang salah.• INVALID_POOL_ID — ID Pool yang disediakan tidak valid. ID Pangkalan bisa dalam format yang salah atau bukan milik akun Anda.• SENDER_ID_NOT_SUPPORTED_FOR_DESTINATION - Negara tujuan tidak mendukung ID Pengirim. Anda harus

Atribut	Deskripsi
	<p>menggunakan nomor telepon atau identitas asal lain untuk mengirim.</p> <ul style="list-style-type: none">INVALID_PHONE_NUMBER - Nomor telepon asal yang disediakan tidak valid. Nomor telepon bisa dalam format atau panjang yang salah.
iso_country_code	Negara yang terkait dengan nomor telepon penerima, ditampilkan dalam format ISO 3166-1 alpha-2.
treatment_id	ID perlakuan pesan, jika pesan dikirim dalam kampanye A/B.
treatment_id	Jika pesan dikirim menggunakan kampanye pengujian A/B, nilai ini menunjukkan nomor perlakuan pesan. Untuk pesan SMS transaksional, nilai ini adalah 0.
number_of_message_parts	<p>Jumlah bagian pesan yang dibuat Amazon Pinpoint untuk mengirim pesan.</p> <p>Jika Anda mengirim pesan yang melebihi batas ini, Amazon Pinpoint secara otomatis membagi pesan menjadi bagian-bagian yang lebih kecil. Kami menagih Anda berdasarkan jumlah bagian pesan yang Anda kirim.</p>
message_id	ID unik yang dihasilkan Amazon Pinpoint saat menerima pesan.

Atribut	Deskripsi
<code>message_type</code>	Jenis pesan. Nilai yang mungkin adalah Promosi dan Transaksional. Anda menentukan nilai ini saat membuat kampanye, atau saat Anda mengirim pesan transaksional dengan menggunakan SendMessage operasi di Amazon Pinpoint API.
<code>campaign_id</code>	ID unik kampanye Amazon Pinpoint yang mengirim pesan.

Klien

Menyertakan informasi tentang klien aplikasi yang diinstal pada perangkat yang melaporkan kejadian tersebut.

Atribut	Deskripsi
<code>client_id</code>	<p>Untuk peristiwa yang dihasilkan oleh aplikasi, nilai ini adalah ID unik klien aplikasi yang diinstal pada perangkat. ID ini secara otomatis dihasilkan oleh AWS Mobile SDK for iOS dan AWS Mobile SDK for Android.</p> <p>Untuk peristiwa yang dihasilkan saat Anda mengirim kampanye dan pesan transaksional, nilai ini sama dengan ID titik akhir tempat Anda mengirim pesan.</p>
<code>cognito_id</code>	ID unik yang ditetapkan ke klien aplikasi di pangkalan identitas Amazon Cognito yang digunakan oleh aplikasi Anda.


Perangkat

Termasuk informasi tentang perangkat yang melaporkan kejadian tersebut.

Atribut	Deskripsi
<code>locale</code>	Perangkat lokal.
<code>make</code>	Perangkat membuat, seperti Apple atau Samsung.
<code>model</code>	Model perangkat, seperti iPhone.
<code>platform</code>	Platform perangkat, seperti ios atau android.

Metrik

Termasuk informasi tentang metrik yang terkait dengan peristiwa tersebut.

Atribut	Deskripsi
<code>price_in_millicents_usd</code>	<p>Jumlah yang kami bebaskan kepada Anda untuk mengirim pesan. Harga ini ditampilkan dalam seperseribu sen Amerika Serikat. Misalnya, jika nilai atribut ini 645, maka kami menagih Anda 0,645¢ untuk mengirim pesan ($645/1000 = 0,645¢ = 0,00645$ USD).</p> <div data-bbox="829 1266 1508 1530"><p> Note</p><p>Properti ini tidak muncul untuk pesan dengan <code>event_type</code> <code>_SMS.BUFFERED</code>.</p></div>

Menanyakan data analitik Amazon Pinpoint

Selain menggunakan halaman analitik di konsol Amazon Pinpoint, Anda dapat menggunakan Amazon Pinpoint Analytics API untuk menanyakan data analitik untuk subset metrik standar yang memberikan wawasan tentang tren yang terkait dengan keterlibatan pengguna, penjangkauan kampanye, dan lainnya. Metrik ini, juga disebut sebagai indikator kinerja utama (KPI), adalah nilai terukur yang dapat membantu Anda memantau dan menilai kinerja proyek, kampanye, dan perjalanan Anda.

Jika Anda menggunakan API untuk menanyakan data analitik, Anda dapat menganalisis data dengan menggunakan alat pelaporan pilihan Anda, tanpa harus masuk ke konsol Amazon Pinpoint atau menganalisis data peristiwa mentah dari sumber seperti aliran Amazon Kinesis. Misalnya, Anda dapat membuat dasbor khusus yang menampilkan hasil kampanye mingguan atau memberikan analisis mendalam tentang tarif pengiriman untuk kampanye Anda.

Anda dapat melakukan kueri data menggunakan Amazon Pinpoint REST API, the AWS Command Line Interface (AWS CLI), atau SDK. AWS Untuk menanyakan data, Anda mengirim permintaan ke Amazon Pinpoint API dan menggunakan parameter yang didukung untuk menentukan data yang Anda inginkan dan filter apa pun yang ingin Anda terapkan. Setelah mengirimkan kueri, Amazon Pinpoint mengembalikan hasil kueri dalam respons JSON. Anda kemudian dapat meneruskan hasilnya ke layanan atau aplikasi lain untuk analisis, penyimpanan, atau pelaporan yang lebih dalam.

Metrik yang didukung

Amazon Pinpoint menyediakan akses terprogram ke data analitik untuk beberapa jenis metrik standar:

- **Metrik aplikasi** — Metrik ini memberikan wawasan tentang tren untuk semua kampanye dan pesan transaksional yang terkait dengan proyek. Misalnya, Anda dapat menggunakan metrik aplikasi untuk mendapatkan rincian jumlah pesan yang dibuka oleh penerima untuk setiap kampanye yang terkait dengan proyek. Untuk mengakses data metrik aplikasi, gunakan sumber daya [metrik Aplikasi](#) dari Amazon Pinpoint API.
- **Metrik kampanye** — Metrik ini memberikan wawasan tentang kinerja masing-masing kampanye. Misalnya, Anda dapat menggunakan metrik kampanye untuk menentukan jumlah titik akhir pesan kampanye yang dikirim. Untuk mengakses data metrik kampanye, gunakan sumber daya [metrik Kampanye](#) dari Amazon Pinpoint API.

- **Metrik keterlibatan perjalanan** — Metrik ini memberikan wawasan tentang kinerja perjalanan individu. Misalnya, Anda dapat menggunakan metrik keterlibatan perjalanan untuk mendapatkan rincian jumlah pesan yang dibuka oleh peserta di setiap aktivitas perjalanan. Untuk mengakses data untuk metrik keterlibatan perjalanan, gunakan sumber daya [metrik keterlibatan Journey](#) dari Amazon Pinpoint API.
- **Metrik eksekusi perjalanan** — Metrik ini memberikan wawasan tentang tren partisipasi untuk perjalanan individu. Misalnya, Anda dapat menggunakan metrik eksekusi perjalanan untuk menentukan berapa banyak peserta yang melanjutkan aktivitas dalam perjalanan. Untuk mengakses data untuk metrik eksekusi perjalanan, gunakan sumber daya [Metrik Eksekusi Perjalanan](#) dari Amazon Pinpoint API.
- **Metrik eksekusi aktivitas perjalanan** — Metrik ini memberikan wawasan tentang tren partisipasi untuk aktivitas individu dalam suatu perjalanan. Misalnya, Anda dapat menggunakan metrik eksekusi aktivitas perjalanan untuk menentukan jumlah peserta yang menyelesaikan aktivitas. Untuk mengakses data untuk metrik eksekusi aktivitas perjalanan, gunakan sumber daya [metrik eksekusi aktivitas Journey](#) dari Amazon Pinpoint API.

Untuk daftar lengkap metrik standar yang dapat Anda kueri secara terprogram, lihat. [Metrik standar](#)

Amazon Pinpoint secara otomatis mengumpulkan dan menggabungkan data untuk semua metrik yang didukung, untuk semua proyek, kampanye, dan perjalanan Anda. Selain itu, data diperbarui terus menerus, menghasilkan kerangka waktu latensi data yang dibatasi sekitar dua jam. Namun, perhatikan bahwa mungkin ada latensi data tambahan untuk metrik tertentu. Ini karena data untuk beberapa metrik didasarkan pada informasi yang kami terima dari penyedia email penerima. Beberapa penyedia mengirimkan informasi ini kepada kami segera, sementara yang lain mengirimkannya lebih jarang.

Amazon Pinpoint menyimpan data selama 90 hari. Untuk menyimpan data selama lebih dari 90 hari atau mengakses data analitik mentah secara real time, Anda dapat mengonfigurasi project Amazon Pinpoint untuk mengalirkan data peristiwa ke Amazon Kinesis Data Streams atau Amazon Data Firehose. Untuk informasi tentang mengonfigurasi aliran peristiwa, lihat. [Streaming acara Amazon Pinpoint ke Kinesis](#)

Dasar-dasar kueri

Untuk menanyakan data metrik, Anda mengirim get permintaan ke sumber daya metrik yang sesuai dari Amazon Pinpoint API. Dalam permintaan, Anda menentukan kueri dengan menggunakan parameter yang didukung untuk komponen kueri berikut:

- **Proyek** — Tentukan proyek dengan memberikan ID proyek sebagai nilai untuk `application-id` parameter. Parameter ini diperlukan untuk semua metrik.
- **Kampanye** — Tentukan kampanye dengan memberikan ID kampanye sebagai nilai untuk `campaign-id` parameter. Parameter ini hanya diperlukan untuk metrik kampanye.
- **Perjalanan** — Tentukan perjalanan dengan memberikan ID perjalanan sebagai nilai untuk `journey-id` parameter. Parameter ini hanya diperlukan untuk metrik keterlibatan dan eksekusi perjalanan, dan metrik eksekusi aktivitas perjalanan.
- **Aktivitas perjalanan** — Tentukan aktivitas perjalanan dengan memberikan ID aktivitas perjalanan sebagai nilai `journey-activity-id` parameter. Parameter ini hanya diperlukan untuk metrik eksekusi aktivitas perjalanan.
- **Rentang tanggal** - Untuk secara opsional memfilter data berdasarkan rentang tanggal, berikan tanggal dan waktu pertama dan terakhir dari rentang tanggal dengan menggunakan parameter waktu mulai dan akhir yang didukung. Nilai harus dalam format ISO 8601 yang diperluas dan menggunakan Coordinated Universal Time (UTC) —misalnya, `2019-07-19T20:00:00Z` untuk 8:00 PM UTC 19 Juli 2019.

Rentang tanggal bersifat inklusif dan harus dibatasi hingga 31 hari kalender atau lebih sedikit. Selain itu, tanggal dan waktu pertama harus kurang dari 90 hari dari hari ini. Jika Anda tidak menentukan rentang tanggal, Amazon Pinpoint mengembalikan data untuk 31 hari kalender sebelumnya. Parameter rentang tanggal didukung oleh semua metrik kecuali metrik eksekusi perjalanan dan metrik eksekusi aktivitas perjalanan.

- **Metrik** — Tentukan metrik dengan memberikan nama metrik sebagai nilai untuk `kpi-name` parameter. Nilai ini menggambarkan metrik terkait dan terdiri dari dua atau lebih istilah, yang terdiri dari karakter alfanumerik huruf kecil, dipisahkan oleh tanda hubung. Contohnya adalah `email-open-rate` dan `successful-delivery-rate`. Parameter ini diperlukan untuk semua metrik kecuali metrik eksekusi perjalanan dan metrik eksekusi aktivitas perjalanan. Untuk daftar lengkap metrik yang didukung dan `kpi-name` nilai yang akan digunakan untuk masing-masing metrik, lihat [Metrik standar](#).

Setelah mengirim kueri, Amazon Pinpoint mengembalikan hasil kueri dalam respons JSON. Dalam tanggapannya, struktur hasil bervariasi tergantung pada metrik yang Anda tanyakan.

Beberapa metrik hanya memberikan satu nilai—misalnya, jumlah pesan yang dikirimkan oleh kampanye. Metrik lain memberikan beberapa nilai dan biasanya mengelompokkan nilai tersebut berdasarkan bidang yang relevan—misalnya, jumlah pesan yang dikirimkan oleh setiap proses kampanye, dikelompokkan berdasarkan kampanye yang dijalankan. Jika metrik menyediakan dan

mengelompokkan beberapa nilai, respons JSON menyertakan bidang yang menunjukkan bidang mana yang digunakan untuk mengelompokkan data. Untuk mempelajari lebih lanjut tentang struktur hasil kueri, lihat [Menggunakan hasil kueri](#).

Kebijakan IAM untuk kueri data analitik Amazon Pinpoint

Dengan menggunakan API Amazon Pinpoint, Anda dapat melakukan kueri data analitik untuk subset metrik standar, yang juga disebut sebagai indikator kinerja utama (KPI) yang berlaku untuk proyek, kampanye, dan perjalanan Amazon Pinpoint. Metrik ini dapat membantu Anda memantau dan menilai kinerja proyek, kampanye, dan perjalanan.

Untuk mengelola akses ke data ini, Anda dapat membuat AWS Identity and Access Management (IAM) kebijakan yang menentukan izin untuk peran IAM atau pengguna yang berwenang untuk mengakses data. Untuk mendukung kontrol granular akses ke data ini, Amazon Pinpoint menyediakan beberapa tindakan berbeda yang dapat Anda tetapkan dalam kebijakan IAM. Ada tindakan berbeda untuk melihat data analitik di konsol Amazon Pinpoint (`mobiletargeting:GetReports`), dan ada tindakan lain untuk mengakses data analitik secara terprogram dengan menggunakan API Amazon Pinpoint.

Untuk membuat kebijakan IAM yang mengelola akses ke data analitik, Anda dapat menggunakan AWS Management Console, yang AWS CLI, atau API IAM. Perhatikan bahwa Editor visual tab pada AWS Management Console saat ini tidak menyertakan tindakan untuk melihat atau melakukan kueri data analitik Amazon Pinpoint. Namun, Anda dapat menambahkan tindakan yang diperlukan untuk kebijakan IAM secara manual dengan menggunakan JSON tab di konsol.

Misalnya, kebijakan berikut ini memungkinkan akses terprogram ke semua data analitik untuk semua proyek, kampanye, dan perjalanan Anda di semua AWS Wilayah:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "QueryAllAnalytics",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:GetApplicationDateRangeKpi",
        "mobiletargeting:GetCampaignDateRangeKpi",
        "mobiletargeting:GetJourneyDateRangeKpi",
        "mobiletargeting:GetJourneyExecutionMetrics",
        "mobiletargeting:GetJourneyExecutionActivityMetrics"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
      "arn:aws:mobiletargeting:*:accountId:apps/*/kpis/*",
      "arn:aws:mobiletargeting:*:accountId:apps/*/campaigns/*/kpis/*",
      "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/kpis/*",
      "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/execution-
metrics",
      "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/activities/*/
execution-metrics"
    ]
  }
]
}

```

Di mana *accountId* adalah AWS ID akun.

Namun, sebagai praktik terbaik, Anda harus membuat kebijakan yang mengikuti prinsip hak istimewa paling rendah. Dengan kata lain, Anda harus membuat kebijakan yang mencakup hanya izin yang diperlukan untuk melakukan tugas tertentu. Untuk mendukung praktik ini dan menerapkan kontrol yang lebih terperinci, Anda dapat membatasi akses terprogram ke data analitik hanya untuk proyek tertentu secara spesifik AWS Wilayah, misalnya:

```

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

```

```

        "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/
activities/*/execution-metrics"
    ]
}
]
}

```

Dengan:

- *daerah* adalah nama AWS Wilayah yang menjadi tuan rumah proyek.
- *accountId* adalah AWS ID akun.
- *projectId* adalah pengenal proyek yang ingin Anda berikan akses.

Demikian pula, kebijakan contoh berikut memungkinkan akses terprogram ke data analitik hanya untuk kampanye tertentu:

```

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

```

Dengan:

- *daerah* adalah nama AWS Wilayah yang menjadi tuan rumah proyek.
- *accountId* adalah Akun AWS ID.
- *projectId* adalah pengenal untuk proyek yang terkait dengan kampanye.
- *campaignId* adalah pengenal untuk kampanye yang ingin Anda berikan akses.

Dan contoh kebijakan berikut memungkinkan akses terprogram ke semua data analitik, baik data keterlibatan dan eksekusi, untuk perjalanan tertentu dan aktivitas yang terdiri dari perjalanan tersebut:


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

Dengan:

- *daerah* adalah nama AWS Wilayah yang menjadi tuan rumah proyek.
- *accountId* adalah AWS ID akun.
- *projectId* adalah pengenal untuk proyek yang terkait dengan perjalanan.
- *journeyId* adalah pengenal untuk perjalanan yang ingin Anda berikan akses.

Untuk daftar lengkap tindakan API Amazon Pinpoint yang dapat Anda gunakan dalam kebijakan IAM, lihat [Tindakan Amazon Pinpoint untuk kebijakan IAM](#). Untuk informasi rinci tentang membuat dan mengelola kebijakan IAM, lihat [Panduan Pengguna IAM](#).

Metrik analitik Amazon Pinpoint standar

Anda dapat menggunakan Amazon Pinpoint Analytics API untuk menanyakan data analitik untuk subset metrik standar yang berlaku untuk proyek, kampanye, dan perjalanan Amazon Pinpoint. Metrik

ini, juga disebut sebagai indikator kinerja utama (KPI), adalah nilai terukur yang dapat membantu Anda memantau dan menilai kinerja proyek, kampanye, dan perjalanan.

Amazon Pinpoint menyediakan akses terprogram ke data analitik untuk beberapa jenis metrik standar:

- **Metrik aplikasi** — Metrik ini memberikan wawasan tentang tren untuk semua kampanye dan pesan transaksional yang terkait dengan proyek, juga disebut sebagai aplikasi. Misalnya, Anda dapat menggunakan metrik aplikasi untuk mendapatkan rincian jumlah pesan yang dibuka oleh penerima untuk setiap kampanye yang terkait dengan proyek.
- **Metrik kampanye** — Metrik ini memberikan wawasan tentang kinerja masing-masing kampanye. Misalnya, Anda dapat menggunakan metrik kampanye untuk menentukan jumlah titik akhir pesan kampanye yang dikirim atau berapa banyak pesan yang dikirim ke titik akhir.
- **Metrik keterlibatan perjalanan** — Metrik ini memberikan wawasan tentang kinerja perjalanan individu. Misalnya, Anda dapat menggunakan metrik keterlibatan perjalanan untuk mendapatkan rincian jumlah pesan yang dibuka oleh peserta di setiap aktivitas perjalanan.
- **Metrik eksekusi perjalanan** — Metrik ini memberikan wawasan tentang tren partisipasi untuk perjalanan individu. Misalnya, Anda dapat menggunakan metrik eksekusi perjalanan untuk menentukan berapa banyak peserta yang memulai perjalanan.
- **Metrik eksekusi aktivitas perjalanan** — Metrik ini memberikan wawasan tentang tren partisipasi untuk aktivitas individu dalam suatu perjalanan. Misalnya, Anda dapat menggunakan metrik eksekusi aktivitas perjalanan untuk menentukan berapa banyak peserta yang memulai suatu aktivitas dan berapa banyak peserta yang menyelesaikan setiap jalur dalam suatu aktivitas.

Topik di bagian ini mencantumkan dan menjelaskan metrik individual yang dapat Anda kueri untuk setiap jenis metrik.

Topik

- [Metrik aplikasi untuk kampanye](#)
- [Metrik aplikasi untuk pesan email transaksional](#)
- [Metrik aplikasi untuk pesan SMS transaksional](#)
- [Metrik kampanye](#)
- [Metrik keterlibatan perjalanan](#)
- [Metrik eksekusi perjalanan](#)
- [Metrik eksekusi aktivitas perjalanan](#)

- [Metrik eksekusi perjalanan dan kampanye](#)

Metrik aplikasi untuk kampanye

Tabel berikut mencantumkan dan menjelaskan metrik aplikasi standar yang dapat Anda kueri untuk menilai kinerja semua kampanye yang terkait dengan proyek Amazon Pinpoint. Untuk menanyakan data metrik ini, gunakan sumber daya [metrik Aplikasi](#) dari Amazon Pinpoint API. Kolom kpi-name dalam tabel menunjukkan nilai yang akan digunakan untuk kpi-name parameter dalam kueri.

Metrik	Nama KPI	Deskripsi
Tingkat pengiriman	successful-delivery-rate	<p>Untuk semua kampanye yang terkait dengan proyek, persentase pesan yang dikirim ke penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim oleh semua kampanye untuk proyek dan dikirim ke penerima, dibagi dengan jumlah pesan yang dikirim oleh semua kampanye tersebut.</p>
Tingkat pengiriman, dikelompokkan berdasarkan tanggal	successful-delivery-rate-grouped-by-date	<p>Untuk semua kampanye yang terkait dengan proyek, persentase pesan yang dikirim ke penerima, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim oleh semua kampanye untuk proyek dan dikirim ke penerima, dibagi dengan</p>

Metrik	Nama KPI	Deskripsi
		<p>jumlah pesan yang dikirim oleh semua kampanye tersebut, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Tingkat buka email	email-open-rate	<p>Untuk semua kampanye yang terkait dengan proyek, persentase pesan email yang dibuka oleh penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan email yang dikirim oleh semua kampanye untuk proyek dan dibuka oleh penerima, dibagi dengan jumlah pesan email yang dikirim oleh semua kampanye tersebut dan dikirim ke penerima.</p>

Metrik	Nama KPI	Deskripsi
Rasio buka email, dikelompokkan berdasarkan kampanye	<code>email-open-rate-grouped-by-campaign</code>	<p>Untuk setiap kampanye yang terkait dengan proyek, persentase pesan email yang dibuka oleh penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan email yang dikirim oleh kampanye dan dibuka oleh penerima, dibagi dengan jumlah pesan email yang dikirim oleh kampanye dan dikirim ke penerima.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID kampanye (<code>CampaignId</code>), yang merupakan string yang mengidentifikasi kampanye secara unik.</p>
Pengiriman titik akhir	<code>unique-deliveries</code>	<p>Untuk semua kampanye yang terkait dengan proyek, jumlah titik akhir unik yang dikirimkan pesan.</p>
Pengiriman titik akhir, dikelompokkan berdasarkan kampanye	<code>unique-deliveries-grouped-by-campaign</code>	<p>Untuk setiap kampanye yang terkait dengan proyek, jumlah titik akhir unik yang dikirimkan pesan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID kampanye (<code>CampaignId</code>), yang merupakan string yang mengidentifikasi kampanye secara unik.</p>

Metrik	Nama KPI	Deskripsi
Pengiriman titik akhir, dikelompokkan berdasarkan tanggal	unique-deliveries-grouped-by-date	<p>Untuk semua kampanye yang terkait dengan proyek, jumlah titik akhir unik yang dikirimkan pesan, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Pesan terkirim, dikelompokkan berdasarkan kampanye	successful-deliveries-grouped-by-campaign	<p>Untuk setiap kampanye yang terkait dengan proyek, jumlah pesan yang dikirimkan ke penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim oleh kampanye, dikurangi jumlah pesan yang dikirim oleh kampanye dan tidak dapat dikirim ke penerima karena pantulan keras.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID kampanye (CampaignId), yang merupakan string yang mengidentifikasi kampanye secara unik.</p>

Metrik	Nama KPI	Deskripsi
Dorong tingkat buka	push-open-rate	<p>Untuk semua kampanye yang terkait dengan proyek, persentase pemberitahuan push yang dibuka oleh penerima.</p> <p>Metrik ini dihitung sebagai jumlah pemberitahuan push yang dikirim oleh semua kampanye untuk proyek dan dibuka oleh penerima, dibagi dengan jumlah pemberitahuan push yang dikirim oleh semua kampanye tersebut dan dikirimkan ke penerima.</p>

Metrik	Nama KPI	Deskripsi
Push open rate, dikelompokkan berdasarkan kampanye	push-open-rate-grouped-by-campaign	<p>Untuk setiap kampanye yang terkait dengan proyek, persentase pemberitahuan push yang dibuka oleh penerima.</p> <p>Metrik ini dihitung sebagai jumlah pemberitahuan push yang dikirim oleh kampanye dan dibuka oleh penerima, dibagi dengan jumlah pemberitahuan push yang dikirim oleh kampanye dan dikirim ke penerima.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID kampanye (CampaignId), yang merupakan string yang mengidentifikasi kampanye secara unik.</p>

Metrik aplikasi untuk pesan email transaksional

Tabel berikut mencantumkan dan menjelaskan metrik aplikasi standar yang dapat Anda kueri untuk memantau tren semua pesan email transaksional yang terkait dengan proyek Amazon Pinpoint. Untuk menanyakan data metrik ini, gunakan sumber daya [metrik Aplikasi](#) dari Amazon Pinpoint API. Kolom kpi-name dalam tabel menunjukkan nilai yang akan digunakan untuk kpi-name parameter dalam kueri.

Perhatikan bahwa metrik ini tidak menyediakan data tentang pesan email yang dikirim oleh kampanye. Mereka menyediakan data tentang pesan email transaksional saja. Untuk menanyakan data pesan yang dikirim oleh satu atau beberapa kampanye, gunakan [metrik kampanye atau metrik aplikasi untuk kampanye](#).

Metrik	Nama KPI	Deskripsi
Klik	<code>txn-emails-clicked</code>	Berapa kali penerima mengklik tautan dalam pesan. Jika satu penerima mengklik beberapa tautan dalam pesan, atau mengklik tautan yang sama lebih dari sekali, setiap klik disertakan dalam hitungan.
Klik, dikelompokkan berdasarkan tanggal	<code>txn-emails-clicked-grouped-by-date</code>	<p>Berapa kali penerima mengklik tautan dalam pesan, untuk setiap hari dalam rentang tanggal yang ditentukan. Jika satu penerima mengklik beberapa tautan dalam pesan, atau mengklik tautan yang sama lebih dari sekali, setiap klik disertakan dalam hitungan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Tingkat keluhan	<code>txn-emails-complaint-rate</code>	<p>Persentase pesan yang dilaporkan oleh penerima sebagai email yang tidak diminta atau tidak diinginkan.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dilaporkan oleh penerima sebagai email yang tidak diminta atau tidak diinginkan, dibagi dengan jumlah pesan yang dikirim.</p>

Metrik	Nama KPI	Deskripsi
Tingkat keluhan, dikelompokkan berdasarkan tanggal	<code>txn-emails-complaint-rate-grouped-by-date</code>	<p>Persentase pesan yang dilaporkan oleh penerima sebagai email yang tidak diminta atau tidak diinginkan, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dilaporkan oleh penerima sebagai email yang tidak diminta atau tidak diinginkan, dibagi dengan jumlah pesan yang dikirim, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Aduan	<code>txn-emails-with-complaints</code>	Jumlah pesan yang dilaporkan oleh penerima sebagai email yang tidak diminta atau tidak diinginkan.

Metrik	Nama KPI	Deskripsi
Keluhan, dikelompokkan berdasarkan tanggal	<code>txn-emails-with-complaints-grouped-by-date</code>	<p>Jumlah pesan yang dilaporkan oleh penerima sebagai email yang tidak diminta atau tidak diinginkan, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Pengiriman	<code>txn-emails-delivered</code>	<p>Jumlah pesan yang dikirimkan ke penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim, dikurangi jumlah pesan yang tidak dapat dikirim karena pantulan lunak atau keras atau karena ditolak. Pesan ditolak jika Amazon Pinpoint menentukan bahwa pesan tersebut berisi malware. Amazon Pinpoint tidak mencoba mengirim pesan yang ditolak.</p>

Metrik	Nama KPI	Deskripsi
Pengiriman, dikelompokkan berdasarkan tanggal	txn-emails-delivered-grouped-by-date	<p>Jumlah pesan yang dikirim ke penerima, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim, dikurangi jumlah pesan yang tidak dapat dikirim karena pantulan lunak atau keras atau karena ditolak, untuk setiap hari dalam rentang tanggal yang ditentukan. Pesan ditolak jika Amazon Pinpoint menentukan bahwa pesan tersebut berisi malware. Amazon Pinpoint tidak mencoba mengirim pesan yang ditolak.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Tingkat pengiriman	txn-emails-delivery-rate	<p>Persentase pesan yang dikirim ke penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim dan dikirim ke penerima, dibagi dengan jumlah pesan yang dikirim.</p>

Metrik	Nama KPI	Deskripsi
Tingkat pengiriman, dikelompokkan berdasarkan tanggal	<code>txn-emails-delivery-rate-grouped-by-date</code>	<p>Persentase pesan yang dikirim ke penerima, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim dan dikirim ke penerima, dibagi dengan jumlah pesan yang dikirim, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
pentalan keras	<code>txn-emails-hard-bounced</code>	<p>Jumlah pesan yang tidak dapat dikirim ke penerima karena pantulan keras. Hard bounce terjadi jika masalah persisten mencegah pesan dikirimkan—misalnya, jika alamat email penerima tidak ada.</p>

Metrik	Nama KPI	Deskripsi
Pantulan keras, dikelompokkan berdasarkan tanggal	<code>txn-emails-hard-bounced-grouped-by-date</code>	<p>Jumlah pesan yang tidak dapat dikirim ke penerima karena pantulan keras, untuk setiap hari dalam rentang tanggal yang ditentukan. Hard bounce terjadi jika masalah persisten mencegah pesan dikirimkan—misalnya, jika alamat email penerima tidak ada.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Membuka	<code>txn-emails-opened</code>	Jumlah pesan yang dibuka oleh penerima.
Dibuka, dikelompokkan berdasarkan tanggal	<code>txn-emails-opened-grouped-by-date</code>	<p>Jumlah pesan yang dibuka oleh penerima, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Mengirim	<code>txn-emails-sent</code>	Jumlah pesan yang dikirim.

Metrik	Nama KPI	Deskripsi
Mengirim, dikelompokkan berdasarkan tanggal	<code>txn-emails-sent-grouped-by-date</code>	<p>Jumlah pesan yang dikirim, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Pantulan lembut	<code>txn-emails-soft-bounced</code>	<p>Jumlah pesan yang tidak dapat dikirim ke penerima karena soft bounce. Soft bounce terjadi jika masalah sementara mencegah pesan dikirimkan—misalnya, jika kotak masuk penerima penuh atau server penerima sementara tidak tersedia.</p>

Metrik	Nama KPI	Deskripsi
Pantulan lunak, dikelompokkan berdasarkan tanggal	txn-emails-soft-bounced-grouped-by-date	<p>Jumlah pesan yang tidak dapat dikirim ke penerima karena soft bounce, untuk setiap hari dalam rentang tanggal yang ditentukan. Soft bounce terjadi jika masalah sementara mencegah pesan dikirimkan—misalnya, jika kotak masuk penerima penuh atau server penerima sementara tidak tersedia.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Acara klik pengguna yang unik	txn-emails-unique-clicks	<p>Jumlah penerima unik (endpoint) yang mengklik link dalam pesan.</p> <p>Berbeda dengan metrik Klik, metrik ini melaporkan jumlah penerima unik yang mengklik tautan, bukan jumlah peristiwa klik yang terjadi. Misalnya, jika satu penerima mengklik beberapa tautan dalam pesan yang sama, atau mengklik tautan yang sama lebih dari sekali, metrik ini hanya melaporkan satu peristiwa klik untuk penerima tersebut.</p>

Metrik	Nama KPI	Deskripsi
Acara klik pengguna unik, dikelompokkan berdasarkan tanggal	txn-emails-unique-clicks-grouped-by-date	<p>Jumlah penerima unik (titik akhir) yang mengklik tautan dalam pesan, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Berbeda dengan Klik, dikelompokkan berdasarkan metrik tanggal, metrik ini melaporkan jumlah penerima unik yang mengklik tautan, bukan jumlah peristiwa klik yang terjadi. Misalnya, jika satu penerima mengklik beberapa tautan dalam pesan yang sama, atau mengklik tautan yang sama lebih dari sekali, metrik ini hanya melaporkan satu peristiwa klik untuk penerima tersebut.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>

Metrik	Nama KPI	Deskripsi
Acara terbuka pengguna yang unik	txn-emails-unique-opens	<p>Jumlah penerima unik (endpoint) yang membuka pesan.</p> <p>Berbeda dengan metrik Buka, metrik ini melaporkan jumlah penerima unik yang membuka pesan, bukan jumlah peristiwa terbuka yang terjadi. Misalnya, jika satu penerima membuka pesan yang sama beberapa kali, metrik ini hanya melaporkan satu peristiwa terbuka untuk penerima tersebut.</p>

Metrik	Nama KPI	Deskripsi
Acara terbuka pengguna unik, dikelompokkan berdasarkan tanggal	txn-emails-unique-opens-grouped-by-date	<p>Jumlah penerima unik (endpoint) yang membuka pesan, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Berbeda dengan Buka, dikelompokkan berdasarkan metrik tanggal, metrik ini melaporkan jumlah penerima unik yang membuka pesan, bukan jumlah peristiwa terbuka yang terjadi. Misalnya, jika satu penerima membuka pesan yang sama beberapa kali, metrik ini hanya melaporkan satu peristiwa terbuka untuk penerima tersebut.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>

Metrik aplikasi untuk pesan SMS transaksional

Tabel berikut mencantumkan dan menjelaskan metrik aplikasi standar yang dapat Anda kueri untuk memantau tren semua pesan SMS transaksional yang terkait dengan proyek Amazon Pinpoint. Untuk menanyakan data metrik ini, gunakan sumber daya [metrik Aplikasi](#) dari Amazon Pinpoint API. Kolom kpi-name dalam tabel menunjukkan nilai yang akan digunakan untuk kpi-name parameter dalam kueri.

Perhatikan bahwa metrik ini tidak menyediakan data tentang pesan SMS yang dikirim oleh kampanye. Mereka menyediakan data tentang pesan SMS transaksional saja. Untuk menanyakan

data pesan yang dikirim oleh satu atau beberapa kampanye, gunakan [metrik kampanye atau metrik aplikasi untuk kampanye](#).

Metrik	Nama KPI	Deskripsi
Harga rata-rata per pesan, dikelompokkan berdasarkan negara	<code>txn-sms-average-price-grouped-by-country</code>	<p>Biaya rata-rata pengiriman setiap pesan, untuk setiap negara atau wilayah tempat pesan dikirim. Harganya ditunjukkan dalam seperseribu sen Amerika Serikat. Misalnya, jika nilai atribut ini adalah 645, maka kami menagih Anda 0,645¢ untuk mengirim pesan ($645/1000 = 0,645¢ = \\$0,00645$).</p> <p>Metrik ini dihitung sebagai total biaya semua pesan yang dikirim ke penerima di setiap negara atau wilayah, dibagi dengan jumlah pesan yang dikirim ke penerima di masing-masing negara dan wilayah tersebut.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan negara atau wilayah, dalam format ISO 3166-1 alpha-2.</p>
Harga rata-rata per bagian pesan, dikelompokkan berdasarkan negara	<code>txn-sms-average-price-by-parts-grouped-by-country</code>	Biaya rata-rata pengiriman setiap bagian pesan, untuk setiap negara atau wilayah tempat pesan dikirim. Bagian pesan adalah bagian dari pesan SMS. Harganya ditunjukkan dalam seperseri

Metrik	Nama KPI	Deskripsi
		<p>bu sen Amerika Serikat. Misalnya, jika nilai atribut ini adalah 645, maka kami menagih Anda 0,645¢ untuk mengirim pesan ($645/1000 = 0,645¢ = \\$0,00645$).</p> <p>Metrik ini dihitung sebagai total biaya semua bagian pesan yang dikirim ke penerima di setiap negara atau wilayah, dibagi dengan jumlah bagian pesan yang dikirim ke penerima di masing-masing negara dan wilayah tersebut.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan negara atau wilayah, dalam format ISO 3166-1 alpha-2.</p>
Pengiriman	txn-sms-delivered	Jumlah pesan yang dikirimkan ke penerima.
Pengiriman, dikelompokkan berdasarkan negara	txn-sms-delivered-grouped-by-country	<p>Jumlah pesan yang dikirimkan ke penerima, untuk setiap negara atau wilayah tempat pesan dikirim.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan negara atau wilayah, dalam format ISO 3166-1 alpha-2.</p>

Metrik	Nama KPI	Deskripsi
Pengiriman, dikelompokkan berdasarkan tanggal	<code>txn-sms-delivered-grouped-by-date</code>	<p>Jumlah pesan yang dikirim ke penerima, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Kesalahan pengiriman	<code>txn-sms-error-distribution</code>	<p>Berapa kali kesalahan terjadi saat mencoba pengiriman pesan, untuk setiap jenis kesalahan yang terjadi.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan kode kesalahan, untuk setiap jenis kesalahan yang terjadi.</p>
Tingkat pengiriman	<code>txn-sms-delivery-rate</code>	<p>Persentase pesan yang dikirim ke penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim dan dikirim ke penerima, dibagi dengan jumlah pesan yang dikirim.</p>

Metrik	Nama KPI	Deskripsi
Tingkat pengiriman, dikelompokkan berdasarkan tanggal	<code>txn-sms-delivery-rate-grouped-by-date</code>	<p>Persentase pesan yang dikirim ke penerima, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim dan dikirim ke penerima, dibagi dengan jumlah pesan yang dikirim, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Bagian pesan terkirim	<code>txn-sms-delivered-by-parts</code>	<p>Jumlah bagian pesan yang dikirimkan. Bagian pesan adalah bagian dari pesan SMS. Jika pesan SMS berisi lebih banyak karakter daripada yang diizinkan oleh protokol SMS, Amazon Pinpoint membagi pesan menjadi bagian pesan sebanyak yang diperlukan untuk mengirim pesan ke penerima.</p>

Metrik	Nama KPI	Deskripsi
Bagian pesan dikirim, dikelompokkan berdasarkan negara	<code>txn-sms-delivered-by-parts-grouped-by-country</code>	<p>Jumlah bagian pesan yang dikirimkan, untuk setiap negara atau wilayah tempat pesan dikirim. Bagian pesan adalah bagian dari pesan SMS.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan negara atau wilayah, dalam format ISO 3166-1 alpha-2.</p>
Bagian pesan terkirim	<code>txn-sms-sent-by-parts</code>	<p>Jumlah bagian pesan yang dikirim. Bagian pesan adalah bagian dari pesan SMS. Jika pesan SMS berisi lebih banyak karakter daripada yang diizinkan oleh protokol SMS, Amazon Pinpoint membagi pesan menjadi bagian pesan sebanyak yang diperlukan untuk mengirim pesan ke penerima.</p>
Bagian pesan dikirim, dikelompokkan berdasarkan negara	<code>txn-sms-sent-by-parts-grouped-by-country</code>	<p>Jumlah bagian pesan yang dikirim, untuk setiap negara atau wilayah tempat pesan dikirim. Bagian pesan adalah bagian dari pesan SMS.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan negara atau wilayah, dalam format ISO 3166-1 alpha-2.</p>
Pesan terkirim	<code>txn-sms-sent</code>	Jumlah pesan yang dikirim.

Metrik	Nama KPI	Deskripsi
Pesan dikirim, dikelompokkan berdasarkan negara	<code>txn-sms-sent-grouped-by-country</code>	<p>Jumlah pesan yang dikirim, untuk setiap negara atau wilayah tempat pesan dikirim.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan negara atau wilayah, dalam format ISO 3166-1 alpha-2.</p>
Pesan dikirim, dikelompokkan berdasarkan tanggal	<code>txn-sms-sent-grouped-by-date</code>	<p>Jumlah pesan yang dikirim, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>
Total harga, dikelompokkan berdasarkan negara	<code>txn-sms-total-price-grouped-by-country</code>	<p>Total biaya pengiriman pesan, untuk setiap negara atau wilayah tempat pesan dikirim. Harganya ditunjukkan dalam seperseribu sen Amerika Serikat. Misalnya, jika nilai atribut ini adalah 645, maka kami menagih Anda 0,645¢ untuk mengirim pesan ($645/1000 = 0,645¢ = \\$0,00645$).</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan negara atau wilayah, dalam format ISO 3166-1 alpha-2.</p>

Metrik kampanye

Tabel berikut mencantumkan dan menjelaskan metrik kampanye standar yang dapat Anda kueri untuk menilai kinerja kampanye individual. Untuk menanyakan data metrik ini, gunakan sumber daya [metrik Kampanye](#) dari Amazon Pinpoint API. Kolom kpi-name dalam tabel menunjukkan nilai yang akan digunakan untuk kpi-name parameter dalam kueri Anda.

Metrik	Nama KPI	Deskripsi
Tingkat pentalan	hard-bounce-rate	<p>Untuk semua kampanye berjalan, persentase pesan email yang tidak dapat dikirim ke penerima. Metrik ini hanya mengukur hard bounce — yaitu, pesan di mana alamat email penerima memiliki masalah permanen yang mencegah pesan dikirim.</p> <p>Metrik ini dihitung sebagai jumlah pesan email yang dipantulkan yang dikirim oleh semua kampanye yang dijalankan, dibagi dengan jumlah pesan email yang dikirim oleh semua kampanye yang dijalankan.</p>
Rasio pentalan, dikelompokkan berdasarkan kampanye yang dijalankan	hard-bounce-rate-grouped-by-campaign-activity	<p>Untuk setiap kampanye yang dijalankan, persentase pesan email yang tidak dapat dikirimkan ke penerima. Metrik ini hanya mengukur hard bounce — yaitu, pesan di mana alamat email penerima memiliki masalah permanen yang mencegah pesan dikirim.</p>

Metrik	Nama KPI	Deskripsi
		<p>Metrik ini dihitung sebagai jumlah pesan email yang dipantulkan yang dikirim oleh kampanye yang dijalankan, dibagi dengan jumlah pesan email yang dikirim oleh kampanye yang dijalankan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>
Tingkat pengiriman	successful-delivery-rate	<p>Untuk semua kampanye berjalan, persentase pesan yang dikirim ke penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim oleh semua kampanye yang dijalankan dan dikirim ke penerima, dibagi dengan jumlah pesan yang dikirim oleh semua kampanye yang dijalankan.</p>

Metrik	Nama KPI	Deskripsi
Tingkat pengiriman, dikelompokkan berdasarkan kampanye	<code>successful-delivery-rate-grouped-by-campaign-activity</code>	<p>Untuk setiap kampanye yang dijalankan, persentase pesan yang dikirim ke penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim oleh kampanye yang dijalankan dan dikirim ke penerima, dibagi dengan jumlah pesan yang dikirim oleh kampanye yang dijalankan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>

Metrik	Nama KPI	Deskripsi
Tingkat pengiriman, dikelompokkan berdasarkan tanggal	successful-delivery-rate-grouped-by-date	<p>Untuk semua kampanye berjalan, persentase pesan yang dikirim ke penerima setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim oleh semua kampanye yang dijalankan dan dikirim ke penerima, dibagi dengan jumlah pesan yang dikirim oleh semua kampanye yang dijalankan, untuk setiap hari dalam rentang tanggal yang ditentukan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.</p>

Metrik	Nama KPI	Deskripsi
Tingkat buka email	<code>email-open-rate</code>	<p>Untuk semua kampanye berjalan, persentase pesan email yang dibuka oleh penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan email yang dikirim oleh semua kampanye yang dijalankan dan dibuka oleh penerima, dibagi dengan jumlah pesan email yang dikirim oleh semua kampanye yang dijalankan dan dikirim ke penerima.</p>

Metrik	Nama KPI	Deskripsi
Rasio buka email, dikelompokkan berdasarkan kampanye yang dijalankan	<code>email-open-rate-grouped-by-campaign-activity</code>	<p>Untuk setiap kampanye yang dijalankan, persentase pesan email yang dibuka oleh penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan email yang dikirim oleh kampanye yang dijalankan dan dibuka oleh penerima, dibagi dengan jumlah pesan email yang dikirim oleh kampanye yang dijalankan dan dikirim ke penerima.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>
Email dibuka, dikelompokkan berdasarkan kampanye	<code>direct-email-opens-grouped-by-campaign-activity</code>	<p>Untuk setiap kampanye yang dijalankan, jumlah pesan email yang dibuka oleh penerima.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>

Metrik	Nama KPI	Deskripsi
Pengiriman titik akhir	<code>unique-deliveries</code>	Untuk semua kampanye berjalan, jumlah titik akhir unik yang dikirimkan pesan.
Pengiriman titik akhir, dikelompokkan berdasarkan kampanye	<code>unique-deliveries-grouped-by-campaign-activity</code>	Untuk setiap kampanye yang dijalankan, jumlah titik akhir unik yang dikirimkan pesan. Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.
Pengiriman titik akhir, dikelompokkan berdasarkan tanggal	<code>unique-deliveries-grouped-by-date</code>	Untuk semua kampanye berjalan, jumlah titik akhir unik yang dikirimkan pesan, untuk setiap hari dalam rentang tanggal yang ditentukan. Hasil kueri untuk metrik ini dikelompokkan berdasarkan hari kalender, dalam format ISO 8601 yang diperpanjang.

Metrik	Nama KPI	Deskripsi
Tautan diklik, dikelompokkan berdasarkan kampanye yang dijalankan	<code>clicks-grouped-by-campaign-activity</code>	<p>Untuk setiap kampanye yang dijalankan, berapa kali penerima mengklik tautan dalam pesan email. Jika satu penerima mengklik beberapa tautan dalam pesan, atau mengklik tautan yang sama lebih dari sekali, setiap klik disertakan dalam hitungan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>

Metrik	Nama KPI	Deskripsi
Pesan terkirim, dikelompokkan berdasarkan kampanye yang dijalankan	successful-deliveries-grouped-by-campaign-activity	<p>Untuk setiap kampanye yang dijalankan, jumlah pesan yang dikirim ke penerima.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim oleh kampanye yang dijalankan, dikurangi jumlah pesan yang tidak dapat dikirimkan ke penerima proses karena pantulan keras.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>
Pesan terkirim, dikelompokkan berdasarkan kampanye yang dijalankan	attempted-deliveries-grouped-by-campaign-activity	<p>Untuk setiap kampanye yang dijalankan, jumlah pesan yang dikirim.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>

Metrik	Nama KPI	Deskripsi
Dorong tingkat buka	push-open-rate	<p>Untuk semua kampanye berjalan, persentase pemberitahuan push yang dibuka oleh penerima.</p> <p>Metrik ini dihitung sebagai jumlah pemberitahuan push yang dikirim oleh semua kampanye yang dijalankan dan dibuka oleh penerima, dibagi dengan jumlah pemberitahuan push yang dikirim oleh semua kampanye yang dijalankan dan dikirim ke penerima.</p>

Metrik	Nama KPI	Deskripsi
<p>Push open rate, dikelompokkan berdasarkan kampanye yang dijalankan</p>	<p><code>push-open-rate-grouped-by-campaign-activity</code></p>	<p>Untuk setiap kampanye yang dijalankan, persentase pemberitahuan push yang dibuka oleh penerima.</p> <p>Metrik ini dihitung sebagai jumlah pemberitahuan push yang dikirim oleh kampanye yang dijalankan dan dibuka oleh penerima, dibagi dengan jumlah pemberitahuan push yang dikirim oleh kampanye yang dijalankan dan dikirim ke penerima.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>
<p>Total push dibuka, dikelompokkan berdasarkan kampanye</p>	<p><code>direct-push-opens-grouped-by-campaign-activity</code></p>	<p>Untuk setiap kampanye yang dijalankan, jumlah pemberitahuan push yang dibuka oleh penerima.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas kampanye (<code>CampaignActivityId</code>), yang merupakan string yang secara unik mengidentifikasi kampanye yang dijalankan.</p>

Metrik	Nama KPI	Deskripsi
Total pengeluaran SMS	sms-spend	Untuk semua kampanye, jumlah total uang, dalam milisen, dihabiskan untuk mengirim SMS.

Metrik keterlibatan perjalanan

Tabel berikut mencantumkan dan menjelaskan metrik keterlibatan perjalanan standar yang dapat Anda kueri untuk memantau tren semua pesan email yang dikirim oleh perjalanan Amazon Pinpoint. Untuk menanyakan data metrik ini, gunakan sumber daya [metrik keterlibatan Journey](#) dari Amazon Pinpoint API. Kolom kpi-name dalam tabel menunjukkan nilai yang akan digunakan untuk kpi-name parameter dalam kueri.

Metrik	Nama KPI	Deskripsi
Klik	journey-emails-clicked	Berapa kali peserta mengklik tautan dalam pesan. Jika satu peserta mengklik beberapa tautan dalam pesan, atau mengklik tautan yang sama lebih dari sekali, setiap klik disertakan dalam hitungan.
Klik, dikelompokkan berdasarkan aktivitas	emails-clicked-grouped-by-journey-activity	Untuk setiap aktivitas dalam perjalanan, berapa kali peserta mengklik tautan dalam pesan. Jika satu peserta mengklik beberapa tautan dalam pesan, atau mengklik tautan yang sama lebih dari sekali, setiap klik disertakan dalam hitungan. Hasil kueri untuk metrik ini dikelompokkan berdasarkan

Metrik	Nama KPI	Deskripsi
		an ID aktivitas (JourneyActivityId), yang merupakan string yang mengidentifikasi aktivitas secara unik.
Aduan	journey-emails-complained	Jumlah pesan yang dilaporkan oleh peserta sebagai email yang tidak diminta atau tidak diinginkan.
Keluhan, dikelompokkan berdasarkan aktivitas	emails-complained-grouped-by-journey-activity	<p>Untuk setiap aktivitas dalam perjalanan, jumlah pesan yang dilaporkan oleh peserta sebagai email yang tidak diminta atau tidak diinginkan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas (JourneyActivityId), yang merupakan string yang mengidentifikasi aktivitas secara unik.</p>
Pengiriman	journey-emails-delivered	<p>Jumlah pesan yang dikirimkan ke peserta.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim, dikurangi jumlah pesan yang tidak dapat dikirim karena pantulan lunak atau keras, atau karena ditolak.</p>

Metrik	Nama KPI	Deskripsi
Pengiriman, dikelompokkan berdasarkan aktivitas	emails-delivered-grouped-by-journey-activity	<p>Untuk setiap kegiatan dalam perjalanan, jumlah pesan yang disampaikan kepada peserta.</p> <p>Metrik ini dihitung sebagai jumlah pesan yang dikirim, dikurangi jumlah pesan yang tidak dapat dikirim karena pantulan lunak atau keras, atau karena ditolak, untuk setiap aktivitas dalam perjalanan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas (JourneyActivityId), yang merupakan string yang mengidentifikasi aktivitas secara unik.</p>
Pantulan keras	journey-emails-hardbounced	<p>Jumlah pesan yang tidak dapat dikirim ke peserta karena pantulan yang keras. Hard bounce terjadi jika masalah persisten mencegah pesan dikirimkan—misalnya, jika alamat email peserta tidak ada.</p>

Metrik	Nama KPI	Deskripsi
Pantulan keras, dikelompokkan berdasarkan aktivitas	<code>emails-hardbounced-grouped-by-journey-activity</code>	<p>Untuk setiap aktivitas dalam perjalanan, jumlah pesan yang tidak dapat dikirimkan ke peserta karena pantulan yang keras. Hard bounce terjadi jika masalah persisten mencegah pesan dikirimkan—misalnya, jika alamat email peserta tidak ada.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas (<code>JourneyActivityId</code>), yang merupakan string yang mengidentifikasi aktivitas secara unik.</p>
Membuka	<code>journey-emails-opened</code>	Jumlah pesan yang dibuka oleh peserta.
Dibuka, dikelompokkan berdasarkan aktivitas	<code>emails-opened-grouped-by-journey-activity</code>	<p>Untuk setiap kegiatan dalam perjalanan, jumlah pesan yang dibuka oleh peserta.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas (<code>JourneyActivityId</code>), yang merupakan string yang mengidentifikasi aktivitas secara unik.</p>

Metrik	Nama KPI	Deskripsi
Penolakan	<code>journey-emails-rejected</code>	Jumlah pesan yang tidak dikirim ke peserta karena ditolak. Pesan ditolak jika Amazon Pinpoint menentukan bahwa pesan tersebut berisi malware. Amazon Pinpoint tidak mencoba mengirim pesan yang ditolak.
Penolakan, dikelompokkan berdasarkan aktivitas	<code>emails-rejected-grouped-by-journey-activity</code>	<p>Untuk setiap aktivitas dalam perjalanan, jumlah pesan yang tidak dikirim ke peserta karena ditolak. Pesan ditolak jika Amazon Pinpoint menentukan bahwa pesan tersebut berisi malware. Amazon Pinpoint tidak mencoba mengirim pesan yang ditolak.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas (<code>JourneyActivityId</code>), yang merupakan string yang mengidentifikasi aktivitas secara unik.</p>
Mengirim	<code>journey-emails-sent</code>	Jumlah pesan yang dikirim.

Metrik	Nama KPI	Deskripsi
Mengirim, dikelompokkan berdasarkan aktivitas	emails-sent-grouped-by-journey-activity	<p>Untuk setiap aktivitas dalam perjalanan, jumlah pesan yang dikirim.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas (JourneyActivityId), yang merupakan string yang mengidentifikasi aktivitas secara unik.</p>
Pantulan lembut	journey-emails-softbounced	Jumlah pesan yang tidak dapat dikirim ke peserta karena soft bounce. Soft bounce terjadi jika masalah sementara mencegah pesan dikirimkan—misalnya, jika kotak masuk peserta penuh atau server penerima sementara tidak tersedia.

Metrik	Nama KPI	Deskripsi
Pantulan lunak, dikelompokkan berdasarkan aktivitas	<code>emails-softbounced-grouped-by-journey-activity</code>	<p>Untuk setiap aktivitas dalam perjalanan, jumlah pesan yang tidak dapat dikirimkan ke peserta karena soft bounce. Soft bounce terjadi jika masalah sementara mencegah pesan dikirimkan—misalnya, jika kotak masuk peserta penuh atau server penerima sementara tidak tersedia.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas (<code>JourneyActivityId</code>), yang merupakan string yang mengidentifikasi aktivitas secara unik.</p>
Berhenti berlangganan	<code>journey-emails-unsubscribed</code>	<p>Berapa kali peserta mengklik tautan berhenti berlangganan dalam pesan. Jika satu peserta mengklik tautan berhenti berlangganan yang sama lebih dari sekali, setiap klik disertakan dalam hitungan.</p>

Metrik	Nama KPI	Deskripsi
Berhenti berlangganan, dikelompokkan berdasarkan aktivitas	emails-unsubscribed-grouped-by-journey-activity	<p>Untuk setiap aktivitas dalam perjalanan, berapa kali peserta mengklik tautan berhenti berlangganan dalam pesan. Jika satu peserta mengklik tautan berhenti berlangganan yang sama lebih dari sekali, setiap klik disertakan dalam hitungan.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan ID aktivitas (JourneyActivityId), yang merupakan string yang mengidentifikasi aktivitas secara unik.</p>

Metrik eksekusi perjalanan

Tabel berikut mencantumkan dan menjelaskan metrik eksekusi standar yang dapat Anda kueri untuk menilai status peserta dalam perjalanan Amazon Pinpoint. Untuk menanyakan data metrik ini, gunakan sumber daya [metrik eksekusi Journey](#) dari Amazon Pinpoint API. Kolom Bidang dalam tabel mengidentifikasi nama bidang yang muncul dalam hasil kueri untuk setiap metrik.

Metrik	Bidang	Deskripsi
Peserta aktif	ENDPOINT_ACTIVE	<p>Jumlah peserta yang aktif melanjutkan kegiatan dalam perjalanan.</p> <p>Metrik ini dihitung sebagai jumlah peserta yang memulai perjalanan, dikurangi jumlah peserta yang meninggal</p>

Metrik	Bidang	Deskripsi
		kan perjalanan dan jumlah peserta yang dikeluarkan dari perjalanan.
Pembatalan peserta	CANCELLED	Jumlah peserta yang tidak menyelesaikan perjalanan karena perjalanan dibatalkan.
Keberangkatan peserta	ENDPOINT_LEFT	Jumlah peserta yang meninggalkan perjalanan.
Entri peserta	ENDPOINT_ENTERED	Jumlah peserta yang memulai perjalanan.
Pengecualian peserta, batas masuk kembali	REENTRY_CAP_EXCEEDED	Jumlah peserta yang tidak menyelesaikan perjalanan karena mereka akan melebihi jumlah maksimum kali satu peserta dapat memasuki kembali perjalanan.
Pengecualian peserta, penolakan	ACTIVE_ENDPOINT_REJECTED	<p>Jumlah peserta yang tidak dapat memulai perjalanan karena mereka sudah menjadi peserta aktif dalam perjalanan.</p> <p>Peserta ditolak jika mereka memulai perjalanan dan Anda kemudian memperbarui definisi titik akhir mereka dengan cara yang memengaruhi inklusi mereka dalam suatu segmen (berdasarkan kriteria segmen) atau perjalanan (berdasarkan kondisi aktivitas).</p>

Metrik eksekusi aktivitas perjalanan

Tabel berikut mencantumkan dan menjelaskan metrik eksekusi standar yang dapat Anda kueri untuk menilai status peserta di setiap jenis aktivitas individual untuk perjalanan Amazon Pinpoint. Untuk menanyakan data metrik ini, gunakan sumber daya [metrik eksekusi aktivitas Journey](#) dari Amazon Pinpoint API. Kolom Metrik dalam tabel mencantumkan bidang yang muncul dalam hasil kueri untuk setiap jenis aktivitas. Ini juga memberikan deskripsi singkat dari setiap bidang.

Jenis aktivitas	Metrik
Ya/Tidak split () CONDITIONAL_SPLIT	<p>Metriknya adalah:</p> <ul style="list-style-type: none"> • <code>Branch_FALSE</code> — Jumlah peserta yang tidak memenuhi kondisi kegiatan dan melanjutkan ke aktivitas di jalur “Tidak”. • <code>Branch_TRUE</code> — Jumlah peserta yang memenuhi persyaratan kegiatan dan melanjutkan ke kegiatan di jalur “Ya”. <p>Metrik tambahan tersedia untuk aktivitas di setiap jalur. Untuk informasi tentang metrik tersebut, lihat baris dalam tabel ini untuk jenis aktivitas tersebut.</p>
Penahanan () HOLDOUT	<p>Metriknya adalah:</p> <ul style="list-style-type: none"> • <code>HOLDOUT</code>— Jumlah peserta yang dikeluarkan dari perjalanan sebagai bagian dari persentase holdout untuk kegiatan tersebut. • <code>PASSED</code>— Jumlah peserta yang melanjutkan ke kegiatan selanjutnya dalam perjalanan.
Email (MESSAGE)	<p>Metriknya adalah:</p> <ul style="list-style-type: none"> • <code>DAILY_CAP_EXCEEDED</code> — Jumlah pesan yang tidak dikirim karena mereka akan melebihi jumlah maksimum pesan yang

Jenis aktivitas	Metrik
	<p>dapat diterima oleh satu peserta selama periode 24 jam.</p> <ul style="list-style-type: none"> • FAILURE_PERMANENT — Jumlah pesan yang tidak dikirim karena masalah permanen. • QUIET_TIME — Jumlah pesan yang tidak dikirim karena akan dikirim selama waktu tenang untuk zona waktu peserta. • SERVICE_FAILURE — Jumlah pesan yang tidak dikirim karena masalah dengan Amazon Pinpoint. • SUCCESS— Jumlah pesan yang berhasil dikirim ke peserta. • THROTTLED — Jumlah pesan yang tidak dikirim karena mengirimnya akan melebihi kuota pengiriman untuk akun Amazon Pinpoint Anda. • TRANSIENT_FAILURE — Jumlah pesan yang tidak dikirim karena masalah sementara . • UNKNOWN— Jumlah pesan yang tidak dikirim karena masalah yang tidak diketahui.
Pemisahan multivariat () MULTI_CONDITIONAL_SPLIT	<p>Untuk setiap jalur kegiatan, jumlah peserta yang melanjutkan ke kegiatan di jalur.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan jalur, Branch_# di mana # adalah pengidentifikasi numerik untuk jalur—misalnya, Branch_1 untuk jalur pertama aktivitas.</p> <p>Metrik tambahan tersedia untuk aktivitas di setiap jalur. Untuk informasi tentang metrik tersebut, lihat baris dalam tabel ini untuk jenis aktivitas tersebut.</p>

Jenis aktivitas	Metrik
Pemisahan acak (RANDOM_SPLIT)	<p>Untuk setiap jalur kegiatan, jumlah peserta yang melanjutkan ke kegiatan di jalur.</p> <p>Hasil kueri untuk metrik ini dikelompokkan berdasarkan jalur, Branch_# di mana # adalah pengidentifikasi numerik untuk jalur—misalnya, Branch_1 untuk jalur pertama aktivitas.</p> <p>Metrik tambahan tersedia untuk aktivitas di setiap jalur. Untuk informasi tentang metrik tersebut, lihat baris dalam tabel ini untuk jenis aktivitas tersebut.</p>
Tunggu (WAIT)	<p>Metriknya adalah:</p> <ul style="list-style-type: none">• WAIT_FINISHED — Jumlah peserta yang selesai menunggu jumlah waktu yang ditentukan.• WAIT_SKIPPED — Jumlah peserta yang tidak menunggu jumlah waktu yang ditentukan, biasanya karena mereka memulai aktivitas atau perjalanan setelah waktu akhir yang dijadwalkan untuk kegiatan tersebut.• WAIT_STARTED — Jumlah peserta yang mulai menunggu, dan belum melewati atau selesai menunggu jumlah waktu yang ditentukan.

Jenis aktivitas	Metrik
Pusat Kontak (CONTACT_CENTER)	<p>Metriknya adalah:</p> <ul style="list-style-type: none">• CALL_QUEUED — Jumlah peserta yang telah dipanggil dan mengantri ke Amazon Connect. Termasuk upaya panggilan ulang.• CONTINUE_WAITING — Jumlah peserta yang terus menunggu upaya dial dilakukan.• DIAL_FAILURE — Jumlah peserta dengan upaya dial yang gagal.• DROPPED— Jumlah peserta yang tidak lagi memenuhi persyaratan yang ditentukan dalam kegiatan perjalanan sebelumnya pada waktu kirim.• TIMEOUT— Jumlah peserta yang tidak menerima kode disposisi Amazon Connect setelah beberapa kali mencoba panggilan.• WAIT_FINISHED — Jumlah peserta yang selesai menunggu jumlah waktu yang ditentukan.• WAIT_FOR_QUIET_HOURS — Jumlah peserta yang menunggu waktu tenang selesai untuk mengantarkan ke saluran.• WAIT_STARTED — Jumlah peserta yang mulai menunggu, dan belum melewati atau selesai menunggu jumlah waktu yang ditentukan.

Metrik eksekusi perjalanan dan kampanye

Anda dapat melakukan kueri metrik eksekusi standar untuk menilai status peserta di setiap jenis aktivitas individual untuk perjalanan atau kampanye Amazon Pinpoint. Untuk menanyakan data metrik ini, gunakan metrik [eksekusi aktivitas yang dijalankan Journey](#) atau sumber daya [Metrik](#)

[Kampanye](#) Amazon Pinpoint API. Tabel berikut mencantumkan bidang yang muncul dalam hasil kueri untuk setiap jenis aktivitas.

Nama Metrik	Berlaku untuk Perjalanan, Kampanye, atau Keduanya	Deskripsi
ENDPOINT_PRODUCED	Keduanya	Jumlah titik akhir yang awalnya dihasilkan dari segmen atau peristiwa sebelum penyaringan apa pun.
TITIK AKHIR_FROM_USER	Keduanya	Jika pelanggan hanya memiliki segmen user-id, maka semua titik akhir pengguna tersebut akan ditambahkan. Metrik ini mengukur jumlah titik akhir yang ditambahkan dengan cara ini.
TITIK AKHIR_OPT_OUT	Keduanya	Titik akhir dipilih keluar dan tidak memasuki kampanye atau perjalanan.
ENDPOINT_INACTIVE	Keduanya	Titik akhir tidak aktif dan tidak memasuki kampanye atau perjalanan.
DISARING_OUT_BY_SEGMENT	Keduanya	Endpoint tidak cocok dengan filter segmen dan tidak memasuki kampanye atau perjalanan.
ENDPOINT_MISSING_ADDRESS	Keduanya	Endpoint kehilangan alamat dan tidak memasuki kampanye atau perjalanan.

Nama Metrik	Berlaku untuk Perjalanan, Kampanye, atau Keduanya	Deskripsi
ENDPOINT_MISSING_CHANNEL	Keduanya	Endpoint kehilangan saluran dan tidak memasuki kampanye atau perjalanan.
ENDPOINT_MISSING_TIMEZONE	Keduanya	Titik akhir kehilangan nilai untuk zona waktu dan disaring. Ini hanya terjadi ketika nilai zona waktu diperlukan.
ENDPOINT_TIMEZONE_MISMATCH	Keduanya	Endpoint berada di zona waktu yang tidak termasuk dalam eksekusi pada saat itu.
ENDPOINT_CHANNEL_MISMATCH	Kampanye	Kampanye tidak memiliki pesan yang dikonfigurasi untuk jenis saluran titik akhir ini.
DUPLIKAT_TITIK AKHIR	Keduanya	Titik akhir duplikat ditemukan dan dihilangkan ditipu.
DUPLIKATE_USER	Keduanya	Pengguna duplikat ditemukan dan di-de-ditipu dari segmen khusus user-id. Jika mereka memiliki id pengguna yang sama, metrik 1 akan dipancarkan.
DIJEDA	Perjalanan	Dihapus dari eksekusi karena perjalanan dijeda.
BERAKHIR	Perjalanan	Dihapus dari eksekusi karena perjalanan telah berakhir.

Nama Metrik	Berlaku untuk Perjalanan, Kampanye, atau Keduanya	Deskripsi
PENGOBATAN_HOLDOUT	Kampanye	Ini dipancarkan dalam kampanye A/B, untuk titik akhir yang kohortnya tidak cocok dengan perlakuan saat ini. Misalnya dalam pemisahan 50/50 A/B, 50% titik akhir akan memancarkan metrik ini untuk setiap perlakuan
ENDPOINT_ESTIMATED_TIMEZONE	Perjalanan	Estimasi zona waktu mampu memperkirakan zona waktu untuk titik akhir.

Mengkueri data analitik Amazon Pinpoint untuk kampanye

Selain menggunakan halaman analitik di konsol Amazon Pinpoint, Anda dapat menggunakan API Amazon Pinpoint Analytics untuk mengkueri data analitik untuk subset metrik standar yang memberikan wawasan tentang tren pengiriman dan keterlibatan untuk kampanye.

Masing-masing metrik ini adalah nilai yang dapat diukur, juga disebut sebagai indikator kinerja kunci (KPI), yang dapat membantu Anda memantau dan menilai kinerja satu atau lebih kampanye. Misalnya, Anda dapat menggunakan metrik untuk mengetahui berapa banyak titik akhir yang dikirim ke pesan kampanye, atau berapa banyak pesan tersebut yang dikirim ke titik akhir yang dimaksud.

Amazon Pinpoint secara otomatis mengumpulkan dan mengumpulkan data ini untuk semua kampanye Anda. Ini menyimpan data selama 90 hari. Jika Anda mengintegrasikan aplikasi seluler dengan Amazon Pinpoint dengan menggunakan AWS Mobile SDK, Amazon Pinpoint memperluas dukungan ini untuk menyertakan metrik tambahan, seperti persentase notifikasi push yang dibuka oleh penerima. Untuk informasi tentang mengintegrasikan aplikasi seluler, lihat [Mengintegrasikan Amazon Pinpoint dengan aplikasi Anda](#).

Jika Anda menggunakan API Amazon Pinpoint Analytics untuk melakukan kueri data, Anda dapat memilih berbagai opsi yang menentukan cakupan, data, pengelompokan, dan filter untuk kueri Anda.

Anda melakukan ini dengan menggunakan parameter yang menentukan proyek, kampanye, dan metrik yang ingin Anda kueri, selain filter berbasis tanggal apa pun yang ingin Anda terapkan.

Topik ini menjelaskan dan memberikan contoh cara memilih opsi ini dan menanyakan data untuk satu atau beberapa kampanye.

Prasyarat

Sebelum Anda membuat kueri data analitik untuk satu atau beberapa kampanye, ini membantu mengumpulkan informasi berikut, yang Anda gunakan untuk menentukan kueri Anda:

- ID Proyek— Pengenal unik untuk proyek yang terkait dengan kampanye atau kampanye. Di Amazon Pinpoint API, nilai ini disimpan dalam `application-id` properti. Pada konsol Amazon Pinpoint, nilai ini ditampilkan sebagai ID Proyek pada Semua proyek halaman.
- ID kampanye— Pengenal unik untuk kampanye, jika Anda ingin query data hanya untuk satu kampanye. Di Amazon Pinpoint API, nilai ini disimpan dalam `campaign-id` properti. Nilai ini tidak ditampilkan di konsol.
- Rentang tanggal— Opsional, tanggal pertama dan terakhir dan waktu rentang tanggal untuk query data untuk. Rentang tanggal inklusif dan harus dibatasi hingga 31 hari kalender atau lebih sedikit. Selain itu, mereka harus mulai kurang dari 90 hari dari hari ini. Jika Anda tidak menentukan rentang tanggal, Amazon Pinpoint secara otomatis mengkueri data 31 hari kalender sebelumnya.
- Tipe metrik— Jenis metrik untuk query. Ada dua tipe, metrik aplikasi dan metrik kampanye. Sesimetri aplikasi menyediakan data untuk semua kampanye yang terkait dengan proyek, juga disebut sebagai penerapan. SEBUAH metrik kampanye menyediakan data untuk hanya satu kampanye.
- Metrik- Nama metrik untuk kueri — lebih khusus, `kpi-name` untuk metrik. Untuk daftar lengkap terkait metrik yang didukung dan `kpi-name` nilai untuk masing-masing, lihat [Metrik standar](#).

Hal ini juga membantu untuk menentukan apakah Anda ingin mengelompokkan data dengan bidang yang relevan. Jika Anda melakukannya, Anda dapat menyederhanakan analisis dan pelaporan dengan memilih metrik yang dirancang untuk mengelompokkan data untuk Anda secara otomatis. Misalnya, Amazon Pinpoint menyediakan beberapa metrik standar yang melaporkan persentase pesan yang dikirim ke penerima kampanye. Salah satu metrik ini secara otomatis mengelompokkan data berdasarkan tanggal (`successful-delivery-rate-grouped-by-date`). Metrik lain secara otomatis mengelompokkan data berdasarkan kampanye yang dijalankan (`successful-delivery-rate-grouped-by-campaign-activity`). Metrik ketiga hanya mengembalikan nilai tunggal

—persentase pesan yang dikirim ke penerima oleh semua kampanye berjalan (`successful-delivery-rate`).

Jika Anda tidak dapat menemukan metrik standar yang mengelompokkan data seperti yang Anda inginkan, Anda dapat mengembangkan serangkaian kueri yang mengembalikan data yang Anda inginkan. Anda kemudian dapat secara manual memecah atau menggabungkan hasil kueri ke dalam kelompok kustom yang Anda desain.

Akhirnya, penting untuk memverifikasi bahwa Anda berwenang untuk mengakses data yang ingin Anda kueri. Untuk informasi selengkapnya, lihat [Kebijakan IAM untuk kueri data analitik Amazon Pinpoint](#).

Mengkueri data untuk satu kampanye

Untuk melakukan kueri data untuk satu kampanye, Anda menggunakan [Metrik kampanye](#) API dan tentukan nilai untuk parameter wajib berikut:

- `Application-id`— ID proyek, yang merupakan pengenalan unik untuk proyek yang terkait dengan kampanye. Di Amazon Pinpoint, istilahnya `proyek` dan `penerapan` memiliki arti yang sama.
- `kampanye-id`— Pengidentifikasi unik untuk kampanye.
- `name`— Nama metrik untuk query. Nilai ini menggambarkan metrik terkait dan terdiri dari dua atau lebih istilah, yang terdiri dari karakter alfanumerik huruf kecil, dipisahkan oleh tanda hubung. Untuk daftar lengkap terkait metrik yang didukung dan `i-name` nilai untuk masing-masing, lihat [Metrik standar](#).

Anda juga dapat menerapkan filter yang query data untuk rentang tanggal tertentu. Jika Anda tidak menentukan rentang tanggal, Amazon Pinpoint mengembalikan data 31 hari kalender sebelumnya. Untuk memfilter data berdasarkan tanggal yang berbeda, gunakan parameter rentang tanggal yang didukung untuk menentukan tanggal dan waktu pertama dan terakhir dari rentang tanggal. Nilai-nilai harus dalam format ISO 8601 diperpanjang dan menggunakan Coordinated Universal Time (UTC) — misalnya, `2019-07-19T20:00:00Z` untuk 8:00 PM UTC 19 Juli 2019. Rentang tanggal inklusif dan harus dibatasi hingga 31 hari kalender atau lebih sedikit. Selain itu, tanggal dan waktu pertama harus kurang dari 90 hari dari hari ini.

Contoh berikut menunjukkan cara mengkueri data analitik untuk kampanye dengan menggunakan Amazon Pinpoint REST API, AWS CLI, dan AWS SDK for Java. Anda dapat menggunakan apa pun yang didukung AWS SDK untuk mengkueri data analitik untuk kampanye. Parameter AWS CLI contoh

diformat untuk Microsoft Windows. Untuk Unix, Linux, dan macOS, ganti karakter kelanjutan baris tanda sisipan (^) dengan garis miring terbalik (\).

REST API

Untuk mengkueri data analitik kampanye menggunakan API REST Amazon Pinpoint, kirim permintaan GET HTTP (S) ke [Metrik kampanye](#)URI. Dalam URI, tentukan nilai yang sesuai untuk parameter jalur yang diperlukan:

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

Dengan:

- *titik akhir* adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menjadi tuan rumah proyek terkait dengan kampanye.
- *Application-id* adalah pengidentifikasi unik untuk proyek yang dikaitkan dengan kampanye.
- *kampanye-id* adalah pengidentifikasi unik untuk kampanye.
- *name* adalah kpi-name value untuk metrik untuk query.

Semua parameter harus URL dikodekan.

Untuk menerapkan filter yang query data untuk rentang tanggal tertentu, tambahkan *start-time* dan *end-time* parameter kueri dan nilai ke URI. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Gunakan ampersand (&) untuk memisahkan parameter.

Misalnya, permintaan berikut mengambil jumlah titik akhir unik yang dikirimkan ke pesan, oleh semua menjalankan kampanye, dari 19 Juli 2019 hingga 26 Juli 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/1234567890123456789012345example/campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Dengan:

- *pinpoint.us-east-1.amazonaws.com* adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menjadi tuan rumah proyek.

- 1234567890123456789012345example adalah pengidentifikasi unik untuk proyek yang dikaitkan dengan kampanye.
- 80b8efd84042ff8d9c96ce2f8example adalah pengidentifikasi unik untuk kampanye.
- unique-deliveries adalah kpi-name value untuk pengiriman endpoint metrik kampanye, yang merupakan metrik yang melaporkan jumlah titik akhir unik yang dikirim ke pesan, oleh semua menjalankan kampanye.
- 2019-07-19T00:00:00Z adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- 2019-07-26T23:59:59Z adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

AWS CLI

Untuk melakukan kueri data analitik untuk kampanye dengan menggunakan AWS CLI, menggunakan `get-campaign-date-range-kpi` perintah dan menentukan nilai yang sesuai untuk parameter yang diperlukan:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
  --application-id application-id ^
  --campaign-id campaign-id ^
  --kpi-name kpi-name
```

Dengan:

- *Application-id* adalah pengidentifikasi unik untuk proyek yang dikaitkan dengan kampanye.
- *kampanye-id* adalah pengidentifikasi unik untuk kampanye.
- *name* adalah kpi-name value untuk metrik untuk query.

Untuk menerapkan filter yang query data untuk rentang tanggal tertentu, tambahkan `start-time` dan `end-time` parameter dan nilai untuk kueri Anda. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Misalnya, permintaan berikut mengambil jumlah titik akhir unik yang dikirimkan ke pesan, oleh semua menjalankan kampanye, dari 19 Juli 2019 hingga 26 Juli 2019:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
  --application-id 1234567890123456789012345example ^
```



```
--campaign-id 80b8efd84042ff8d9c96ce2f8example ^
--kpi-name unique-deliveries ^
--start-time 2019-07-19T00:00:00Z ^
--end-time 2019-07-26T23:59:59Z
```

Dengan:

- 1234567890123456789012345example adalah pengidentifikasi unik untuk proyek yang dikaitkan dengan kampanye.
- 80b8efd84042ff8d9c96ce2f8example adalah pengidentifikasi unik untuk kampanye.
- unique-deliveries adalah kpi-name value untuk pengiriman endpoint metrik kampanye, yang merupakan metrik yang melaporkan jumlah titik akhir unik yang dikirim ke pesan, oleh semua menjalankan kampanye.
- 2019-07-19T00:00:00Z adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- 2019-07-26T23:59:59Z adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

SDK for Java

Untuk melakukan kueri data analitik untuk kampanye dengan menggunakan AWS SDK for Java, menggunakan `GetCampaignDateRangeKpiRequest` metode [Metrik kampanye API](#). Tentukan nilai yang sesuai untuk parameter yang diperlukan:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
    .withApplicationId("applicationId")
    .withCampaignId("campaignId")
    .withKpiName("kpiName")
```

Dengan:

- *applicationId* adalah pengidentifikasi unik untuk proyek yang dikaitkan dengan kampanye.
- *campaignId* adalah pengidentifikasi unik untuk kampanye.
- *kpiName* adalah kpi-name value untuk metrik untuk query.

Untuk menerapkan filter yang meminta data untuk rentang tanggal tertentu, sertakan `startTime` dan `endTime` parameter dan nilai dalam kueri Anda. Dengan menggunakan

parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Misalnya, permintaan berikut mengambil jumlah titik akhir unik yang dikirimkan ke pesan, oleh semua menjalankan kampanye, dari 19 Juli 2019 hingga 26 Juli 2019:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()  
    .withApplicationId("1234567890123456789012345example")  
    .withCampaignId("80b8efd84042ff8d9c96ce2f8example")  
    .withKpiName("unique-deliveries")  
    .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))  
    .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Dengan:

- 1234567890123456789012345example adalah pengidentifikasi unik untuk proyek yang dikaitkan dengan kampanye.
- 80b8efd84042ff8d9c96ce2f8example adalah pengidentifikasi unik untuk kampanye.
- unique-deliveries adalah kpi-name value untuk pengiriman endpoint metrik kampanye, yang merupakan metrik yang melaporkan jumlah titik akhir unik yang dikirim ke pesan, oleh semua menjalankan kampanye.
- 2019-07-19T00:00:00Z adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- 2019-07-26T23:59:59Z adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

Setelah Anda mengirim kueri Anda, Amazon Pinpoint mengembalikan hasil kueri dalam respons JSON. Struktur hasil bervariasi tergantung pada metrik yang Anda tanyakan. Beberapa metrik hanya mengembalikan satu nilai. Misalnya, pengiriman endpoint (unique-deliveries) metrik kampanye, yang digunakan dalam contoh sebelumnya, mengembalikan satu nilai—jumlah titik akhir unik yang dikirim ke pesan, oleh semua menjalankan kampanye. Dalam kasus ini, respons JSON adalah sebagai berikut:

```
{  
  "CampaignDateRangeKpiResponse": {  
    "ApplicationId": "1234567890123456789012345example",  
    "CampaignId": "80b8efd84042ff8d9c96ce2f8example",  
    "EndTime": "2019-07-26T23:59:59Z",
```

```

    "KpiName": "unique-deliveries",
    "KpiResult": {
      "Rows": [
        {
          "Values": [
            {
              "Key": "UniqueDeliveries",
              "Type": "Double",
              "Value": "123.0"
            }
          ]
        }
      ]
    },
    "StartTime": "2019-07-19T00:00:00Z"
  }
}

```

Metrik lain mengembalikan beberapa nilai, dan kelompokkan nilai dengan bidang yang relevan. Jika metrik mengembalikan beberapa nilai, respon JSON mencakup bidang yang menunjukkan bidang yang digunakan untuk mengelompokkan data.

Untuk mempelajari selengkapnya tentang struktur hasil kueri, lihat [Menggunakan hasil kueri](#).

Mengkueri data untuk beberapa kampanye

Ada dua cara untuk kueri data untuk beberapa kampanye. Cara terbaik tergantung pada apakah Anda ingin query data untuk kampanye yang semua terkait dengan proyek yang sama. Jika Anda melakukannya, itu juga tergantung pada apakah Anda ingin query data untuk semua atau hanya atau subset dari kampanye tersebut.

Untuk mengkueri data kampanye yang terkait dengan proyek yang berbeda atau hanya untuk subset kampanye yang terkait dengan proyek yang sama, pendekatan terbaik adalah membuat dan menjalankan serangkaian kueri individual, satu untuk setiap kampanye yang ingin Anda kueri data. Bagian sebelumnya menjelaskan cara menanyakan data hanya untuk satu kampanye.

Untuk query data untuk semua kampanye yang terkait dengan proyek yang sama, Anda dapat menggunakan [Metrik Aplikasi](#) API. Tentukan parameter wajib berikut:

- **Application-id**— ID proyek, yang merupakan pengenalan unik untuk proyek. Di Amazon Pinpoint, istilahnya `proyek` dan `penerapan` memiliki arti yang sama.

- **name**- Nama metrik untuk kueri. Nilai ini menggambarkan metrik terkait dan terdiri dari dua atau lebih istilah, yang terdiri dari karakter alfanumerik huruf kecil, dipisahkan oleh tanda hubung. Untuk daftar lengkap terkait metrik yang didukung `kpi-name` untuk masing-masing, lihat [Metrik standar](#).

Anda juga dapat memfilter data berdasarkan rentang tanggal. Jika Anda tidak menentukan rentang tanggal, Amazon Pinpoint mengembalikan data 31 hari kalender sebelumnya. Untuk memfilter data berdasarkan tanggal yang berbeda, gunakan parameter rentang tanggal yang didukung untuk menentukan tanggal dan waktu pertama dan terakhir dari rentang tanggal. Nilai-nilai harus dalam format ISO 8601 diperpanjang dan menggunakan Coordinated Universal Time (UTC) — misalnya, `2019-07-19T20:00:00Z` untuk 8:00 PM UTC 19 Juli 2019. Rentang tanggal inklusif dan harus dibatasi hingga 31 hari kalender atau lebih sedikit. Selain itu, tanggal dan waktu pertama harus kurang dari 90 hari dari hari ini.

Contoh berikut menunjukkan cara mengkueri data analitik untuk kampanye dengan menggunakan Amazon Pinpoint REST API, AWS CLI, dan AWS SDK for Java. Anda dapat menggunakan apa pun yang didukung AWS SDK untuk mengkueri data analitik untuk kampanye. Parameter AWS CLI contoh diformat untuk Microsoft Windows. Untuk Unix, Linux, dan macOS, ganti karakter kelanjutan baris tanda sisipan (^) dengan garis miring terbalik (\).

REST API

Untuk mengkueri data analitik untuk beberapa kampanye menggunakan API REST Amazon Pinpoint, kirim permintaan GET HTTP (S) ke [Metrik Aplikasi](#) URI. Dalam URI, tentukan nilai yang sesuai untuk parameter jalur yang diperlukan:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dengan:

- *titik akhir* adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menjadi tuan rumah proyek terkait dengan kampanye.
- *Application-id* adalah pengenal unik untuk proyek yang terkait dengan kampanye.
- *name* adalah `kpi-name` value untuk metrik untuk query.

Semua parameter harus URL dikodekan.

Untuk menerapkan filter yang mengambil data untuk rentang tanggal tertentu, tambahkan `start-time` dan `end-time` parameter kueri dan nilai ke URI. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Gunakan ampersand (&) untuk memisahkan parameter.

Misalnya, permintaan berikut mengambil jumlah titik akhir unik yang dikirimkan pesan, oleh masing-masing kampanye proyek, dari 19 Juli 2019 hingga 26 Juli 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/1234567890123456789012345example/kpis/daterange/unique-deliveries-grouped-by-campaign?start-time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Dengan:

- `pinpoint.us-east-1.amazonaws.com` adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menjadi tuan rumah proyek.
- `1234567890123456789012345example` adalah pengenal unik untuk proyek yang terkait dengan kampanye.
- `unique-deliveries-grouped-by-campaign` adalah `kpi-name` value untuk pengiriman endpoint, dikelompokkan berdasarkan kampanye metrik aplikasi, yang merupakan metrik yang mengembalikan jumlah titik akhir unik yang dikirim oleh pesan, oleh setiap kampanye.
- `2019-07-19T00:00:00Z` adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- `2019-07-26T23:59:59Z` adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

AWS CLI

Untuk melakukan kueri data analitik untuk beberapa kampanye dengan menggunakan AWS CLI, menggunakan `get-application-date-range-kpi` perintah dan menentukan nilai yang sesuai untuk parameter yang diperlukan:

```
C:\> aws pinpoint get-application-date-range-kpi ^  
  --application-id application-id ^  
  --kpi-name kpi-name
```

Dengan:

- *Application-id* adalah pengenal unik untuk proyek yang terkait dengan kampanye.
- *name* adalah kpi-namevalue untuk metrik untuk query.

Untuk menerapkan filter yang mengambil data untuk rentang tanggal tertentu, sertakan `start-time` dan `end-time` parameter dan nilai dalam kueri Anda. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Misalnya, permintaan berikut mengambil jumlah titik akhir unik yang dikirimkan pesan, oleh masing-masing kampanye proyek, dari 19 Juli 2019 hingga 26 Juli 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
  --application-id 1234567890123456789012345example ^
  --kpi-name unique-deliveries-grouped-by-campaign ^
  --start-time 2019-07-19T00:00:00Z ^
  --end-time 2019-07-26T23:59:59Z
```

Dengan:

- 1234567890123456789012345example adalah pengidentifikasi unik untuk proyek yang dikaitkan dengan kampanye.
- unique-deliveries-grouped-by-campaign adalah kpi-namevalue untuk pengiriman endpoint, dikelompokkan berdasarkan kampanye metrik aplikasi, yang merupakan metrik yang mengembalikan jumlah titik akhir unik yang dikirim oleh pesan, oleh setiap kampanye.
- 2019-07-19T00:00:00Z adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- 2019-07-26T23:59:59Z adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

SDK for Java

Untuk melakukan kueri data analitik untuk beberapa kampanye dengan menggunakan AWS SDK for Java, menggunakan `GetApplicationDateRangeKpiRequest` metode [Metrik Aplikasi API](#) Tentukan nilai yang sesuai untuk parameter yang diperlukan:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("applicationId")
    .withKpiName("kpiName")
```

Dengan:

- *applicationId* adalah pengenal unik untuk proyek yang terkait dengan kampanye.
- *kpiName* adalah *kpi-name* value untuk metrik untuk query.

Untuk menerapkan filter yang mengambil data untuk rentang tanggal tertentu, sertakan *startTime* dan *endTime* parameter dan nilai dalam kueri Anda. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Misalnya, permintaan berikut mengambil jumlah titik akhir unik yang dikirimkan pesan, oleh masing-masing kampanye proyek, dari 19 Juli 2019 hingga 26 Juli 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()  
    .withApplicationId("1234567890123456789012345example")  
    .withKpiName("unique-deliveries-grouped-by-campaign")  
    .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))  
    .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Dengan:

- 1234567890123456789012345example adalah pengenal unik untuk proyek yang terkait dengan kampanye.
- unique-deliveries-grouped-by-campaign adalah *kpi-name* value untuk pengiriman endpoint, dikelompokkan berdasarkan kampanye metrik aplikasi, yang merupakan metrik yang mengembalikan jumlah titik akhir unik yang dikirim oleh pesan, oleh setiap kampanye.
- 2019-07-19T00:00:00Z adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- 2019-07-26T23:59:59Z adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

Setelah Anda mengirim kueri Anda, Amazon Pinpoint mengembalikan hasil kueri dalam respons JSON. Struktur hasil bervariasi tergantung pada metrik yang Anda tanyakan. Beberapa metrik hanya mengembalikan satu nilai. Metrik lain mengembalikan beberapa nilai, dan nilai-nilai tersebut dikelompokkan berdasarkan bidang yang relevan. Jika metrik mengembalikan beberapa nilai, respon JSON mencakup bidang yang menunjukkan bidang yang digunakan untuk mengelompokkan data.

Misalnya, pengiriman endpoint, dikelompokkan berdasarkan kampanye (unique-deliveries-grouped-by-campaign) metrik aplikasi, yang digunakan dalam contoh sebelumnya, mengembalikan beberapa nilai—jumlah titik akhir unik yang dikirim ke pesan, untuk setiap kampanye yang terkait dengan proyek. Dalam kasus ini, respons JSON adalah sebagai berikut:

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



```
    },
    {
      "GroupedBy": [
        {
          "Key": "CampaignId",
          "Type": "String",
          "Value": "42d8c7eb0990a57ba1d5476a3example"
        }
      ],
      "Values": [
        {
          "Key": "UniqueDeliveries",
          "Type": "Double",
          "Value": "789.0"
        }
      ]
    }
  ],
  "StartTime": "2019-07-19T00:00:00Z"
}
```

Dalam kasus ini, `GroupedBy` menunjukkan bahwa nilai dikelompokkan berdasarkan ID kampanye (`CampaignId`).

Untuk mempelajari selengkapnya tentang struktur hasil kueri, lihat [Menggunakan hasil kueri](#).

Mengkueri data analitik Amazon Pinpoint untuk pesan transaksional

Selain menggunakan halaman analitik di konsol Amazon Pinpoint, Anda dapat menggunakan API Amazon Pinpoint Analytics untuk mengkueri data analitik untuk subset metrik standar yang memberikan wawasan tentang tren pengiriman dan keterlibatan untuk pesan transaksional yang dikirim untuk proyek.

Masing-masing metrik ini adalah nilai terukur, juga disebut sebagai indikator kinerja kunci (KPI), yang dapat membantu Anda memantau dan menilai kinerja pesan transaksional. Misalnya, Anda dapat menggunakan metrik untuk mengetahui berapa banyak email transaksional atau pesan SMS yang Anda kirim, atau berapa banyak pesan yang dikirim ke penerima. Amazon Pinpoint secara otomatis mengumpulkan dan mengumpulkan data ini untuk semua email transaksional dan pesan SMS yang Anda kirim untuk sebuah proyek. Ini menyimpan data selama 90 hari.

Jika Anda menggunakan API Amazon Pinpoint Analytics untuk melakukan kueri data, Anda dapat memilih berbagai opsi yang menentukan cakupan, data, pengelompokan, dan filter untuk kueri Anda. Anda melakukan ini dengan menggunakan parameter yang menentukan proyek dan metrik yang ingin Anda kueri, selain filter berbasis tanggal yang ingin Anda terapkan.

Topik ini menjelaskan dan memberikan contoh bagaimana memilih opsi ini dan query data pesan transaksional untuk proyek.

Prasyarat

Sebelum Anda melakukan kueri data analitik untuk pesan transaksional, ini membantu mengumpulkan informasi berikut, yang Anda gunakan untuk menentukan kueri Anda:

- ID proyek— Pengenal unik untuk proyek bahwa pesan dikirim dari. Di Amazon Pinpoint API, nilai ini disimpan dalam `application-id` properti. Pada konsol Amazon Pinpoint, nilai ini ditampilkan sebagai ID proyek pada Semua proyek halaman.
- Rentang tanggal— Opsional, tanggal pertama dan terakhir dan waktu rentang tanggal untuk query data untuk. Rentang tanggal sudah inklusif dan harus dibatasi hingga 31 hari kalender atau lebih sedikit. Selain itu, mereka harus mulai kurang dari 90 hari dari hari ini. Jika Anda tidak menentukan rentang tanggal, Amazon Pinpoint secara otomatis melakukan kueri 31 hari kalender sebelumnya.
- Metrik— Nama metrik untuk kueri — lebih khusus, `kpi-name` value untuk metrik. Untuk daftar lengkap terkait metrik yang didukung dan `kpi-name` nilai untuk masing-masing, lihat [Metrik standar](#).

Hal ini juga membantu untuk menentukan apakah Anda ingin mengelompokkan data dengan bidang yang relevan. Jika Anda melakukannya, Anda dapat menyederhanakan analisis dan pelaporan dengan memilih metrik yang dirancang untuk mengelompokkan data untuk Anda secara otomatis. Misalnya, Amazon Pinpoint menyediakan beberapa metrik standar yang melaporkan jumlah pesan SMS transaksional yang dikirimkan ke penerima. Salah satu metrik ini secara otomatis mengelompokkan data berdasarkan tanggal (`txn-sms-delivered-grouped-by-date`). Metrik lain secara otomatis mengelompokkan data berdasarkan negara atau wilayah (`txn-sms-delivered-grouped-by-country`). Metrik ketiga hanya mengembalikan nilai tunggal—jumlah pesan yang dikirim ke penerima (`txn-sms-delivered`). Jika Anda tidak dapat menemukan metrik standar yang mengelompokkan data seperti yang Anda inginkan, Anda dapat mengembangkan serangkaian kueri yang mengembalikan data yang Anda inginkan. Anda kemudian dapat secara manual memecah atau menggabungkan hasil kueri ke dalam kelompok kustom yang Anda desain.

Akhirnya, penting untuk memverifikasi bahwa Anda berwenang untuk mengakses data yang ingin Anda kueri. Untuk informasi selengkapnya, lihat [Kebijakan IAM untuk kueri data analitik Amazon Pinpoint](#).

Mengkueri data untuk pesan email transaksional

Untuk query data untuk pesan email transaksional yang dikirim untuk proyek, Anda menggunakan [Metrik Aplikasi](#) API dan tentukan parameter wajib berikut:

- **Application-id**— ID proyek, yang merupakan pengenalan unik untuk proyek. Di Amazon Pinpoint, istilahnya proyek dan penerapannya memiliki arti yang sama.
- **kpi-name**— Nama metrik untuk query. Nilai ini menggambarkan metrik terkait dan terdiri dari dua atau lebih istilah, yang terdiri dari karakter alfanumerik huruf kecil, dipisahkan oleh tanda hubung. Untuk daftar lengkap terkait metrik yang didukung dan `kpi-` nilai untuk masing-masing, lihat [Metrik standar](#).

Anda juga dapat menerapkan filter yang query data untuk rentang tanggal tertentu. Jika Anda tidak menentukan rentang tanggal, Amazon Pinpoint mengembalikan data 31 hari sebelumnya. Untuk memfilter data berdasarkan tanggal yang berbeda, gunakan parameter rentang tanggal yang didukung untuk menentukan tanggal dan waktu pertama dan terakhir dari rentang tanggal. Nilai-nilai harus dalam format ISO 8601 diperpanjang dan menggunakan Coordinated Universal Time (UTC) — misalnya, `2019-09-06T20:00:00Z` untuk 8:00 PM UTC 6 September 2019. Rentang tanggal sudah inklusif dan harus dibatasi hingga 31 hari kalender atau lebih sedikit. Selain itu, tanggal dan waktu pertama harus kurang dari 90 hari dari hari ini.

Contoh berikut menunjukkan cara mengkueri data analitik untuk pesan email transaksional dengan menggunakan Amazon Pinpoint REST API, AWS CLI, dan AWS SDK for Java. Anda dapat menggunakan apa pun yang didukung AWS SDK untuk query data analisis untuk pesan transaksional. Parameter AWS CLI contoh diformat untuk Microsoft Windows. Untuk Unix, Linux, dan macOS, ganti karakter kelanjutan baris tanda sisipan (^) dengan garis miring terbalik (\).

REST API

Untuk melakukan kueri data analitik untuk pesan email transaksional menggunakan API Amazon Pinpoint REST, kirim permintaan GET HTTP (S) ke [Metrik Aplikasi](#) URI. Dalam URI, tentukan nilai yang sesuai untuk parameter jalur yang diperlukan:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dengan:

- *titik akhir* adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menjadi tuan rumah proyek.
- *Application-id* Pengidentifikasi unik untuk proyek.
- *kpi-name* adalah kpi-name value untuk metrik untuk query.

Semua parameter harus URL dikodekan.

Untuk menerapkan filter yang query data untuk rentang tanggal tertentu, tambahkan `start-time` dan `end-time` parameter kueri dan nilai ke URI. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Gunakan ampersand (&) untuk memisahkan parameter.

Misalnya, permintaan berikut mengambil jumlah pesan email transaksional yang dikirim untuk proyek mulai 6 September 2019 hingga 13 September 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/1234567890123456789012345example/kpis/daterange/txn-emails-sent?start-time=2019-09-06T00:00:00Z&end-time=2019-09-13T23:59:59Z
```

Dengan:

- `pinpoint.us-east-1.amazonaws.com` adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menjadi tuan rumah proyek.
- `1234567890123456789012345example` Pengidentifikasi unik untuk proyek.
- `txn-emails-sent` adalah kpi-name value untuk mengutus metrik aplikasi, yang merupakan metrik yang melaporkan jumlah pesan email transaksional yang dikirim untuk proyek.
- `2019-09-06T00:00:00Z` adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- `2019-09-13T23:59:59Z` adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

AWS CLI

Mengkueri data analitik untuk pesan email transaksional dengan menggunakan AWS CLI, menggunakan `get-application-date-range-kpi` perintah, dan menentukan nilai yang sesuai untuk parameter yang diperlukan:

```
C:\> aws pinpoint get-application-date-range-kpi ^  
  --application-id application-id ^  
  --kpi-name kpi-name
```

Dengan:

- *Application-id* Pengidentifikasi unik untuk proyek.
- *kpi-name* adalah `kpi-name` value untuk metrik untuk query.

Untuk menerapkan filter yang mempertanyakan data untuk rentang tanggal tertentu, tambahkan `start-time` dan `end-time` parameter dan nilai untuk kueri Anda. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Misalnya, permintaan berikut mengambil jumlah pesan email transaksional yang dikirim untuk proyek mulai 6 September 2019 hingga 13 September 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^  
  --application-id 1234567890123456789012345example ^  
  --kpi-name txn-emails-sent ^  
  --start-time 2019-09-06T00:00:00Z ^  
  --end-time 2019-09-13T23:59:59Z
```

Dengan:

- 1234567890123456789012345example Pengidentifikasi unik untuk proyek.
- `txn-emails-sent` adalah `kpi-name` value untuk metrik aplikasi, yang merupakan metrik yang melaporkan jumlah pesan email transaksional yang dikirim untuk proyek.
- 2019-09-06T00:00:00Z adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- 2019-09-13T23:59:59Z adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

SDK for Java

Mengkueri data analitik untuk pesan email transaksional dengan menggunakan AWS SDK for Java, menggunakan `GetApplicationDateRangeKpiRequest` metode [Metrik Aplikasi](#) API. Tentukan nilai yang sesuai untuk parameter yang diperlukan:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("applicationId")
    .withKpiName("kpiName")
```

Dengan:

- *applicationId* Pengidentifikasi unik untuk proyek.
- *kpiName* adalah *kpi-name* value untuk metrik untuk query.

Untuk menerapkan filter yang meminta data untuk rentang tanggal tertentu, sertakan `startTime` dan `endTime` parameter dan nilai dalam kueri Anda. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Misalnya, permintaan berikut mengambil jumlah pesan email transaksional yang dikirim untuk proyek mulai 6 September 2019 hingga 13 September 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("1234567890123456789012345example")
    .withKpiName("txn-emails-sent")
    .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
    .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

Dengan:

- `1234567890123456789012345example` Pengidentifikasi unik untuk proyek.
- `txn-emails-sent` adalah *kpi-name* value untuk mengutus metrik aplikasi, yang merupakan metrik yang melaporkan jumlah pesan email transaksional yang dikirim untuk proyek.
- `2019-09-06T00:00:00Z` adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- `2019-09-13T23:59:59Z` adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

Setelah Anda mengirim kueri Anda, Amazon Pinpoint mengembalikan hasil kueri dalam respons JSON. Struktur hasil bervariasi tergantung pada metrik yang Anda tanyakan. Beberapa metrik hanya mengembalikan satu nilai. Misalnya, `txn-emails-sent` metrik aplikasi, yang digunakan dalam contoh sebelumnya, mengembalikan satu nilai—jumlah pesan email transaksional yang dikirim dari proyek. Dalam hal ini, respons JSON adalah sebagai berikut:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"1234567890123456789012345example",
    "EndTime":"2019-09-13T23:59:59Z",
    "KpiName":"txn-emails-sent",
    "KpiResult":{
      "Rows":[
        {
          "Values":[
            {
              "Key":"TxnEmailsSent",
              "Type":"Double",
              "Value":"62.0"
            }
          ]
        }
      ]
    },
    "StartTime":"2019-09-06T00:00:00Z"
  }
}
```

Metrik lain mengembalikan beberapa nilai dan mengelompokkan nilai dengan bidang yang relevan. Jika metrik mengembalikan beberapa nilai, respon JSON mencakup bidang yang menunjukkan bidang yang digunakan untuk mengelompokkan data.

Untuk mempelajari selengkapnya tentang struktur hasil kueri, lihat [Menggunakan hasil kueri](#).

Data kueri untuk pesan SMS transaksional

Untuk query data untuk pesan SMS transaksional yang dikirim untuk proyek, Anda menggunakan [Metrik Aplikasi](#) API dan tentukan parameter wajib berikut:

- `Application-id`— ID proyek, yang merupakan pengenalan unik untuk proyek. Di Amazon Pinpoint, istilahnya `proyek` dan `penerapan` memiliki arti yang sama.

- `kpi-name`— Nama metrik untuk query. Nilai ini menggambarkan metrik terkait dan terdiri dari dua atau lebih istilah, yang terdiri dari karakter alfanumerik huruf kecil, dipisahkan oleh tanda hubung. Untuk daftar lengkap terkait metrik yang didukung dan `kpi-name` nilai untuk masing-masing, lihat [Metrik standar](#).

Anda juga dapat menerapkan filter yang query data untuk rentang tanggal tertentu. Jika Anda tidak menentukan rentang tanggal, Amazon Pinpoint mengembalikan data 31 hari sebelumnya. Untuk menyaring data berdasarkan tanggal yang berbeda, gunakan parameter rentang tanggal yang didukung untuk menentukan tanggal dan waktu pertama serta tanggal dan waktu. Nilai-nilai harus dalam format ISO 8601 diperpanjang dan menggunakan Coordinated Universal Time (UTC) — misalnya, `2019-09-06T20:00:00Z` untuk 8:00 PM UTC 6 September 2019. Rentang tanggal sudah inklusif dan harus dibatasi hingga 31 hari kalender atau lebih sedikit. Selain itu, tanggal dan waktu pertama harus kurang dari 90 hari dari hari ini.

Contoh berikut menunjukkan cara mengkueri data analitik untuk pesan SMS transaksional dengan menggunakan Amazon Pinpoint REST API, AWS CLI, dan AWS SDK for Java. Anda dapat menggunakan apa pun yang didukung AWS SDK untuk query data analisis untuk pesan transaksional. Parameter AWS CLI contoh diformat untuk Microsoft Windows. Untuk Unix, Linux, dan macOS, ganti karakter kelanjutan baris tanda sisipan (^) dengan garis miring terbalik (\).

REST API

Untuk melakukan kueri data analitik untuk pesan SMS transaksional menggunakan Amazon Pinpoint REST API, kirim permintaan GET HTTP (S) ke [Metrik Aplikasi](#) URI. Dalam URI, tentukan nilai yang sesuai untuk parameter jalur yang diperlukan:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dengan:

- *titik akhir* adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menjadi tuan rumah proyek.
- *Application-id* Pengidentifikasi unik untuk proyek.
- *kpi-name* adalah `kpi-name` value untuk metrik untuk query.

Semua parameter harus URL dikodekan.

Untuk menerapkan filter yang mengambil data untuk rentang tanggal tertentu, tambahkan `start-time` dan `end-time` parameter kueri dan nilai ke URI. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Gunakan ampersand (&) untuk memisahkan parameter.

Misalnya, permintaan berikut mengambil jumlah pesan SMS transaksional yang dikirim setiap hari dari 6 September 2019 hingga 8 September 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/1234567890123456789012345example/kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-time=2019-09-08T23:59:59Z
```

Dengan:

- `pinpoint.us-east-1.amazonaws.com` adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menjadi tuan rumah proyek.
- `1234567890123456789012345example` Pengidentifikasi unik untuk proyek.
- `txn-sms-sent-grouped-by-date` adalah `kpi-name` value untuk mengirim, dikelompokkan berdasarkan tanggal metrik aplikasi, yang merupakan metrik yang mengembalikan jumlah pesan SMS transaksional yang dikirim selama setiap hari dari rentang tanggal.
- `2019-09-06T00:00:00Z` adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- `2019-09-08T23:59:59Z` adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

AWS CLI

Untuk query data analisis untuk pesan SMS transaksional dengan menggunakan AWS CLI, menggunakan `get-application-date-range-kpi` perintah, dan menentukan nilai yang sesuai untuk parameter yang diperlukan:

```
C:\> aws pinpoint get-application-date-range-kpi ^  
  --application-id application-id ^  
  --kpi-name kpi-name
```

Dengan:

- *Application-id* Pengidentifikasi unik untuk proyek.
- *kpi-name* adalah *kpi-name* value untuk metrik untuk query.

Untuk menerapkan filter yang mengambil data untuk rentang tanggal tertentu, sertakan *start-time* dan *end-time* parameter dan nilai dalam kueri Anda. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Misalnya, permintaan berikut mengambil jumlah pesan SMS transaksional yang dikirim setiap hari dari 6 September 2019 hingga 8 September 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
  --application-id 1234567890123456789012345example ^
  --kpi-name txn-sms-sent-grouped-by-date ^
  --start-time 2019-09-06T00:00:00Z ^
  --end-time 2019-09-08T23:59:59Z
```

Dengan:

- 1234567890123456789012345example Pengidentifikasi unik untuk proyek.
- *txn-sms-sent-grouped-by-date* adalah *kpi-name* value untuk mengirim, dikelompokkan berdasarkan tanggal metrik aplikasi, yang merupakan metrik yang mengembalikan jumlah pesan SMS transaksional yang dikirim selama setiap hari dari rentang tanggal.
- 2019-09-06T00:00:00Z adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- 2019-09-08T23:59:59Z adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

SDK for Java

Untuk query data analisis untuk pesan SMS transaksional dengan menggunakan AWS SDK for Java, menggunakan `GetApplicationDateRangeKpiRequest` metode [Metrik Aplikasi API](#), dan tentukan nilai yang sesuai untuk parameter yang diperlukan:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
    .withApplicationId("applicationId")
    .withKpiName("kpiName")
```

Dengan:

- *applicationId* Pengidentifikasi unik untuk proyek.
- *kpiName* adalah *kpi-name* value untuk metrik untuk query.

Untuk menerapkan filter yang mengambil data untuk rentang tanggal tertentu, sertakan *startTime* dan *endTime* parameter dan nilai dalam kueri Anda. Dengan menggunakan parameter ini, Anda dapat menentukan tanggal dan waktu pertama dan terakhir, dalam format ISO 8601 yang diperluas, dari rentang tanggal inklusif untuk mengambil data. Misalnya, permintaan berikut mengambil jumlah pesan SMS transaksional yang dikirim setiap hari dari 6 September 2019 hingga 8 September 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()  
    .withApplicationId("1234567890123456789012345example")  
    .withKpiName("txn-sms-sent-grouped-by-date")  
    .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))  
    .withEndTime(Date.from(Instant.parse("2019-09-08T23:59:59Z")));
```

Dengan:

- *1234567890123456789012345example* Pengidentifikasi unik untuk proyek.
- *txn-sms-sent-grouped-by-date* adalah *kpi-name* value untuk mengirim, dikelompokkan berdasarkan tanggal metrik aplikasi, yang merupakan metrik yang mengembalikan jumlah pesan SMS transaksional yang dikirim selama setiap hari dari rentang tanggal.
- *2019-09-06T00:00:00Z* adalah tanggal dan waktu pertama untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.
- *2019-09-08T23:59:59Z* adalah tanggal dan waktu terakhir untuk mengambil data untuk, sebagai bagian dari rentang tanggal inklusif.

Setelah Anda mengirim kueri Anda, Amazon Pinpoint mengembalikan hasil kueri dalam respons JSON. Struktur hasil bervariasi tergantung pada metrik yang Anda tanyakan. Beberapa metrik hanya mengembalikan satu nilai. Metrik lain mengembalikan beberapa nilai dan mengelompokkan nilai-nilai tersebut dengan bidang yang relevan. Jika metrik mengembalikan beberapa nilai, respon JSON mencakup bidang yang menunjukkan bidang yang digunakan untuk mengelompokkan data.

Misalnya, mengirim, dikelompokkan berdasarkan tanggal (*txn-sms-sent-grouped-by-date*) metrik aplikasi, yang digunakan dalam contoh sebelumnya, mengembalikan beberapa nilai-jumlah

pesan SMS transaksional yang dikirim selama setiap hari dari rentang tanggal yang ditentukan. Dalam hal ini, respons JSON adalah sebagai berikut:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"1234567890123456789012345example",
    "EndTime":"2019-09-08T23:59:59Z",
    "KpiName":"txn-sms-sent-grouped-by-date",
    "KpiResult":{
      "Rows":[
        {
          "GroupedBy":[
            {
              "Key":"Date",
              "Type":"String",
              "Value":"2019-09-06"
            }
          ],
          "Values":[
            {
              "Key":"TxnSmsSent",
              "Type":"Double",
              "Value":"29.0"
            }
          ]
        },
        {
          "GroupedBy":[
            {
              "Key":"Date",
              "Type":"String",
              "Value":"2019-09-07"
            }
          ],
          "Values":[
            {
              "Key":"TxnSmsSent",
              "Type":"Double",
              "Value":"35.0"
            }
          ]
        }
      ],
    }
  }
}
```

```
    "GroupedBy": [
      {
        "Key": "Date",
        "Type": "String",
        "Value": "2019-09-08"
      }
    ],
    "Values": [
      {
        "Key": "TxnSmsSent",
        "Type": "Double",
        "Value": "10.0"
      }
    ]
  }
},
"StartTime": "2019-09-06T00:00:00Z"
}
```

Dalam kasus ini, `GroupedBy` bidang menunjukkan bahwa nilai-nilai dikelompokkan berdasarkan hari kalender (`Date`). Ini artinya bahwa:

- 29 pesan dikirim pada tanggal 6 September 2019.
- 35 pesan dikirim pada tanggal 7 September 2019.
- 10 pesan dikirim pada 8 September 2019.

Untuk mempelajari selengkapnya tentang struktur hasil kueri, lihat [Menggunakan hasil kueri](#).

Menggunakan hasil kueri analitik Amazon Pinpoint

Saat Anda menggunakan API Amazon Pinpoint Analytics untuk kueri data analitik, Amazon Pinpoint mengembalikan hasil dalam respons JSON. Untuk metrik aplikasi, metrik kampanye, dan metrik keterlibatan perjalanan, data dalam respons mematuhi skema JSON standar untuk melaporkan data analitik Amazon Pinpoint.

Ini berarti bahwa Anda dapat menggunakan bahasa pemrograman atau alat pilihan Anda untuk menerapkan solusi kustom yang query data untuk satu atau lebih metrik ini, menangkap hasil dari

setiap kueri, dan kemudian menulis hasilnya ke tabel, objek, atau lokasi lainnya. Anda kemudian dapat bekerja dengan hasil query di lokasi itu dengan menggunakan layanan atau aplikasi lain.

Sebagai contoh, Anda dapat:

- Buat dasbor khusus yang menanyakan seperangkat metrik secara teratur dan tampilkan hasilnya dengan menggunakan kerangka visualisasi data pilihan Anda.
- Buat laporan yang melacak tingkat keterlibatan dengan menanyakan metrik yang sesuai dan menampilkan hasil dalam bagan atau jenis laporan lain yang Anda rancang.
- Mengurai dan menulis data analisis ke format penyimpanan tertentu, dan kemudian port hasilnya ke solusi penyimpanan jangka panjang.

Perhatikan bahwa API Amazon Pinpoint Analytics tidak dirancang untuk membuat atau menyimpan objek persisten yang kemudian dapat Anda baca atau gunakan dalam proyek Amazon Pinpoint atau akun Amazon Pinpoint Anda. Sebagai gantinya, API dirancang untuk membantu Anda mengambil data analitik dan mentransfer data tersebut ke layanan dan aplikasi lain untuk analisis, penyimpanan, atau pelaporan lebih lanjut. Mereka melakukan ini sebagian dengan menggunakan struktur respons dan skema JSON yang sama untuk semua data analitik yang dapat Anda kueri secara terprogram untuk metrik aplikasi, metrik kampanye, dan metrik keterlibatan perjalanan.

Topik ini menjelaskan struktur, objek, dan bidang dalam respons JSON terhadap kueri metrik aplikasi, metrik kampanye, atau metrik keterlibatan perjalanan. Untuk informasi tentang bidang dalam respons JSON terhadap kueri metrik eksekusi perjalanan atau metrik eksekusi aktivitas perjalanan, lihat [Metrik analitik Amazon Pinpoint standar](#).

Struktur JSON

Untuk membantu Anda mengurai dan menggunakan hasil kueri, API Amazon Pinpoint Analytics menggunakan struktur respons JSON yang sama untuk semua data analitik Amazon Pinpoint yang dapat Anda kueri secara terprogram untuk metrik aplikasi, metrik kampanye, dan metrik keterlibatan perjalanan. Setiap respon JSON menentukan nilai-nilai yang mendefinisikan query, seperti ID proyek (`ApplicationId`). Respon juga mencakup satu (dan hanya satu) `KpiResult` objek. Parameter `KpiResult` objek berisi hasil keseluruhan set untuk query.

MASING `KpiResult` objek berisi `Rows` objek. Ini adalah array objek yang berisi hasil query dan metadata yang relevan tentang nilai-nilai dalam hasil tersebut. Struktur dan isi dari `Rows` objek memiliki karakteristik umum sebagai berikut:

- Setiap baris hasil query adalah objek JSON terpisah, bernama `Values`, di `Rows` objek. Misalnya, jika query mengembalikan tiga nilai, `Rows` objek berisi tiga `Values` objek. `MASING` `Values` objek berisi hasil individu untuk query.
- Setiap kolom hasil query adalah milik `Values` objek yang berlaku untuk. Nama kolom disimpan dalam `Key` bidang `Values` objek.
- Untuk hasil kueri yang dikelompokkan, masing-masing `Values` objek memiliki terkait `GroupedBy` objek. Parameter `GroupedBy` objek menunjukkan bidang yang digunakan untuk mengelompokkan hasil. Hal ini juga menyediakan nilai pengelompokan untuk terkait `Values` objek.
- Jika hasil query untuk metrik adalah `null`, `Rows` objek kosong.

Di luar karakteristik umum ini, struktur dan isi `Rows` objek bervariasi tergantung pada metrik. Ini karena Amazon Pinpoint mendukung dua jenis metrik, metrik nilai tunggal dan metrik nilai ganda.

SEBUAH metrik nilai tunggalnya menyediakan satu nilai kumulatif. Contohnya adalah persentase pesan yang dikirimkan ke penerima oleh semua menjalankan kampanye. SEBUAH beberapa nilai metrik menyediakan lebih dari satu nilai dan kelompok nilai-nilai tersebut dengan bidang yang relevan. Contohnya adalah persentase pesan yang dikirimkan ke penerima untuk setiap menjalankan kampanye, dikelompokkan berdasarkan kampanye yang dijalankan.

Anda dapat dengan cepat menentukan apakah metrik adalah metrik nilai tunggal atau metrik nilai ganda dengan mengacu pada nama metrik. Jika nama tidak mengandung `grouped-by`, itu adalah metrik nilai tunggal. Jika tidak, itu adalah metrik nilai ganda. Untuk daftar lengkap metrik yang dapat Anda kueri secara terprogram, lihat [Metrik analitik Amazon Pinpoint standar](#).

Metrik nilai tunggal

Untuk metrik nilai tunggal, `Rows` objek berisi `Values` objek yang:

- Menentukan nama metrik yang dikueri.
- Memberikan nilai untuk metrik yang ditanyakan.
- Mengidentifikasi jenis data dari nilai yang dikembalikan.

Misalnya, respons JSON berikut berisi hasil kueri untuk metrik nilai tunggal. Metrik ini melaporkan jumlah titik akhir unik yang dikirimkan oleh semua kampanye yang terkait dengan proyek, mulai 1 Agustus 2019 hingga 31 Agustus 2019:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"1234567890123456789012345example",
    "EndTime":"2019-08-31T23:59:59Z",
    "KpiName":"unique-deliveries",
    "KpiResult":{
      "Rows":[
        {
          "Values":[
            {
              "Key":"UniqueDeliveries",
              "Type":"Double",
              "Value":"1368.0"
            }
          ]
        }
      ]
    },
    "StartTime":"2019-08-01T00:00:00Z"
  }
}
```

Dalam contoh ini, tanggapan menunjukkan bahwa semua kampanye proyek mengirimkan pesan ke 1,368 endpoint unik dari 1 Agustus 2019 hingga 31 Agustus 2019, di mana:

- Key adalah nama metrik yang nilainya ditentukan dalam `Value` Bidang (`UniqueDeliveries`).
- Type adalah jenis data dari nilai yang ditentukan dalam `Value` Bidang (`Double`).
- Value adalah nilai aktual untuk metrik yang ditanyakan, termasuk filter apa pun yang diterapkan (`1368.0`).

Jika hasil kueri untuk metrik nilai tunggal adalah null (tidak lebih besar dari atau sama dengan nol), `Rows` objek kosong. Amazon Pinpoint mengembalikan nilai nol untuk metrik jika tidak ada data yang akan dikembalikan untuk metrik. Misalnya:

```
{
  "ApplicationDateRangeKpiResponse":{
    "ApplicationId":"2345678901234567890123456example",
    "EndTime":"2019-08-31T23:59:59Z",
    "KpiName":"unique-deliveries",
    "KpiResult":{
```



```

        "Rows": [
            ]
        },
        "StartTime": "2019-08-01T00:00:00Z"
    }
}

```

Metrik nilai ganda

Struktur dan isi `Rows` objek untuk metrik nilai ganda sebagian besar sama dengan metrik nilai tunggal. Parameter `Rows` objek untuk metrik multiple-nilai juga berisi `Values` objek. Parameter `Values` objek menentukan nama ramah metrik yang ditanyakan, memberikan nilai untuk metrik itu, dan mengidentifikasi tipe data dari nilai itu.

Namun, `Rows` objek untuk metrik multiple-value juga berisi satu atau lebih `GroupedBy` objek. Ada satu `GroupedBy` objek untuk setiap `Values` objek dalam hasil kueri. Parameter `GroupedBy` objek menunjukkan bidang yang digunakan untuk mengelompokkan data dalam hasil dan tipe data dari bidang itu. Hal ini juga menunjukkan nilai pengelompokan untuk bidang itu (untuk terkait `Values` objek).

Misalnya, respons JSON berikut berisi hasil kueri untuk metrik nilai ganda yang melaporkan jumlah titik akhir unik yang dikirimkan pesan, untuk setiap kampanye yang terkait dengan proyek, mulai 1 Agustus 2019 hingga 31 Agustus 2019:

```

{
  "ApplicationDateRangeKpiResponse": {
    "ApplicationId": "1234567890123456789012345example",
    "EndTime": "2019-08-31T23:59:59Z",
    "KpiName": "unique-deliveries-grouped-by-campaign",
    "KpiResult": {
      "Rows": [
        {
          "GroupedBy": [
            {
              "Key": "CampaignId",
              "Type": "String",
              "Value": "80b8efd84042ff8d9c96ce2f8example"
            }
          ],
          "Values": [
            {

```

```
        "Key": "UniqueDeliveries",
        "Type": "Double",
        "Value": "123.0"
      }
    ]
  },
  {
    "GroupedBy": [
      {
        "Key": "CampaignId",
        "Type": "String",
        "Value": "810c7aab86d42fb2b56c8c966example"
      }
    ],
    "Values": [
      {
        "Key": "UniqueDeliveries",
        "Type": "Double",
        "Value": "456.0"
      }
    ]
  },
  {
    "GroupedBy": [
      {
        "Key": "CampaignId",
        "Type": "String",
        "Value": "42d8c7eb0990a57ba1d5476a3example"
      }
    ],
    "Values": [
      {
        "Key": "UniqueDeliveries",
        "Type": "Double",
        "Value": "789.0"
      }
    ]
  }
],
"StartTime": "2019-08-01T00:00:00Z"
}
```

Dalam contoh ini, tanggapan menunjukkan bahwa tiga kampanye proyek mengirimkan pesan ke titik akhir yang unik dari 1 Agustus 2019 hingga 31 Agustus 2019. Untuk setiap kampanye tersebut, rincian jumlah pengiriman adalah:

- Kampanye80b8efd84042ff8d9c96ce2f8examplemengirim pesan ke 123 endpoint unik.
- Kampanye810c7aab86d42fb2b56c8c966examplemengirim pesan ke 456 titik akhir yang unik.
- Kampanye42d8c7eb0990a57ba1d5476a3examplemengirim pesan ke 789 titik akhir yang unik.

Dimana struktur umum objek dan bidang adalah:

- `GroupedBy.Key`— Nama properti atau bidang yang menyimpan nilai pengelompokan yang ditentukan dalam `GroupedBy.ValueBidang` (`CampaignId`).
- `GroupedBy.Type`— Jenis data dari nilai yang ditentukan dalam `GroupedBy.ValueBidang` (`String`).
- `GroupedBy.Value`— Nilai aktual untuk bidang yang digunakan untuk mengelompokkan data, seperti yang ditentukan dalam `GroupedBy.Keybidang` (ID kampanye).
- `Values.Key`— Nama metrik yang nilainya ditentukan dalam `Values.ValueBidang` (`UniqueDeliveries`).
- `Values.Type`— Jenis data dari nilai yang ditentukan dalam `Values.ValueBidang` (`Double`).
- `Values.Value`— Nilai aktual untuk metrik yang ditanyakan, termasuk filter yang diterapkan.

Jika hasil kueri untuk metrik nilai ganda adalah nol (tidak lebih besar dari atau sama dengan nol) untuk proyek, kampanye, atau sumber daya tertentu, Amazon Pinpoint tidak mengembalikan objek atau bidang apa pun untuk sumber daya. Jika hasil kueri untuk metrik nilai ganda adalah null untuk semua sumber daya, Amazon Pinpoint mengembalikan yang kosong `Rows` objek.

Objek dan bidang JSON

Selain menentukan nilai-nilai yang mendefinisikan query, seperti ID proyek (`ApplicationId`), setiap respons JSON terhadap kueri metrik aplikasi, metrik kampanye, atau metrik keterlibatan perjalanan mencakup `KpiResult` objek. Objek ini berisi kumpulan hasil keseluruhan untuk kueri, yang dapat Anda parse untuk mengirim data analitik ke layanan atau aplikasi lain. `MASINGKpiResult` objek berisi beberapa atau semua objek standar berikut dan bidang, tergantung pada metrik.

Bidang objek	Deskripsi
Rows	Array objek yang berisi hasil set untuk query.
Rows.GroupedBy	Untuk metrik multiple-value, array bidang yang mendefinisikan bidang dan nilai-nilai yang digunakan untuk mengelompokkan data dalam hasil query.
Rows.GroupedBy.Key	Untuk metrik nilai ganda, nama properti atau bidang yang menyimpan nilai yang ditentukan dalam GroupedBy.Value Bidang.
Rows.GroupedBy.Type	Untuk metrik multiple-value, tipe data dari nilai yang ditentukan dalam GroupedBy s.Value Bidang.
Rows.GroupedBy.Value	Untuk metrik nilai ganda, nilai aktual untuk bidang yang digunakan untuk mengelompokkan data dalam hasil kueri. Nilai ini berkorelasi dengan terkaitValuesobjek.
Rows.Values	Array bidang yang berisi hasil query.
Rows.Values.Key	Nama metrik yang dikenal. Nilai metrik ditentukan dalam Values.Value Bidang.
Rows.Values.Type	Jenis data dari nilai yang ditentukan dalam Values.Value Bidang.
Rows.Values.Value	Nilai aktual untuk metrik yang ditanyakan, termasuk filter yang diterapkan.

Untuk informasi tentang bidang dalam respons JSON terhadap kueri metrik eksekusi perjalanan atau metrik eksekusi aktivitas perjalanan, lihat [Metrik analitik Amazon Pinpoint standar](#).

Mencatat panggilan API Amazon Pinpoint dengan AWS CloudTrail

Amazon Pinpoint terintegrasi dengan AWS CloudTrail, yang merupakan layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon Pinpoint. CloudTrail menangkap panggilan API untuk Amazon Pinpoint sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol Amazon Pinpoint dan panggilan kode ke operasi Amazon Pinpoint API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon Simple Storage Service (Amazon S3), termasuk peristiwa untuk Amazon Pinpoint. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru dengan menggunakan Riwayat acara di CloudTrail konsol. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon Pinpoint, alamat IP tempat permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengonfigurasi dan mengaktifkannya, lihat [Panduan AWS CloudTrail Pengguna](#).

Amazon Pinpoint informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas peristiwa yang didukung terjadi di Amazon Pinpoint, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat [Melihat peristiwa dengan riwayat CloudTrail acara](#).

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Amazon Pinpoint, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- [Gambaran umum untuk membuat jejak](#)

- [CloudTrail layanan dan integrasi yang didukung](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan:

- Apakah permintaan dibuat dengan root atau kredensi AWS Identity and Access Management pengguna.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat elemen [CloudTrail UserIdentity](#).

Anda dapat membuat jejak dan menyimpan file log Anda di bucket Amazon S3 selama yang Anda inginkan. Selain itu, Anda dapat menentukan aturan siklus hidup Amazon S3 untuk mengarsipkan atau menghapus file log secara otomatis. Secara default, berkas log Anda dienkripsi dengan menggunakan enkripsi sisi server (SSE) Amazon S3.

Untuk diberitahu tentang pengiriman file log, konfigurasi CloudTrail untuk mempublikasikan notifikasi Amazon SNS saat file log baru dikirimkan. Untuk informasi selengkapnya, lihat [Mengonfigurasi notifikasi Amazon SNS](#) untuk CloudTrail

Anda juga dapat menggabungkan file log Amazon Pinpoint dari AWS beberapa Wilayah dan AWS beberapa akun ke dalam satu bucket Amazon S3. Untuk informasi selengkapnya, lihat [Menerima file CloudTrail log dari beberapa wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#).

Anda dapat menggunakan CloudTrail untuk mencatat tindakan untuk API Amazon Pinpoint berikut:

- [Amazon Pinpoint API](#)
- [Amazon Pinpoint SMS dan API Suara](#)

Amazon Pinpoint Tindakan API yang dapat dicatat oleh CloudTrail

Amazon Pinpoint API mendukung pencatatan tindakan berikut sebagai peristiwa dalam file CloudTrail log:

- [CreateApp](#)
- [CreateCampaign](#)
- [CreateEmailTemplate](#)
- [CreateExportJob](#)
- [CreateImportJob](#)
- [CreateJourney](#)
- [CreatePushTemplate](#)
- [CreateRecommenderConfiguration](#)
- [CreateSegment](#)
- [CreateSmsTemplate](#)
- [CreateVoiceTemplate](#)
- [DeleteAdmChannel](#)
- [DeleteApnsChannel](#)
- [DeleteApnsSandboxChannel](#)
- [DeleteApnsVoipChannel](#)
- [DeleteApnsVoipSandboxChannel](#)
- [DeleteApp](#)
- [DeleteBaiduChannel](#)
- [DeleteCampaign](#)
- [DeleteEmailChannel](#)
- [DeleteEmailTemplate](#)
- [DeleteEndpoint](#)
- [DeleteEventStream](#)
- [DeleteGcmChannel](#)
- [DeleteJourney](#)

- [DeletePushTemplate](#)
- [DeleteRecommenderConfiguration](#)
- [DeleteSegment](#)
- [DeleteSmsChannel](#)
- [DeleteSmsTemplate](#)
- [DeleteUserEndpoints](#)
- [DeleteVoiceChannel](#)
- [DeleteVoiceTemplate](#)
- [GetAdmChannel](#)
- [GetApnsChannel](#)
- [GetApnsSandboxChannel](#)
- [GetApnsVoipChannel](#)
- [GetApnsVoipSandboxChannel](#)
- [GetApp](#)
- [GetApplicationDateRangeKpi](#)
- [GetApplicationSettings](#)
- [GetApps](#)
- [GetBaiduChannel](#)
- [GetCampaign](#)
- [GetCampaignActivities](#)
- [GetCampaignDateRangeKpi](#)
- [GetCampaignVersion](#)
- [GetCampaignVersions](#)
- [GetCampaigns](#)
- [GetChannels](#)
- [GetEmailChannel](#)
- [GetEmailTemplate](#)
- [GetEndpoint](#)
- [GetEventStream](#)

- [GetExportJob](#)
- [GetExportJobs](#)
- [GetGcmChannel](#)
- [GetImportJob](#)
- [GetImportJobs](#)
- [GetJourney](#)
- [GetJourneyDateRangeKpi](#)
- [GetJourneyExecutionActivityMetrics](#)
- [GetJourneyExecutionMetrics](#)
- [GetPushTemplate](#)
- [GetRecommenderConfiguration](#)
- [GetRecommenderConfigurations](#)
- [GetSegment](#)
- [GetSegmentExportJobs](#)
- [GetSegmentImportJobs](#)
- [GetSegmentVersion](#)
- [GetSegmentVersions](#)
- [GetSegments](#)
- [GetSmsChannel](#)
- [GetSmsTemplate](#)
- [GetUserEndpoints](#)
- [GetVoiceChannel](#)
- [GetVoiceTemplate](#)
- [ListJourneys](#)
- [ListTagsForResource](#)
- [ListTemplates](#)
- [ListTemplateVersions](#)
- [PhoneNumberValidate](#)
- [PutEventStream](#)

- [RemoveAttributes](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateAdmChannel](#)
- [UpdateApnsChannel](#)
- [UpdateApnsSandboxChannel](#)
- [UpdateApnsVoipChannel](#)
- [UpdateApnsVoipSandboxChannel](#)
- [UpdateApplicationSettings](#)
- [UpdateBaiduChannel](#)
- [UpdateCampaign](#)
- [UpdateEmailChannel](#)
- [UpdateEmailTemplate](#)
- [UpdateEndpoint](#)
- [UpdateEndpointsBatch](#)
- [UpdateGcmChannel](#)
- [UpdateJourney](#)
- [UpdateJourneyState](#)
- [UpdatePushTemplate](#)
- [UpdateRecommenderConfiguration](#)
- [UpdateSegment](#)
- [UpdateSmsChannel](#)
- [UpdateSmsTemplate](#)
- [UpdateTemplateActiveVersion](#)
- [UpdateVoiceChannel](#)
- [UpdateVoiceTemplate](#)

Tindakan Amazon Pinpoint API berikut tidak masuk: CloudTrail

- PutEvents

- `SendMessage`
- `SendUserMessages`

Amazon Pinpoint tindakan API email yang dapat dicatat CloudTrail

Amazon Pinpoint Email API mendukung pencatatan tindakan berikut sebagai peristiwa dalam file CloudTrail log:

- [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](#)

Tindakan API Email Amazon Pinpoint berikut tidak masuk: CloudTrail

- SendEmail

Amazon Pinpoint SMS dan voice API versi 1 tindakan yang dapat dicatat CloudTrail

Amazon Pinpoint SMS dan Voice versi 1 API mendukung pencatatan tindakan berikut sebagai peristiwa dalam file CloudTrail log:

- [CreateConfigurationSet](#)
- [CreateConfigurationSetEventDestination](#)
- [DeleteConfigurationSet](#)
- [DeleteConfigurationSetEventDestination](#)
- [GetConfigurationSetEventDestinations](#)
- [UpdateConfigurationSetEventDestination](#)

Tindakan Amazon Pinpoint SMS dan Voice versi 1 API berikut tidak masuk: CloudTrail

- SendVoiceMessage

Contoh: Entri file log Amazon Pinpoint

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa menunjukkan

satu permintaan dari sumber mana pun. Ini mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan GetCampaigns dan CreateCampaign tindakan Amazon Pinpoint API.

```
{
  "Records": [
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "GetCampaigns",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T00:56:48Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": true,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "application-id": "example71dfa4c1aab66332a5839798f",
        "page-size": "1000"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Jersey/${project.version} (HttpURLConnection 1.8.0_144)",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "principalId": "123456789012",
        "sessionContext": {
          "attributes": {
            "creationDate": "2018-02-02T16:55:29Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "type": "Root"
    }
  ],
  {
```

```
"awsRegion": "us-east-1",
"eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
"eventName": "CreateCampaign",
"eventSource": "pinpoint.amazonaws.com",
"eventTime": "2018-02-03T01:05:16Z",
"eventType": "AwsApiCall",
"eventVersion": "1.05",
"readOnly": false,
"recipientAccountId": "123456789012",
"requestID": "example1-b9bb-50fa-abdb-80f274981d60",
"requestParameters": {
  "Description": "****",
  "HoldoutPercent": 0,
  "IsPaused": false,
  "MessageConfiguration": "****",
  "Name": "****",
  "Schedule": {
    "Frequency": "ONCE",
    "IsLocalTime": true,
    "StartTime": "2018-02-03T00:00:00-08:00",
    "Timezone": "utc-08"
  },
  "SegmentId": "exampleda204adf991a80281aa0e591",
  "SegmentVersion": 1,
  "application-id": "example71dfa4c1aab66332a5839798f"
},
"responseElements": {
  "ApplicationId": "example71dfa4c1aab66332a5839798f",
  "CreationDate": "2018-02-03T01:05:16.425Z",
  "Description": "****",
  "HoldoutPercent": 0,
  "Id": "example54a654f80948680cbba240ede",
  "IsPaused": false,
  "LastModifiedDate": "2018-02-03T01:05:16.425Z",
  "MessageConfiguration": "****",
  "Name": "****",
  "Schedule": {
    "Frequency": "ONCE",
    "IsLocalTime": true,
    "StartTime": "2018-02-03T00:00:00-08:00",
    "Timezone": "utc-08"
  },
  "SegmentId": "example4da204adf991a80281example",
  "SegmentVersion": 1,
```

```

    "State": {
      "CampaignStatus": "SCHEDULED"
    },
    "Version": 1
  },
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-cli/1.14.9 Python/3.4.3 Linux/3.4.0+ botocore/1.8.34",
  "userIdentity": {
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "accountId": "123456789012",
    "arn": "arn:aws:iam::123456789012:user/userName",
    "principalId": "AIDAIHTRCDA62EXAMPLE",
    "type": "IAMUser",
    "userName": "userName"
  }
}
]
}

```

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateConfigurationSet dan CreateConfigurationSetEventDestination tindakan di Amazon Pinpoint SMS dan Voice API.

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIHTRCDA62EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/SampleUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "SampleUser"
      },
      "eventTime": "2018-11-06T21:45:55Z",
      "eventSource": "sms-voice.amazonaws.com",
      "eventName": "CreateConfigurationSet",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.0.1",
      "userAgent": "PostmanRuntime/7.3.0",
      "requestParameters": {
        "ConfigurationSetName": "MyConfigurationSet"
      }
    }
  ]
}

```

```

    },
    "responseElements":null,
    "requestID":"56dcc091-e20d-11e8-87d2-9994aexample",
    "eventID":"725843fc-8846-41f4-871a-7c52dexample",
    "readOnly":false,
    "eventType":"AwsApiCall",
    "recipientAccountId":"123456789012"
  },
  {
    "eventVersion":"1.05",
    "userIdentity":{
      "type":"IAMUser",
      "principalId":"AIDAIHTRCDA62EXAMPLE",
      "arn":"arn:aws:iam::111122223333:user/SampleUser",
      "accountId":"111122223333",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
      "userName":"SampleUser"
    },
    "eventTime":"2018-11-06T21:47:08Z",
    "eventSource":"sms-voice.amazonaws.com",
    "eventName":"CreateConfigurationSetEventDestination",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"192.0.0.1",
    "userAgent":"PostmanRuntime/7.3.0",
    "requestParameters":{
      "EventDestinationName":"CloudWatchEventDestination",
      "ConfigurationSetName":"MyConfigurationSet",
      "EventDestination":{
        "Enabled":true,
        "MatchingEventTypes":[
          "INITIATED_CALL",
          "INITIATED_CALL"
        ],
        "CloudWatchLogsDestination":{
          "IamRoleArn":"arn:aws:iam::111122223333:role/iamrole-01",
          "LogGroupArn":"arn:aws:logs:us-east-1:111122223333:log-
group:clientloggroup-01"
        }
      }
    },
    "responseElements":null,
    "requestID":"81de1e73-e20d-11e8-b158-d5536example",
    "eventID":"fcafc21f-7c93-4a3f-9e72-fca2dexample",
    "readOnly":false,

```



```
    "eventType": "AwsApiCall",  
    "recipientAccountId": "111122223333"  
  }  
]  
}
```

Memberi tag pada sumber daya Amazon Pinpoint

Tag adalah label yang secara opsional Anda tentukan dan kaitkan dengan AWS sumber daya, termasuk jenis sumber daya Amazon Pinpoint tertentu. Tag dapat membantu Anda mengategorikan dan mengelola sumber daya dengan cara yang berbeda, seperti berdasarkan tujuan, pemilik, atau kriteria lainnya. Misalnya, Anda dapat menggunakan tag untuk menerapkan kebijakan atau otomatisasi, atau untuk mengidentifikasi sumber daya yang tunduk pada persyaratan kepatuhan tertentu. Anda dapat menambahkan tag ke jenis sumber daya Amazon Pinpoint berikut:

- Kampanye
- Template pesan
- Proyek (aplikasi)
- Segmen

Sumber daya dapat memiliki sebanyak 50 tag.

Mengelola tag

Setiap tag terdiri atas sebuah kunci tag yang diperlukan dan sebuah nilai tag opsional, yang keduanya Anda tentukan. Kunci tag adalah label umum yang bertindak sebagai kategori untuk nilai tag yang lebih spesifik. Nilai tag bertindak sebagai deskriptor untuk kunci tag.

Kunci tag dapat berisi sebanyak 128 karakter. Sebuah nilai tag dapat berisi sebanyak 256 karakter. Karakter dapat berupa huruf Unicode, angka, spasi, atau salah satu simbol berikut: `_` `!` `/'` `+` `-`. Batasan tambahan berikut berlaku untuk tag:

- Kunci dan nilai tag peka huruf besar dan kecil.
- Untuk setiap sumber daya yang terkait, setiap kunci tag harus unik dan hanya dapat memiliki satu nilai.
- `aws` : Awalan dicadangkan untuk digunakan oleh AWS; Anda tidak dapat menggunakannya dalam kunci tag atau nilai apa pun yang Anda tentukan. Selain itu, Anda tidak dapat menyunting atau menghapus kunci atau nilai tag yang menggunakan awalan ini. Tag yang menggunakan awalan ini tidak dihitung terhadap kuota 50 tag per sumber daya.
- Anda tidak dapat memperbarui atau menghapus sumber daya hanya berdasarkan tagnya. Anda juga harus menentukan Amazon Resource Name (ARN) atau ID sumber daya, tergantung pada operasi yang Anda gunakan.

- Anda dapat mengaitkan tag dengan sumber daya publik atau bersama. Namun, tag hanya tersedia untuk AWS akun Anda, bukan akun lain yang berbagi sumber daya. Selain itu, tag hanya tersedia untuk sumber daya yang terletak di AWS Wilayah yang ditentukan untuk AWS akun Anda.

Untuk menambahkan, menampilkan, memperbarui, dan menghapus kunci dan nilai tag dari sumber daya Amazon Pinpoint, Anda dapat menggunakan AWS Command Line Interface (AWS CLI), Amazon Pinpoint API, API AWS Resource Groups Tagging, atau AWS SDK. Untuk mengelola kunci dan nilai tag di semua AWS sumber daya yang berada di AWS Wilayah tertentu untuk AWS akun Anda (termasuk sumber daya Amazon Pinpoint), gunakan [API AWS Resource Groups Penandaan](#).

Menggunakan tag dalam kebijakan IAM

Setelah Anda mulai menerapkan tag, Anda dapat menerapkan izin berbasis tag tingkat sumber daya ke kebijakan AWS Identity and Access Management (IAM) dan operasi API. Ini termasuk operasi yang mendukung penambahan tag ke sumber daya saat sumber daya dibuat. Dengan menggunakan tag dengan cara ini, Anda dapat menerapkan kontrol terperinci dari grup dan pengguna di AWS akun mana yang memiliki izin untuk membuat dan menandai sumber daya, dan grup dan pengguna mana yang memiliki izin untuk membuat, memperbarui, dan menghapus tag secara lebih umum.

Misalnya, Anda dapat membuat kebijakan yang memungkinkan pengguna memiliki akses penuh ke semua sumber daya Amazon Pinpoint di mana namanya adalah nilai dalam `Owner` tag untuk sumber daya:

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

Jika Anda menentukan izin berbasis tag tingkat sumber daya, izin akan segera berlaku. Ini berarti bahwa sumber daya Anda lebih aman segera setelah dibuat, dan Anda dapat dengan cepat mulai menerapkan penggunaan tag untuk sumber daya baru. Anda juga dapat menggunakan izin tingkat sumber daya untuk mengontrol kunci dan nilai tag mana yang dapat dikaitkan dengan sumber daya baru dan yang sudah ada. Untuk informasi selengkapnya, lihat [Mengontrol Akses Menggunakan Tag](#) di Panduan Pengguna AWS IAM.

Menambahkan tag ke sumber daya

Contoh berikut menunjukkan cara menambahkan tag ke sumber daya Amazon Pinpoint dengan menggunakan [AWS CLI](#) dan [Amazon Pinpoint REST API](#). Anda juga dapat menggunakan AWS SDK yang didukung untuk menambahkan tag ke sumber daya.

Untuk menambahkan tag ke beberapa sumber daya Amazon Pinpoint dalam satu operasi, gunakan operasi penandaan grup sumber daya dari API AWS CLI atau [AWS Resource Groups Tagging API](#).

Menambahkan tag dengan menggunakan API

Untuk membuat sumber daya baru dan menambahkan tag ke dalamnya dengan menggunakan Amazon Pinpoint REST API, kirim permintaan POST ke URI sumber daya yang sesuai.

Sertakan `tags` parameter dan nilai di badan permintaan. Contoh berikut menunjukkan cara menentukan tag saat Anda membuat proyek baru.

```
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"
  }
}
```

Untuk menambahkan tag ke sumber daya yang ada, kirim permintaan POST ke URI [Tag](#). Sertakan Amazon Resource Name (ARN) dari sumber daya di URI. ARN harus URL dikodekan. Di badan permintaan, sertakan `tags` parameter dan nilai, seperti yang ditunjukkan dalam contoh berikut.

```
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"
  }
}
```

Menambahkan tag dengan menggunakan AWS CLI

Untuk membuat sumber daya baru dan menambahkan tag ke dalamnya dengan menggunakan AWS CLI, gunakan `create` perintah yang sesuai untuk sumber daya. Sertakan `tags` parameter dan nilai. Contoh berikut menunjukkan cara menentukan tag saat Anda membuat proyek baru.

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}
```

Di contoh sebelumnya, lakukan hal berikut:

- Ganti *MyProject* dengan nama yang ingin Anda berikan pada proyek.
- Ganti *key1* dan *key2* dengan kunci tag yang ingin Anda tambahkan ke sumber daya.

- Ganti *value1* dan *value2* dengan nilai tag yang ingin Anda tambahkan untuk kunci masing-masing.

Untuk informasi tentang perintah yang dapat Anda gunakan untuk membuat sumber daya Amazon Pinpoint, lihat [Referensi AWS CLI Perintah](#).

Untuk menambahkan tag ke sumber daya yang ada, gunakan `tag-resource` perintah dan tentukan nilai yang sesuai untuk parameter yang diperlukan:

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}
```

Di contoh sebelumnya, lakukan hal berikut:

- Ganti *source* daya dengan Amazon Resource Name (ARN) sumber daya yang mana tandanya ingin Anda tambahkan.
- Ganti *key1* dan *key2* dengan kunci tag yang ingin Anda tambahkan ke sumber daya.
- Ganti *value1* dan *value2* dengan nilai tag yang ingin Anda tambahkan untuk kunci masing-masing.

Penunjukan tag untuk sumber daya

Contoh berikut menunjukkan cara menggunakan [AWS CLI](#) dan [Amazon Pinpoint REST API](#) untuk menampilkan daftar semua tag (kunci dan nilai) yang terkait dengan sumber daya Amazon Pinpoint.

Anda juga dapat menggunakan AWS SDK yang didukung untuk menampilkan tag yang terkait dengan sumber daya.

Menampilkan tag dengan menggunakan API

Untuk menggunakan Amazon Pinpoint REST API untuk menampilkan semua tag yang terkait dengan sumber daya tertentu, kirim permintaan GET ke URI [Tag](#) dan sertakan Amazon Resource Name (ARN) sumber daya dalam URI. ARN harus URL dikodekan. Misalnya, permintaan berikut mengambil semua tag yang terkait dengan kampanye tertentu (*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
```

Respons JSON terhadap permintaan mencakup `tags` objek. `tags` objek mencantumkan semua kunci dan nilai tag yang terkait dengan kampanye.

Untuk menampilkan semua tag yang terkait dengan lebih dari satu sumber daya dengan jenis yang sama, kirim permintaan GET ke URI yang sesuai untuk jenis sumber daya tersebut. Misalnya, permintaan berikut mengambil informasi tentang semua kampanye dalam proyek tertentu (*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
```

Respons JSON terhadap permintaan mencantumkan semua kampanye dalam proyek. `tags` objek setiap kampanye mencantumkan semua kunci tag dan nilai yang terkait dengan kampanye.

Menampilkan tag dengan menggunakan AWS CLI

Untuk menggunakan AWS CLI untuk menampilkan daftar tag yang terkait dengan sumber daya tertentu, jalankan `list-tags-for-resource` perintah dan tentukan Amazon Resource Name (ARN) sumber daya untuk `resource-arn` parameter, seperti yang ditunjukkan pada contoh berikut.

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
```

Untuk menampilkan daftar semua sumber daya Amazon Pinpoint yang memiliki tag, dan semua tag yang terkait dengan masing-masing sumber daya tersebut, gunakan perintah [get-resources](#) dari APIAWS Resource Groups Tagging. Mengatur `resource-type-filters` parameter untuk `mobiletargeting`, seperti yang ditunjukkan pada contoh berikut.

Linux, macOS, or Unix

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

Windows Command Prompt

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

Output perintah adalah daftar ARN untuk semua sumber daya Amazon Pinpoint yang memiliki tag. Daftar ini mencakup semua kunci tag dan nilai yang terkait dengan setiap sumber daya.

Memperbarui tag untuk sumber daya

Ada beberapa cara untuk memperbarui (menimpa) tag untuk sumber daya Amazon Pinpoint. Cara terbaik untuk memperbarui tag tergantung pada:

- Jenis sumber daya yang mana tandanya ingin Anda perbarui.
- Apakah Anda ingin memperbarui tag untuk satu atau beberapa sumber daya pada waktu yang sama.

- Apakah Anda ingin memperbarui kunci tag, nilai tag, atau keduanya.

Untuk memperbarui tag untuk proyek Amazon Pinpoint atau untuk beberapa sumber daya secara bersamaan, gunakan operasi penandaan grup sumber daya dari AWS CLI atau [API AWS Resource Groups Penandaan](#). Amazon Pinpoint API saat ini tidak memberikan dukungan langsung untuk salah satu tugas tersebut.

Untuk memperbarui tag untuk satu sumber daya, Anda dapat [menghapus tag saat ini](#) dan [menambahkan tag baru](#) menggunakan Amazon Pinpoint API.

Menghapus tag dari sumber daya

Contoh berikut menunjukkan cara menghapus tag (baik kunci maupun nilai) dari sumber daya Amazon Pinpoint dengan menggunakan [AWS CLI](#) dan [Amazon Pinpoint REST API](#). Anda juga dapat menggunakan AWS SDK yang didukung untuk menghapus tag dari sumber daya.

Untuk menghapus tag dari beberapa sumber daya Amazon Pinpoint dalam satu operasi, gunakan operasi penandaan grup sumber daya dari API AWS CLI atau [AWS Resource Groups Tagging API](#). Untuk menghapus hanya nilai tag tertentu (bukan kunci tag) dari sumber daya, [perbarui tag untuk sumber daya](#).

Menghapus tag dengan menggunakan API

Untuk menghapus tag dari sumber daya dengan menggunakan Amazon Pinpoint REST API, kirim permintaan DELETE ke URI [Tag](#). Di URI, sertakan Amazon Resource Name (ARN) dari sumber daya yang mana tandanya ingin Anda hapus. `tagKeys` Misalnya:

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

Dengan:

- *endpoint* adalah titik akhir Amazon Pinpoint untuk AWS Wilayah yang menghosting sumber daya.
- *sumber daya* adalah ARN dari sumber daya yang mana tandanya ingin Anda hapus.
- *kunci* adalah tag yang ingin Anda hapus dari sumber daya.

Semua parameter harus URL dikodekan.

Untuk menghapus beberapa kunci tag dan nilai terkait dari sumber daya, tambahkan `tagKeys` parameter dan argumen untuk setiap tag tambahan untuk dihapus, dipisahkan oleh ampersand (&). Misalnya:

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

Semua parameter harus URL dikodekan.

Menghapus tag dengan menggunakan AWS CLI

Untuk menghapus tag dari sumber daya menggunakan AWS CLI, jalankan `untag-resource` perintah. Sertakan `tag-keys` parameter dan argumen, seperti yang ditunjukkan dalam contoh berikut.

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
```

Pada contoh sebelumnya, lakukan perubahan berikut:

- Ganti *sumber daya* dengan ARN dari sumber daya yang mana tandanya ingin Anda hapus.
- Ganti *key1* dan *key2* dengan kunci tag yang ingin Anda hapus dari sumber daya.

Informasi terkait

Untuk informasi selengkapnya tentang perintah CLI yang dapat Anda gunakan untuk mengelola sumber daya Amazon Pinpoint, lihat bagian Amazon Pinpoint pada [Referensi AWS CLI Perintah](#).

Untuk informasi selengkapnya tentang sumber daya di Amazon Pinpoint API, termasuk metode, parameter, dan skema HTTP (S) yang didukung, lihat [Referensi API Amazon Pinpoint](#).

Menyesuaikan rekomendasi dengan AWS Lambda

Di Amazon Pinpoint, Anda dapat mengambil rekomendasi yang dipersonalisasi dari model recommender dan menambahkannya ke pesan yang Anda kirim dari kampanye dan perjalanan. SEBUAH Model recommender adalah jenis model machine learning (L) yang menemukan pola dalam data dan menghasilkan prediksi dan rekomendasi berdasarkan pola yang ditemukan. Ini memprediksi apa yang pengguna tertentu akan lebih memilih dari satu set tertentu produk atau item, dan memberikan informasi itu sebagai satu set rekomendasi bagi pengguna.

Dengan menggunakan model recommender dengan Amazon Pinpoint, Anda dapat mengirim rekomendasi yang dipersonalisasi ke penerima pesan berdasarkan atribut dan perilaku masing-masing penerima. Dengan AWS Lambda, Anda juga dapat menyesuaikan dan meningkatkan rekomendasi ini. Misalnya, Anda dapat mengubah rekomendasi secara dinamis dari satu nilai teks (seperti nama produk atau ID) menjadi konten yang lebih canggih (seperti nama produk, deskripsi, dan gambar). Dan Anda dapat melakukannya secara real time, ketika Amazon Pinpoint mengirimkan pesan.

Fitur ini tersedia dalam hal berikut AWS Wilayah: AS Timur (Virginia U.); AS Barat (Oregon); Asia Pasifik (Mumbai); Asia Pasifik (Sydney); dan, Eropa (Irlandia).

Topik

- [Menggunakan rekomendasi dalam pesan](#)
- [Buat fungsi Lambda](#)
- [Menetapkan kebijakan fungsi Lambda](#)
- [Mengotorisasi Amazon Pinpoint untuk memanggil fungsi](#)
- [Mengkonfigurasi Model Recommender](#)

Menggunakan rekomendasi dalam pesan

Untuk menggunakan model rekomendasi dengan Amazon Pinpoint, Anda mulai dengan membuat solusi Amazon Personalize dan menerapkan solusi tersebut sebagai kampanye Amazon Personalize. Kemudian, Anda membuat konfigurasi untuk model recommender di Amazon Pinpoint. Dalam konfigurasi, Anda menentukan pengaturan yang menentukan cara mengambil dan memproses data rekomendasi dari kampanye Amazon Personalize. Ini termasuk apakah akan memanggil AWS Lambda berfungsi untuk melakukan pemrosesan tambahan dari data yang diambil.

Amazon Personalize adalah AWS layanan yang dirancang untuk membantu Anda membuat model ML-nya yang memberikan rekomendasi yang dipersonalisasi secara real-time bagi pelanggan yang menggunakan aplikasi Anda. Amazon Personalize memandu Anda melalui proses membuat dan melatih model ML-nya, dan kemudian mempersiapkan dan menerapkan model sebagai kampanye Amazon Personalize. Anda kemudian dapat mengambil rekomendasi real-time yang dipersonalisasi dari kampanye. Untuk mempelajari selengkapnya tentang Amazon Personalize, lihat [Panduan Developer Amazon Personalize](#).

AWS Lambda adalah layanan komputasi yang dapat Anda gunakan untuk menjalankan kode tanpa perlu menyediakan atau mengelola server. Anda mengemas kode Anda dan mengunggahnya ke AWS Lambda sebagai Fungsi Lambda. AWS Lambda kemudian menjalankan fungsi saat fungsi tersebut dipanggil. Fungsi dapat dipanggil secara manual oleh Anda, secara otomatis dalam menanggapi peristiwa, atau dalam menanggapi permintaan dari aplikasi atau layanan, termasuk Amazon Pinpoint. Untuk informasi tentang membuat dan memanggil fungsi Lambda, lihat [Panduan Developer AWS Lambda](#).

Setelah membuat konfigurasi Amazon Pinpoint untuk model recommender, Anda dapat menambahkan rekomendasi dari model ke pesan yang Anda kirim dari kampanye dan perjalanan. Anda melakukan ini dengan menggunakan template pesan yang berisi variabel pesan untuk atribut yang direkomendasikan. SEBUAH atribut direkomendasikan adalah endpoint dinamis atau atribut pengguna yang dirancang untuk menyimpan data rekomendasi. Anda menentukan atribut ini ketika Anda membuat konfigurasi untuk model recommender.

Anda dapat menggunakan variabel untuk atribut yang direkomendasikan dalam jenis template pesan berikut:

- Templat email, untuk pesan email yang Anda kirim dari kampanye atau perjalanan.
- Templat notifikasi push, untuk notifikasi push yang Anda kirim dari kampanye.
- Template SMS, untuk pesan teks SMS yang Anda kirim dari kampanye.

Untuk informasi selengkapnya tentang menggunakan model rekomendasi dengan Amazon Pinpoint, lihat [Model Machine Learning](#) di Panduan Pengguna Amazon Pinpoint.

Jika Anda mengkonfigurasi Amazon Pinpoint untuk memanggil fungsi Lambda yang memproses data rekomendasi, Amazon Pinpoint melakukan tugas umum berikut setiap kali mengirimkan rekomendasi yang dipersonalisasi dalam pesan untuk kampanye atau perjalanan:

1. Mengevaluasi dan memproses pengaturan konfigurasi dan isi pesan dan pesan template.

2. Menentukan bahwa template pesan terhubung ke model recommender.
3. Mengevaluasi pengaturan konfigurasi untuk menghubungkan ke dan menggunakan model. Ini didefinisikan oleh [Model Recommenders](#) sumber daya untuk model.
4. Mendeteksi satu atau lebih variabel pesan untuk atribut yang direkomendasikan yang didefinisikan oleh pengaturan konfigurasi untuk model.
5. Mengambil data rekomendasi dari kampanye Amazon Personalize yang ditentukan dalam pengaturan konfigurasi untuk model. Menggunakan [GetRecommendations](#) pengoperasian Amazon Personalize Runtime API untuk melakukan tugas ini.
6. Menambahkan data rekomendasi yang sesuai ke atribut direkomendasikan dinamis (`RecommendationItems`) untuk setiap penerima pesan.
7. Memanggil fungsi Lambda Anda dan mengirimkan data rekomendasi untuk setiap penerima ke fungsi tersebut untuk diproses.

Data dikirim sebagai objek JSON yang berisi definisi endpoint untuk setiap penerima. Setiap definisi endpoint mencakup `RecommendationItems` bidang yang berisi array memerintahkan 1-5 nilai. Jumlah nilai dalam array tergantung pada pengaturan konfigurasi untuk model.

8. Menunggu fungsi Lambda Anda untuk memproses data dan mengembalikan hasilnya.

Hasilnya adalah objek JSON yang berisi definisi endpoint yang diperbarui untuk setiap penerima. Setiap definisi endpoint yang diperbarui berisi `Recommendations` objek. Objek ini berisi 1-10 bidang, satu untuk setiap atribut yang direkomendasikan khusus yang Anda tetapkan dalam pengaturan konfigurasi untuk model. Masing-masing bidang ini menyimpan data rekomendasi yang disempurnakan untuk titik akhir.

9. Menggunakan definisi endpoint yang diperbarui untuk setiap penerima untuk mengganti setiap variabel pesan dengan nilai yang sesuai untuk penerima tersebut.
10. Mengirim versi pesan yang berisi rekomendasi yang dipersonalisasi untuk setiap penerima pesan.

Untuk menyesuaikan dan meningkatkan rekomendasi dengan cara ini, mulailah dengan membuat fungsi Lambda yang memproses definisi endpoint yang dikirim oleh Amazon Pinpoint, dan mengembalikan definisi endpoint yang diperbarui. Selanjutnya, tetapkan kebijakan fungsi Lambda ke fungsi dan otorisasi Amazon Pinpoint untuk memanggil fungsi. Kemudian, konfigurasi model recommender di Amazon Pinpoint. Ketika Anda mengkonfigurasi model, menentukan fungsi untuk memanggil dan menentukan atribut direkomendasikan untuk digunakan.

Buat fungsi Lambda

Untuk mempelajari cara membuat fungsi Lambda, lihat [Memulai di AWS Lambda Panduan Pengembang](#). Ketika Anda merancang dan mengembangkan fungsi Anda, ingatlah persyaratan dan panduan berikut ini.

Data kejadian

Ketika Amazon Pinpoint memanggil fungsi Lambda untuk model recommender, Amazon Pinpoint akan mengirimkan muatan yang berisi konfigurasi dan pengaturan lain untuk kampanye atau perjalanan yang mengirim pesan. Muatan termasuk `Endpoint` subjek, yang merupakan peta yang mengaitkan ID endpoint dengan definisi endpoint untuk penerima pesan.

Definisi endpoint menggunakan struktur yang didefinisikan oleh [Titik akhir](#) sumber daya API Amazon Pinpoint. Namun, mereka juga menyertakan bidang untuk atribut direkomendasikan dinamis bernama `RecommendationItems`. Parameter `RecommendationItems` bidang berisi satu atau lebih item yang direkomendasikan untuk titik akhir, seperti yang dikembalikan dari kampanye Amazon Personalize. Nilai untuk bidang ini adalah array memerintahkan 1-5 item direkomendasikan (sebagai string). Jumlah item dalam array tergantung pada jumlah item yang direkomendasikan yang Anda konfigurasi Amazon Pinpoint untuk mengambil setiap titik akhir atau pengguna.

Misalnya:

```
"Endpoints": {
  "endpointIDexample-1":{
    "ChannelType":"EMAIL",
    "Address":"sofiam@example.com",
    "EndpointStatus":"ACTIVE",
    "OptOut":"NONE",
    "EffectiveDate":"2020-02-26T18:56:24.875Z",
    "Attributes":{
      "AddressType":[
        "primary"
      ]
    },
    "User":{
      "UserId":"SofiaMartínez",
      "UserAttributes":{
        "LastName":[
          "Martínez"
        ],

```

```
        "FirstName":[
            "Sofia"
        ],
        "Neighborhood":[
            "East Bay"
        ]
    }
},
"RecommendationItems":[
    "1815",
    "2009",
    "1527"
],
"CreationDate":"2020-02-26T18:56:24.875Z"
},
"endpointIDexample-2":{
    "ChannelType":"EMAIL",
    "Address":"alejandrор@example.com",
    "EndpointStatus":"ACTIVE",
    "OptOut":"NONE",
    "EffectiveDate":"2020-02-26T18:56:24.897Z",
    "Attributes":{
        "AddressType":[
            "primary"
        ]
    },
    "User":{
        "UserId":"AlejandroRosalez",
        "UserAttributes":{
            "LastName":[
                "Rosalez"
            ],
            "FirstName":[
                "Alejandro"
            ],
            "Neighborhood":[
                "West Bay"
            ]
        }
    },
    "RecommendationItems":[
        "1210",
        "6542",
        "4582"
    ]
}
```

```
    ],  
    "CreationDate":"2020-02-26T18:56:24.897Z"  
  }  
}
```

Dalam contoh sebelumnya, pengaturan Amazon Pinpoint adalah:

- Model recommender dikonfigurasi untuk mengambil tiga item yang direkomendasikan untuk setiap titik akhir atau pengguna. (Nilai untuk `RecommendationsPerMessage` properti diatur ke 3.) Dengan pengaturan ini, Amazon Pinpoint mengambil dan menambahkan hanya item pertama, kedua, dan ketiga yang direkomendasikan untuk setiap titik akhir atau pengguna.
- Proyek ini dikonfigurasi untuk menggunakan atribut pengguna kustom yang menyimpan nama depan setiap pengguna, nama belakang, dan lingkungan tempat mereka tinggal. (Yang `UserAttributes` objek berisi nilai-nilai untuk atribut ini.)
- Proyek ini dikonfigurasi untuk menggunakan atribut endpoint kustom (`AddressType`) yang menunjukkan apakah endpoint adalah alamat pilihan pengguna (saluran) untuk menerima pesan dari proyek. (Yang `Attributes` objek berisi nilai untuk atribut ini.)

Saat Amazon Pinpoint memanggil fungsi Lambda dan mengirimkan muatan ini sebagai data peristiwa, AWS Lambda melewati data ke fungsi Lambda untuk diproses.

Setiap muatan dapat berisi data hingga 50 titik akhir. Jika segmen berisi lebih dari 50 titik akhir, Amazon Pinpoint memanggil fungsi berulang kali, hingga 50 titik akhir sekaligus, hingga fungsi memproses semua data.

Data dan persyaratan respons

Saat Anda merancang dan mengembangkan fungsi Lambda Anda, pertahankan [kuota untuk model pembelajaran mesin](#) dalam pikiran. Jika fungsi tidak memenuhi ketentuan yang ditentukan oleh kuota ini, Amazon Pinpoint tidak akan dapat memproses dan mengirim pesan.

Juga ingatlah persyaratan berikut:

- Fungsi harus mengembalikan definisi endpoint yang diperbarui dalam format yang sama yang disediakan oleh data kejadian input.
- Setiap definisi endpoint yang diperbarui dapat berisi 1-10 atribut yang direkomendasikan khusus untuk endpoint atau pengguna. Nama atribut ini harus sesuai dengan nama atribut yang Anda tetapkan saat Anda mengonfigurasi model recommender di Amazon Pinpoint.

- Semua atribut kustom direkomendasikan harus dikembalikan dalam satu `Recommendations` objek untuk setiap endpoint atau pengguna. Persyaratan ini membantu memastikan bahwa konflik penamaan tidak terjadi. Anda dapat menambahkan `Recommendations` keberatan ke lokasi manapun dalam definisi endpoint.
- Nilai untuk setiap atribut kustom direkomendasikan harus string (nilai tunggal) atau array string (beberapa nilai). Jika nilainya adalah array string, sebaiknya Anda mempertahankan urutan item yang direkomendasikan yang dikembalikan oleh Amazon Personalize, seperti yang ditunjukkan dalam `RecommendationItems` bidang. Jika tidak, konten Anda mungkin tidak mencerminkan prediksi model untuk titik akhir atau pengguna.
- Fungsi tidak boleh memodifikasi elemen lain dalam data peristiwa, termasuk nilai atribut lainnya untuk endpoint atau pengguna. Ini hanya harus menambah dan mengembalikan nilai-nilai untuk atribut kustom direkomendasikan. Amazon Pinpoint tidak akan menerima pembaruan pada nilai lain dalam respons fungsi.
- Fungsi ini harus di-host dalam hal yang sama `AWS` Wilayah sebagai proyek Amazon Pinpoint yang memanggil fungsi. Jika fungsi dan proyek tidak berada di Wilayah yang sama, Amazon Pinpoint tidak dapat mengirim data peristiwa ke fungsi tersebut.

Jika salah satu persyaratan sebelumnya tidak terpenuhi, Amazon Pinpoint tidak akan dapat memproses dan mengirim pesan ke satu atau beberapa titik akhir. Hal ini dapat menyebabkan kampanye atau aktivitas perjalanan gagal.

Akhirnya, kami sarankan Anda memesan 256 eksekusi bersamaan untuk fungsi tersebut.

Secara keseluruhan, fungsi Lambda Anda harus memproses data peristiwa yang dikirim oleh Amazon Pinpoint dan mengembalikan definisi endpoint yang dimodifikasi. Hal ini dapat melakukan ini dengan iterasi melalui setiap titik akhir di `Endpoint` subjek dan, untuk setiap endpoint, membuat dan menetapkan nilai-nilai untuk atribut kustom direkomendasikan yang ingin Anda gunakan. Berikut contoh handler, ditulis dengan Python dan melanjutkan dengan contoh sebelumnya dari input data acara, menunjukkan ini:

```
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
```

Dalam contoh sebelumnya, AWS Lambda melewati data peristiwa ke handler sebagai event parameter. Handler iterates melalui setiap titik akhir di endpoint subjek dan menetapkan nilai untuk atribut kustom direkomendasikan bernama `Recommendations.Title` dan `Recommendations.Genre`. Parameter `return` pernyataan mengembalikan setiap definisi endpoint diperbarui ke Amazon Pinpoint.

Melanjutkan dengan contoh sebelumnya data peristiwa input, definisi endpoint diperbarui adalah:

```
"Endpoints":{
  "endpointIDexample-1":{
    "ChannelType":"EMAIL",
    "Address":"sofiam@example.com",
    "EndpointStatus":"ACTIVE",
    "OptOut":"NONE",
    "EffectiveDate":"2020-02-26T18:56:24.875Z",
    "Attributes":{
      "AddressType":[
        "primary"
      ]
    },
    "User":{
      "UserId":"SofiaMartínez",
      "UserAttributes":{
        "LastName":[
          "Martínez"
        ],
        "FirstName":[
          "Sofia"
        ],
        "Neighborhood":[
          "East Bay"
        ]
      }
    },
    "RecommendationItems":[
      "1815",
      "2009",
      "1527"
    ],
    "CreationDate":"2020-02-26T18:56:24.875Z",
    "Recommendations":{
      "Title":[
        "Fleabag",
```

```
        "Late Night",
        "Catastrophe"
    ],
    "Genre": [
        "Comedy",
        "Comedy",
        "Comedy"
    ]
}
},
"endpointIDexample-2": {
    "ChannelType": "EMAIL",
    "Address": "alejandr@example.com",
    "EndpointStatus": "ACTIVE",
    "OptOut": "NONE",
    "EffectiveDate": "2020-02-26T18:56:24.897Z",
    "Attributes": {
        "AddressType": [
            "primary"
        ]
    },
    "User": {
        "UserId": "AlejandroRosalez",
        "UserAttributes": {
            "LastName": [
                "Rosalez"
            ],
            "FirstName": [
                "Alejandro"
            ],
            "Neighborhood": [
                "West Bay"
            ]
        }
    },
    "RecommendationItems": [
        "1210",
        "6542",
        "4582"
    ],
    "CreationDate": "2020-02-26T18:56:24.897Z",
    "Recommendations": {
        "Title": [
            "Hanna",
```

```

        "Hunters",
        "Agatha Christie\'s The ABC Murders"
    ],
    "Genre":[
        "Action",
        "Drama",
        "Crime"
    ]
}
}
}
}

```

Dalam contoh sebelumnya, fungsi memodifikasi `Endpoint` subjek yang diterima dan mengembalikan hasilnya. Parameter `Endpoint` objek untuk setiap endpoint sekarang berisi baru `Recommendation` objek, yang berisi `Title` dan `Genre` bidang. Masing-masing bidang ini menyimpan array urutan tiga nilai (sebagai string), di mana setiap nilai menyediakan konten yang disempurnakan untuk item yang direkomendasikan sesuai di `RecommendationItem` bidang.

Menetapkan kebijakan fungsi Lambda

Sebelum Anda dapat menggunakan fungsi Lambda untuk memproses data rekomendasi, Anda harus mengotorisasi Amazon Pinpoint untuk memanggil fungsi tersebut. Untuk memberikan izin pemanggilan, tetapkan kebijakan fungsi Lambda ke fungsi. SEBUAH Kebijakan fungsi Lambda adalah kebijakan izin berbasis sumber daya yang menunjuk entitas mana yang dapat menggunakan fungsi dan tindakan apa yang dapat dilakukan entitas tersebut. Untuk informasi selengkapnya, lihat [Menggunakan Kebijakan Berbasis Sumber Daya untuk AWS Lambda](#) di AWS Lambda Panduan Pengembang.

Contoh kebijakan berikut memungkinkan prinsip layanan Amazon Pinpoint untuk menggunakan `lambda:InvokeFunction` tindakan untuk kampanye Amazon Pinpoint tertentu (*campaignId*) dalam proyek Amazon Pinpoint tertentu (*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/*"
  }
}
}

```

Kebijakan fungsi membutuhkan `Condition` blok yang mencakup `AWS:SourceArn` kunci. Kunci ini menentukan sumber daya yang diizinkan untuk memanggil fungsi. Dalam contoh sebelumnya, kebijakan memungkinkan satu kampanye tertentu untuk memanggil fungsi tersebut.

Anda juga dapat menulis kebijakan yang memungkinkan prinsip layanan Amazon Pinpoint untuk menggunakan `lambda:InvokeFunction` tindakan untuk semua kampanye dan perjalanan dalam proyek Amazon Pinpoint tertentu (*projectId*). Kebijakan berikut ini menunjukkan hal ini:

```

{
  "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/*"
    }
  }
}
}

```

Berbeda dengan contoh pertama, `AWS:SourceArn` kunci di `Condition` blok contoh ini memungkinkan satu proyek tertentu untuk memanggil fungsi. Izin ini berlaku untuk semua kampanye dan perjalanan dalam proyek.

Untuk menulis kebijakan yang lebih umum, Anda dapat menggunakan wildcard pencocokan multikarakter (*). Misalnya, Anda dapat menggunakan berikut ini `Condition` blok untuk memungkinkan proyek Amazon Pinpoint untuk memanggil fungsi:

```

"Condition": {
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}

```

```
}
}
```

Jika Anda ingin menggunakan fungsi Lambda dengan semua proyek untuk akun Amazon Pinpoint Anda, kami sarankan Anda mengonfigurasi `Condition` blok kebijakan dengan cara sebelumnya. Namun, sebagai praktik terbaik, Anda harus membuat kebijakan yang hanya mencakup izin yang diperlukan untuk melakukan tindakan tertentu pada sumber daya tertentu.

Mengotorisasi Amazon Pinpoint untuk memanggil fungsi

Setelah menetapkan kebijakan fungsi Lambda ke fungsi tersebut, Anda dapat menambahkan izin yang memungkinkan Amazon Pinpoint untuk memanggil fungsi untuk proyek, kampanye, atau perjalanan tertentu. Anda dapat melakukannya menggunakan `AWS Command Line Interface (AWS CLI)` dan Lambda `add-permission` perintah. Contoh berikut menunjukkan cara melakukannya untuk proyek tertentu (`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/*
```

Contoh sebelumnya diformat untuk Unix, Linux, dan macOS. Untuk Microsoft Windows, ganti karakter kelanjutan baris terbalik (`\`) dengan tanda sisipan (`^`).

Jika perintah berjalan dengan sukses, Anda melihat output yang serupa dengan berikut ini:

```
{
  "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/*\"}}}"
}
```

Nilai `Statement` adalah versi string JSON dari pernyataan yang ditambahkan ke kebijakan fungsi Lambda.

Mengkonfigurasi Model Recommender

Untuk mengkonfigurasi Amazon Pinpoint untuk memanggil fungsi Lambda untuk model recommender, tentukan pengaturan konfigurasi khusus Lambda berikut untuk model:

- `RecommendationTransformerUri`— Properti ini menentukan nama atau Amazon Resource Name (ARN) fungsi Lambda.
- `Attributes`- Objek ini adalah peta yang mendefinisikan atribut yang direkomendasikan khusus yang ditambahkan fungsi ke setiap definisi endpoint. Masing-masing atribut ini dapat digunakan sebagai variabel pesan dalam template pesan.

Anda dapat menentukan pengaturan ini dengan menggunakan [Model Recommender](#) sumber daya API Amazon Pinpoint (saat Anda membuat konfigurasi untuk model) atau [Model Recommenders](#) sumber daya API Amazon Pinpoint (jika Anda memperbarui konfigurasi untuk model). Anda juga dapat menentukan pengaturan ini dengan menggunakan konsol Amazon Pinpoint.

Untuk informasi selengkapnya tentang menggunakan model rekomendasi dengan Amazon Pinpoint, lihat [Model Machine Learning](#) di Panduan Pengguna Amazon Pinpoint.

Menghapus data dari Amazon Pinpoint

Bergantung pada bagaimana Anda menggunakannya, Amazon Pinpoint mungkin menyimpan data tertentu yang dapat dianggap pribadi. Misalnya, titik akhir di Amazon Pinpoint berisi informasi kontak untuk pengguna akhir, seperti alamat email atau nomor ponsel orang tersebut.

Anda dapat menggunakan konsol atau Amazon Pinpoint API untuk menghapus data pribadi secara permanen. Topik ini mencakup prosedur untuk menghapus berbagai jenis data yang dapat dianggap pribadi.

Menghapus titik akhir

Endpoint merupakan metode tunggal untuk menghubungi salah satu pelanggan Anda. Setiap titik akhir dapat merujuk ke alamat email pelanggan, pengenalan perangkat seluler, nomor telepon, atau jenis tujuan lain yang dapat Anda kirim pesan. Di banyak yurisdiksi, jenis informasi ini dapat dianggap pribadi.

Untuk menghapus semua data untuk titik akhir tertentu, Anda dapat menggunakan Amazon Pinpoint API untuk menghapus titik akhir. Prosedur berikut menunjukkan cara menghapus titik akhir dengan menggunakan AWS CLI untuk berinteraksi dengan Amazon Pinpoint API. Prosedur ini mengasumsikan Anda telah menginstal dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya, lihat [Menginstal AWS CLI](#) dalam Panduan Pengguna AWS Command Line Interface .

Untuk menghapus titik akhir dengan menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

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

Pada perintah sebelumnya, ganti *810c7aab86d42fb2b56c8c966example* dengan ID proyek yang terkait dengan titik akhir. Juga, ganti *ad015a3bf4f1b2b0b82example* dengan ID unik dari titik akhir itu sendiri.

Untuk menemukan ID titik akhir untuk titik akhir tertentu, tentukan segmen mana yang dimiliki titik akhir, lalu ekspor segmen tersebut dari Amazon Pinpoint. Data yang diekspor menyertakan ID titik akhir untuk setiap titik akhir. Anda dapat mengekspor segmen ke file dengan menggunakan konsol

Amazon Pinpoint. Untuk mempelajari caranya, lihat [Mengekspor Segmen](#) di Panduan Pengguna Amazon Pinpoint. Anda juga dapat mengekspor segmen ke bucket Amazon Simple Storage Service (Amazon S3) dengan menggunakan Amazon Pinpoint API. Untuk mempelajari caranya, lihat [Mengekspor Titik Akhir dalam panduan](#) ini.

Menghapus data segmen dan titik akhir yang disimpan di Amazon S3

Anda dapat mengimpor segmen dari file yang disimpan di bucket Amazon S3 menggunakan konsol Amazon Pinpoint atau API. Anda juga dapat mengekspor data aplikasi, segmen, atau titik akhir dari Amazon Pinpoint ke bucket Amazon S3. Baik file yang diimpor maupun yang diekspor dapat berisi data pribadi, termasuk alamat email, nomor ponsel, dan informasi tentang lokasi fisik titik akhir. Anda dapat menghapus file-file ini dari Amazon S3.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat [Bagaimana Cara Mengosongkan Bucket S3?](#) atau [Bagaimana Saya Menghapus Bucket S3?](#) .

Menghapus semua data proyek

Anda dapat menghapus semua data yang telah Anda simpan untuk proyek Amazon Pinpoint secara permanen. Anda dapat melakukan ini dengan menghapus proyek.

Warning

Jika Anda menghapus proyek, Amazon Pinpoint menghapus semua setelan dan data khusus proyek untuk proyek tersebut. Informasi tidak dapat dipulihkan.

Saat Anda menghapus proyek, Amazon Pinpoint menghapus semua setelan khusus proyek untuk notifikasi push dan saluran pesan SMS dua arah, serta semua segmen, kampanye, perjalanan, dan data analitik khusus proyek yang disimpan di Amazon Pinpoint, seperti berikut ini:

- **Segmen** — Semua pengaturan segmen dan data. Untuk segmen dinamis, ini termasuk grup segmen dan filter yang Anda tentukan. Untuk segmen yang diimpor, ini termasuk titik akhir, ID pengguna, dan data lain yang Anda impor, dan filter apa pun yang Anda terapkan.
- **Kampanye** — Semua pesan, perlakuan dan variabel pesan, data analitik, jadwal, dan pengaturan lainnya.

- Perjalanan — Semua aktivitas, data analitik, jadwal, dan pengaturan lainnya.
- Analytics — Data untuk semua metrik keterlibatan, seperti jumlah pesan yang dikirim dan dikirim untuk kampanye dan perjalanan, dan semua metrik eksekusi perjalanan. Untuk aplikasi seluler dan web, semua data peristiwa yang tidak dialirkan ke AWS layanan lain seperti Amazon Kinesis, semua saluran, dan data untuk penggunaan aplikasi, pendapatan, dan metrik demografis. Sebelum Anda menghapus proyek, kami sarankan Anda mengeksport data ini ke lokasi lain.

Anda dapat menghapus proyek dengan menggunakan konsol Amazon Pinpoint. Untuk mempelajari selengkapnya, lihat [Menghapus Proyek](#) di Panduan Pengguna Amazon Pinpoint. Anda juga dapat menghapus proyek secara terprogram dengan menggunakan sumber daya [App](#) dari Amazon Pinpoint API.

Menghapus semua AWS data dengan menutup akun Anda AWS

Anda juga dapat menghapus semua data yang telah Anda simpan di Amazon Pinpoint dengan menutup akun Anda AWS . Namun, tindakan ini juga menghapus semua data lain—pribadi atau non-pribadi—yang telah Anda simpan di setiap layanan lainnya. AWS Setelah periode pasca-penutupan, tutup AWS akun Anda AWS secara permanen, dan Anda tidak dapat lagi membukanya kembali. Konten apa pun yang tidak Anda hapus akan dihapus secara permanen, dan AWS layanan apa pun yang tidak Anda hentikan dihentikan. Untuk informasi selengkapnya lihat, [Tutup AWS akun](#) di Panduan AWS Account Management Referensi.


Warning

Prosedur berikut sepenuhnya menghapus semua data yang disimpan di AWS akun Anda di semua AWS layanan dan AWS Wilayah.

Anda dapat menutup AWS akun Anda dengan menggunakan AWS Management Console.

Untuk menutup AWS akun Anda

1. Buka AWS Management Console di <https://console.aws.amazon.com>.
2. Pergi ke halaman Pengaturan Akun di <https://console.aws.amazon.com/billing/home?#/account>.

 Warning

Langkah-langkah berikut secara permanen menghapus semua data yang telah Anda simpan di semua AWS layanan di semua AWS Wilayah.

3. Di bawah Tutup Akun, baca penafian yang menjelaskan konsekuensi dari penutupan akun Anda AWS . Jika Anda menyetujui persyaratan tersebut, pilih kotak centang, lalu pilih Tutup Akun.
4. Pada kotak dialog konfirmasi, pilih Tutup Akun.

Contoh kode untuk Amazon Pinpoint menggunakan AWS SDK

Contoh kode berikut ini menunjukkan cara menggunakan Amazon Pinpoint dengan AWS perangkat pengembangan perangkat lunak (SDK).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- [Contoh kode untuk Amazon Pinpoint menggunakan SDK AWS](#)
 - [Tindakan untuk Amazon Pinpoint menggunakan SDK AWS](#)
 - [Gunakan CreateApp dengan AWS SDK atau CLI](#)
 - [Gunakan CreateCampaign dengan AWS SDK atau CLI](#)
 - [Gunakan CreateExportJob dengan AWS SDK atau CLI](#)
 - [Gunakan CreateImportJob dengan AWS SDK atau CLI](#)
 - [Gunakan CreateSegment dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteApp dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteEndpoint dengan AWS SDK atau CLI](#)
 - [Gunakan GetEndpoint dengan AWS SDK atau CLI](#)
 - [Gunakan GetSegments dengan AWS SDK atau CLI](#)
 - [Gunakan GetSmsChannel dengan AWS SDK atau CLI](#)
 - [Gunakan GetUserEndpoints dengan AWS SDK atau CLI](#)
 - [Gunakan SendMessages dengan AWS SDK atau CLI](#)
 - [Gunakan UpdateEndpoint dengan AWS SDK atau CLI](#)
 - [Contoh kode untuk Amazon Pinpoint SMS dan Voice API menggunakan SDK AWS](#)
 - [Tindakan untuk Amazon Pinpoint SMS dan Voice API menggunakan SDK AWS](#)
 - [Gunakan SendVoiceMessage dengan AWS SDK atau CLI](#)

Contoh kode untuk Amazon Pinpoint menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menggunakan Amazon Pinpoint dengan AWS perangkat pengembangan perangkat lunak (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks pada skenario terkait dan contoh lintas layanan.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- [Tindakan untuk Amazon Pinpoint menggunakan SDK AWS](#)
 - [Gunakan CreateApp dengan AWS SDK atau CLI](#)
 - [Gunakan CreateCampaign dengan AWS SDK atau CLI](#)
 - [Gunakan CreateExportJob dengan AWS SDK atau CLI](#)
 - [Gunakan CreateImportJob dengan AWS SDK atau CLI](#)
 - [Gunakan CreateSegment dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteApp dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteEndpoint dengan AWS SDK atau CLI](#)
 - [Gunakan GetEndpoint dengan AWS SDK atau CLI](#)
 - [Gunakan GetSegments dengan AWS SDK atau CLI](#)
 - [Gunakan GetSmsChannel dengan AWS SDK atau CLI](#)
 - [Gunakan GetUserEndpoints dengan AWS SDK atau CLI](#)
 - [Gunakan SendMessages dengan AWS SDK atau CLI](#)
 - [Gunakan UpdateEndpoint dengan AWS SDK atau CLI](#)

Tindakan untuk Amazon Pinpoint menggunakan SDK AWS

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon Pinpoint individual dengan AWS SDK. Kutipan ini memanggil Amazon Pinpoint API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi [API Amazon Pinpoint](#).

Contoh

- [Gunakan CreateApp dengan AWS SDK atau CLI](#)
- [Gunakan CreateCampaign dengan AWS SDK atau CLI](#)
- [Gunakan CreateExportJob dengan AWS SDK atau CLI](#)
- [Gunakan CreateImportJob dengan AWS SDK atau CLI](#)
- [Gunakan CreateSegment dengan AWS SDK atau CLI](#)
- [Gunakan DeleteApp dengan AWS SDK atau CLI](#)
- [Gunakan DeleteEndpoint dengan AWS SDK atau CLI](#)
- [Gunakan GetEndpoint dengan AWS SDK atau CLI](#)
- [Gunakan GetSegments dengan AWS SDK atau CLI](#)
- [Gunakan GetSmsChannel dengan AWS SDK atau CLI](#)
- [Gunakan GetUserEndpoints dengan AWS SDK atau CLI](#)
- [Gunakan SendMessages dengan AWS SDK atau CLI](#)
- [Gunakan UpdateEndpoint dengan AWS SDK atau CLI](#)

Gunakan **CreateApp** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateApp`.

CLI

AWS CLI

Contoh 1: Untuk membuat aplikasi

`create-app` Contoh berikut membuat aplikasi baru (proyek).

```
aws pinpoint create-app \  
  --create-application-request Name=ExampleCorp
```

Output:

```
{  
  "ApplicationResponse": {
```

```

    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {}
  }
}

```

Contoh 2: Untuk membuat aplikasi yang diberi tag

create-app Contoh berikut membuat aplikasi baru (proyek) dan mengaitkan tag (kunci dan nilai) dengan aplikasi.

```

aws pinpoint create-app \
  --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}

```

Output:

```

{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {
      "Stack": "Test"
    }
  }
}

```

- Untuk detail API, lihat [CreateApp](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).


```
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 "";
}
}
```

- Untuk detail API, lihat [CreateApp](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun createApplication(applicationName: String?): String? {

    val createApplicationRequestObj = CreateApplicationRequest {
        name = applicationName
    }
}
```

```
PinpointClient { region = "us-west-2" }.use { pinpoint ->
    val result = pinpoint.createApp(
        CreateAppRequest {
            createApplicationRequest = createApplicationRequestOb
        }
    )
    return result.applicationResponse?.id
}
}
```

- Untuk detail API, lihat [CreateApp](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateCampaign** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateCampaign`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Buat kampanye.

```
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;
    }
}
```

```
}
```

- Untuk detail API, lihat [CreateCampaign](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun createPinCampaign(appId: String, segmentIdVal: String) {  
  
    val schedule0b = Schedule {  
        startTime = "IMMEDIATE"  
    }  
  
    val defaultMessage0b = Message {  
        action = Action.OpenApp  
        body = "My message body"  
        title = "My message title"  
    }  
  
    val messageConfiguration0b = MessageConfiguration {  
        defaultMessage = defaultMessage0b  
    }  
  
    val writeCampaign = WriteCampaignRequest {  
        description = "My description"  
        schedule = schedule0b  
        name = "MyCampaign"  
        segmentId = segmentIdVal  
        messageConfiguration = messageConfiguration0b  
    }  
  
    PinpointClient { region = "us-west-2" }.use { pinpoint ->  
        val result: CreateCampaignResponse = pinpoint.createCampaign(  
            CreateCampaignRequest {
```

```
        applicationId = appId
        writeCampaignRequest = writeCampaign
    }
)
println("Campaign ID is ${result.campaignResponse?.id}")
}
}
```

- Untuk detail API, lihat [CreateCampaign](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateExportJob** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateExportJob`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Ekspor titik akhir.

```
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);
    }
}
}
```

- Untuk detail API, lihat [CreateExportJob](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateImportJob** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateImportJob`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Impor segmen.

```
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;
    }
}
```

- Untuk detail API, lihat [CreateImportJob](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateSegment** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateSegment`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
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;
    }
}

```

- Untuk detail API, lihat [CreateSegment](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

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 dimensions0b = SegmentDimensions {
    attributes = segmentAttributes
    behavior = segmentBehaviors
    demographic = SegmentDemographics {}
    location = segmentLocation
}

val writeSegmentRequest0b = WriteSegmentRequest {
    name = "MySegment101"
    dimensions = dimensions0b
}

PinpointClient { region = "us-west-2" }.use { pinpoint ->
    val createSegmentResult: CreateSegmentResponse = pinpoint.createSegment(
        CreateSegmentRequest {
            applicationId = applicationIdVal
            writeSegmentRequest = writeSegmentRequest0b
        }
    )
    println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
    return createSegmentResult.segmentResponse?.id
}
}

```

- Untuk detail API, lihat [CreateSegment](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteApp** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteApp`.

CLI

AWS CLI

Untuk menghapus aplikasi

delete-app Contoh berikut menghapus aplikasi (proyek).

```
aws pinpoint delete-app \  
  --application-id 810c7aab86d42fb2b56c8c966example
```

Output:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",  
    "tags": {}  
  }  
}
```

- Untuk detail API, lihat [DeleteApp](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Hapus aplikasi.

```
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);
    }
}
}
```

- Untuk detail API, lihat [DeleteApp](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun deletePinApp(appId: String?) {

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.deleteApp(
            DeleteAppRequest {
                applicationId = appId
            }
        )
        val appName = result.applicationResponse?.name
        println("Application $appName has been deleted.")
    }
}
```

- Untuk detail API, lihat [DeleteApp](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteEndpoint** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteEndpoint`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Hapus titik akhir.

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

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

                Usage:  <appName> <endpointId >

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



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

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

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

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

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

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

- Untuk detail API, lihat [DeleteEndpoint](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun deletePinEndpoint(appIdVal: String?, endpointIdVal: String?) {  
  
    val deleteEndpointRequest = DeleteEndpointRequest {  
        applicationId = appIdVal  
        endpointId = endpointIdVal  
    }  
  
    PinpointClient { region = "us-west-2" }.use { pinpoint ->  
        val result = pinpoint.deleteEndpoint(deleteEndpointRequest)  
        val id = result.endpointResponse?.id  
        println("The deleted endpoint is $id")  
    }  
}
```

- Untuk detail API, lihat [DeleteEndpoint](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetEndpoint** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetEndpoint`.

CLI

AWS CLI

Untuk mengambil informasi tentang pengaturan dan atribut titik akhir tertentu untuk aplikasi

get-endpoint Contoh berikut mengambil informasi tentang pengaturan dan atribut titik akhir tertentu untuk aplikasi.

```
aws pinpoint get-endpoint \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --endpoint-id testendpoint \  
  --region us-east-1
```

Output:

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

- Untuk detail API, lihat [GetEndpoint](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
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");
}
}
```

- Untuk detail API, lihat [GetEndpoint](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun lookupPinpointEndpoint(appId: String?, endpoint: String?) {

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.getEndpoint(
            GetEndpointRequest {
                applicationId = appId
                endpointId = endpoint
            }
        )
        val endResponse = result.endpointResponse

        // Uses the Google Gson library to pretty print the endpoint JSON.
        val gson: com.google.gson.Gson = GsonBuilder()
            .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
            .setPrettyPrinting()
            .create()

        val endpointJson: String = gson.toJson(endResponse)
        println(endpointJson)
    }
}
```

- Untuk detail API, lihat [GetEndpoint](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetSegments** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetSegments`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Daftar segmen.

```
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);
    }
}
}
```

- Untuk detail API, lihat [GetSegments](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun listSegs(appId: String?) {  
  
    PinpointClient { region = "us-west-2" }.use { pinpoint ->  
  
        val response = pinpoint.getSegments(  
            GetSegmentsRequest {  
                applicationId = appId  
            }  
        )  
        response.segmentsResponse?.item?.forEach { segment ->  
            println("Segement id is ${segment.id}")  
        }  
    }  
}
```

- Untuk detail API, lihat [GetSegments](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetSmsChannel** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetSmsChannel`.

CLI

AWS CLI

Untuk mengambil informasi tentang status dan pengaturan saluran SMS untuk suatu aplikasi

`get-sms-channel` Contoh berikut mengambil status dan pengaturan saluran sms untuk aplikasi.

```
aws pinpoint get-sms-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Output:

```
{  
  "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  
  }  
}
```

- Untuk detail API, lihat [GetSmsChannel](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
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);
    }
}
}
```

- Untuk detail API, lihat [GetSmsChannel](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetUserEndpoints** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetUserEndpoints`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
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);
    }
}
}
```

- Untuk detail API, lihat [GetUserEndpoints](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **SendMessage** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `SendMessage`.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Kirim pesan email.

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

namespace SendMessage;

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;
    }
}
```

Kirim pesan 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;
}
}

```

- Untuk detail API, lihat [SendMessages](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Untuk mengirim pesan SMS menggunakan titik akhir aplikasi

`send-messages` Contoh berikut mengirimkan pesan langsung untuk aplikasi dengan titik akhir.

```
aws pinpoint send-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --message-request file://myfile.json \  
  --region us-west-2
```

Isi dari myfile.json:

```
{  
  "MessageConfiguration": {  
    "SMSMessage": {  
      "Body": "hello, how are you?"  
    }  
  },  
  "Endpoints": {  
    "testendpoint": {}  
  }  
}
```

Output:

```
{  
  "MessageResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "EndpointResult": {  
      "testendpoint": {  
        "Address": "+12345678900",  
        "DeliveryStatus": "SUCCESSFUL",  
        "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb31q6g0",  
        "StatusCode": 200,  
        "StatusMessage": "MessageId:  
itnuqhai5alf1n6ahv3udc05n7hhddr6gb31q6g0"  
      }  
    },  
    "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"  
  }  
}
```

Untuk informasi selengkapnya, lihat [saluran SMS Amazon Pinpoint](#) di Panduan Pengguna Amazon Pinpoint.

- Untuk detail API, lihat [SendMessage](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Kirim pesan email.

```
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);
        }
    }
}
```

Kirim pesan email dengan nilai 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();
}
}
```

Kirim pesan SMS.

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

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

```

* environment, including your credentials.
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class SendMessage {

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

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

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

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

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

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

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

```

```
        System.exit(1);
    }

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

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

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

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

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

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

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

        SendMessagesResponse response =
pinpoint.sendMessage(request);
        MessageResponse msg1 = response.getMessageResponse();
        Map map1 = msg1.getResult();

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

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

Kirim pesan SMS batch.

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

```
import java.util.Map;

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

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

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

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

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

            Usage:  <message> <appId> <originationNumber>
<destinationNumber> <destinationNumber1>\s

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



```
        destinationNumber1 - The second recipient's
phone number. For best results, you should specify the phone number in E.164
format (for example, +1-555-555-5654).\s
        """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String message = args[0];
        String appId = args[1];
        String originationNumber = args[2];
        String destinationNumber = args[3];
        String destinationNumber1 = args[4];
        System.out.println("Sending a message");
        PinpointClient pinpoint = PinpointClient.builder()
            .region(Region.US_EAST_1)
            .build();

        sendSMSMessage(pinpoint, message, appId, originationNumber,
            destinationNumber, destinationNumber1);
        pinpoint.close();
    }

    public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
        String originationNumber,
        String destinationNumber, String destinationNumber1) {
        try {
            Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
            AddressConfiguration addConfig =
AddressConfiguration.builder()
                .channelType(ChannelType.SMS)
                .build();

            // Add an entry to the Map object for each number to whom
you want to send a
            // message.
            addressMap.put(destinationNumber, addConfig);
            addressMap.put(destinationNumber1, addConfig);
            SMSMessage smsMessage = SMSMessage.builder()
                .body(message)
```

```

        .messageType(messageType)
        .originationNumber(originationNumber)
        .senderId(senderId)
        .keyword(registeredKeyword)
        .build();

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

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

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

        SendMessagesResponse response =
pinpoint.sendMessage(request);
        MessageResponse msg1 = response.getMessageResponse();
        Map map1 = msg1.getResult();

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

        } catch (PinpointException e) {
            System.err.println(e.getAwsErrorDetails().getErrorMessage());
            System.exit(1);
        }
    }
}

```

- Untuk detail API, lihat [SendMessage](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Buat klien dalam modul terpisah dan ekspor klien tersebut.

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

Kirim pesan email.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "../libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

// The subject line of the email.
var subject = "Amazon Pinpoint Test (AWS SDK for JavaScript in Node.js)";

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

// The body of the email for recipients whose email clients support HTML content.
var body_html = `
<head></head>
```

```
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint Email API</a>
    using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;
```

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

```
const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};
```

```
const run = async () => {
```

```
try {
  const { MessageResponse } = await pinClient.send(
    new SendMessagesCommand(params),
  );

  if (!MessageResponse) {
    throw new Error("No message response.");
  }

  if (!MessageResponse.Result) {
    throw new Error("No message result.");
  }

  const recipientResult = MessageResponse.Result[toAddress];

  if (recipientResult.StatusCode !== 200) {
    throw new Error(recipientResult.StatusMessage);
  } else {
    console.log(recipientResult.MessageId);
  }
} catch (err) {
  console.log(err.message);
}
};

run();
```

Kirim pesan SMS.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "../libs/pinClient.js";

/* The phone number or short code to send the message from. The phone number
   or short code that you specify has to be associated with your Amazon Pinpoint
   account. For best results, specify long codes in E.164 format. */
const originationNumber = "SENDER_NUMBER"; //e.g., +1XXXXXXXXXX

// The recipient's phone number. For best results, you should specify the phone
   number in E.164 format.
const destinationNumber = "RECEIVER_NUMBER"; //e.g., +1XXXXXXXXXX
```

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

/*The Amazon Pinpoint project/application ID to use when you send this message.
Make sure that the SMS channel is enabled for the project or application
that you choose.*/
const projectId = "PINPOINT_PROJECT_ID"; //e.g., XXXXXXXX66e4e9986478cXXXXXXXXXX

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

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

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

var senderId = "MySenderId";

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
}
```

```
    },
  },
};

const run = async () => {
  try {
    const data = await pinClient.send(new SendMessagesCommand(params));
    console.log(
      "Message sent! " +
      data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"],
    );
  } catch (err) {
    console.log(err);
  }
};
run();
```

- Untuk detail API, lihat [SendMessages](#) di Referensi AWS SDK for JavaScript API.

SDK untuk JavaScript (v2)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Kirim pesan email.

```
"use strict";

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

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

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

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

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

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

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

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

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

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

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



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

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

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

```
}  
});
```

Kirim pesan 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.sendMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
```

```

    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});

```

- Untuk detail API, lihat [SendMessages](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

/**
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 destination0b = Destination {  
        toAddresses = listOf(toAddressVal)  
    }  
}
```

```
val emailContent = EmailContent {
    simple = message
}

val sendEmailRequest = SendEmailRequest {
    fromEmailAddress = senderAddress
    destination = destination0b
    this.content = emailContent
}

PinpointEmailClient { region = "us-east-1" }.use { pinpointemail ->
    pinpointemail.sendEmail(sendEmailRequest)
    println("Message Sent")
}
}
```

- Untuk detail API, lihat [SendMessages](#) di AWS SDK untuk referensi API Kotlin.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Kirim pesan email.

```
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()
```

Kirim pesan 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()
```

Kirim pesan email dengan template email yang ada.

```
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()
```

Kirim pesan teks dengan template SMS yang ada.

```
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()
```

- Untuk detail API, lihat [SendMessage](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **UpdateEndpoint** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `UpdateEndpoint`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
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;
}
}
```

- Untuk detail API, lihat [UpdateEndpoint](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode untuk Amazon Pinpoint SMS dan Voice API menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menggunakan Amazon Pinpoint SMS dan Voice API dengan AWS perangkat pengembangan perangkat lunak (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks pada skenario terkait dan contoh lintas layanan.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- [Tindakan untuk Amazon Pinpoint SMS dan Voice API menggunakan SDK AWS](#)
- [Gunakan SendVoiceMessage dengan AWS SDK atau CLI](#)

Tindakan untuk Amazon Pinpoint SMS dan Voice API menggunakan SDK AWS

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon Pinpoint SMS dan Voice API individual dengan AWS SDK. Kutipan ini memanggil Amazon Pinpoint SMS dan Voice API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi API [Amazon Pinpoint SMS dan Voice API](#).

Contoh

- [Gunakan SendVoiceMessage dengan AWS SDK atau CLI](#)

Gunakan **SendVoiceMessage** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `SendVoiceMessage`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
    software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

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

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

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

    // The content of the message. This example uses SSML to customize and
    control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
    static final String ssmlMessage = "<speak>This is a test message sent
    from "
```

```

        + "<emphasis>Amazon Pinpoint</emphasis> "
        + "using the <break strength='weak'>AWS "
        + "SDK for Java. "
        + "<amazon:effect phonation='soft'>Thank "
        + "you for listening.</amazon:effect></speak>";

public static void main(String[] args) {

    final String usage = ""

        Usage:  <originationNumber> <destinationNumber>

\s

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

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

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

    String originationNumber = args[0];
    String destinationNumber = args[1];
    System.out.println("Sending a voice message");

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

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

    PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()

```

```
        .overrideConfiguration(config2)
        .region(Region.US_EAST_1)
        .build();

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

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

        VoiceMessageContent content =
VoiceMessageContent.builder()
            .ssmlMessage(ssmlMessageType)
            .build();

        SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()

            .destinationPhoneNumber(destinationNumber)

            .originationPhoneNumber(originationNumber)
                .content(content)
                .build();

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

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

- Untuk detail API, lihat [SendVoiceMessage](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v2)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
"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"]);
  }
});
```

- Untuk detail API, lihat [SendVoiceMessage](#) di Referensi AWS SDK for JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
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,
    ssmml_message,
):
    """
```

Sends a voice message using speech synthesis provided by Amazon Polly.

:param sms_voice_client: A Boto3 PinpointSMSVoice client.

:param origination_number: The phone number that the message is sent from.
The phone number must be associated with your

Amazon

Pinpoint account and be in E.164 format.

:param caller_id: The phone number that you want to appear on the recipient's device. The phone number must be associated with your

Amazon

Pinpoint account and be in E.164 format.

:param destination_number: The recipient's phone number. Specify the phone number in E.164 format.

:param language_code: The language to use when sending the message.

:param voice_id: The Amazon Polly voice that you want to use to send the message.

:param ssml_message: The content of the message. This example uses SSML to control certain aspects of the message, such as the volume and

the

speech rate. The message must not contain line breaks.

:return: The ID of the message.

"""

try:

```

    response = sms_voice_client.send_voice_message(
        DestinationPhoneNumber=destination_number,
        OriginationPhoneNumber=origination_number,
        CallerId=caller_id,
        Content={
            "SSMLMessage": {
                "LanguageCode": language_code,
                "VoiceId": voice_id,
                "Text": ssml_message,
            }
        },
    )

```

except ClientError:

```

    logger.exception(
        "Couldn't send message from %s to %s.",
        origination_number,
        destination_number,
    )
    raise

```

else:

```
        return response["MessageId"]

def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssmml_message = (
        "<speaK>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speaK>"
    )
    print(f"Sending voice message from {origination_number} to
{destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssmml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")

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

- Untuk detail API, lihat [SendVoiceMessage](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Pinpoint dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Keamanan di Amazon Pinpoint

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program AWS Kepatuhan Program AWS Kepatuhan](#) . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon Pinpoint, lihat [AWS Layanan dalam Lingkup berdasarkan AWS Layanan Program Kepatuhan dalam Lingkup oleh Program](#) .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon Pinpoint. Topik berikut menunjukkan cara mengonfigurasi Amazon Pinpoint untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amazon Pinpoint Anda.

Untuk informasi selengkapnya, lihat tentang arsitektur referensi, lihat Panduan Arsitektur [Tangguh Amazon Pinpoint](#).

Topik

- [Perlindungan data di Amazon Pinpoint](#)
- [Manajemen identitas dan akses untuk Amazon Pinpoint](#)
- [Pencatatan dan pemantauan di Amazon Pinpoint](#)
- [Validasi kepatuhan untuk Amazon Pinpoint](#)
- [Ketahanan di Amazon Pinpoint](#)
- [Keamanan infrastruktur di Amazon Pinpoint](#)

- [Analisis konfigurasi dan kerentanan di Amazon Pinpoint](#)
- [Praktik terbaik keamanan untuk Amazon Pinpoint](#)

Perlindungan data di Amazon Pinpoint

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di Amazon Pinpoint. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon Pinpoint atau lainnya Layanan AWS

menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Bergantung pada cara Anda mengonfigurasi dan menggunakan layanan, Amazon Pinpoint dapat menyimpan jenis data pribadi berikut untuk Anda atau tentang pelanggan Anda:

Data konfigurasi

Ini termasuk data konfigurasi proyek seperti kredensial dan pengaturan yang menentukan bagaimana dan kapan Amazon Pinpoint mengirim pesan melalui saluran yang didukung, dan segmen pengguna yang mengirim pesan. Untuk mengirim pesan, data ini dapat menyertakan alamat IP khusus untuk pesan email, kode pendek dan ID pengirim untuk pesan teks SMS, serta kredensial untuk berkomunikasi dengan layanan pemberitahuan push seperti layanan Pemberitahuan Push Apple (APN) dan Firebase Cloud Messaging (FCM).

Data pengguna dan titik akhir

Ini termasuk atribut standar dan kustom yang Anda gunakan untuk menyimpan dan mengelola data tentang pengguna dan titik akhir untuk proyek Amazon Pinpoint. Atribut dapat menyimpan informasi tentang pengguna tertentu (seperti nama pengguna) atau titik akhir tertentu untuk pengguna (seperti alamat email pengguna, nomor ponsel, atau token perangkat seluler). Data ini juga dapat menyertakan ID pengguna eksternal yang menghubungkan pengguna untuk proyek Amazon Pinpoint dengan pengguna di sistem eksternal, seperti sistem manajemen hubungan pelanggan. Untuk informasi selengkapnya tentang apa yang dapat disertakan data ini, lihat skema [Pengguna](#) dan [Titik Akhir](#) di Referensi API Amazon Pinpoint.

Data analitik

Ini termasuk data untuk metrik, juga disebut sebagai indikator kinerja utama (KPI), yang memberikan wawasan tentang kinerja proyek Amazon Pinpoint untuk area seperti keterlibatan pengguna dan aktivitas pembelian. Ini juga mencakup data untuk metrik yang memberikan wawasan tentang demografi pengguna untuk suatu proyek. Data dapat berasal dari atribut standar dan kustom untuk pengguna dan titik akhir, seperti kota tempat pengguna tinggal. Hal ini juga dapat berasal dari peristiwa, seperti buka dan klik acara untuk pesan email yang Anda kirim untuk proyek.

Data yang diimpor

Ini termasuk data pengguna, segmentasi, dan analitik apa pun yang Anda tambahkan atau impor dari sumber eksternal dan gunakan di Amazon Pinpoint. Contohnya adalah file JSON yang Anda impor ke Amazon Pinpoint (melalui konsol secara langsung atau dari bucket Amazon S3) untuk membangun segmen statis. Contoh lainnya adalah data titik akhir yang Anda tambahkan secara terprogram untuk membangun segmen dinamis, alamat titik akhir yang Anda kirim pesan langsung, dan peristiwa yang Anda konfigurasi aplikasi untuk dilaporkan ke Amazon Pinpoint.

Topik

- [Enkripsi data](#)
- [Privasi lalu lintas antar jaringan](#)
- [Membuat titik akhir VPC antarmuka untuk Amazon Pinpoint](#)

Enkripsi data

Data Amazon Pinpoint dienkripsi saat transit dan saat istirahat. Saat Anda mengirimkan data ke Amazon Pinpoint, itu mengenkripsi data saat menerima dan menyimpannya. Saat Anda mengambil data dari Amazon Pinpoint, data akan ditransmisikan kepada Anda dengan menggunakan protokol keamanan saat ini.

Enkripsi diam

Amazon Pinpoint mengenkripsi semua data yang disimpannya untuk Anda. Ini termasuk data konfigurasi, data pengguna dan titik akhir, data analitik, dan data apa pun yang Anda tambahkan atau impor ke Amazon Pinpoint. Untuk mengenkripsi data Anda, Amazon Pinpoint menggunakan kunci AWS Key Management Service internal AWS KMS() yang dimiliki dan dikelola oleh layanan atas nama Anda. Kami memutar tombol-tombol ini secara teratur. Untuk selengkapnya AWS KMS, lihat [Panduan AWS Key Management Service Pengembang](#).

Enkripsi bergerak

Amazon Pinpoint menggunakan HTTPS dan Transport Layer Security (TLS) 1.2 atau yang lebih baru untuk berkomunikasi dengan klien dan aplikasi Anda. Untuk berkomunikasi dengan AWS layanan lain, Amazon Pinpoint menggunakan HTTPS dan TLS 1.2. Selain itu, saat Anda membuat dan mengelola sumber daya Amazon Pinpoint menggunakan konsol, AWS SDK, atau AWS Command Line Interface, semua komunikasi diamankan menggunakan HTTPS dan TLS 1.2.

Manajemen kunci

Untuk mengenkripsi data Amazon Pinpoint Anda, Amazon Pinpoint menggunakan kunci AWS KMS internal yang dimiliki dan dikelola layanan atas nama Anda. Kami memutar tombol-tombol ini secara teratur. Anda tidak dapat menyediakan dan menggunakan kunci Anda sendiri AWS KMS atau kunci lainnya untuk mengenkripsi data yang Anda simpan di Amazon Pinpoint.

Privasi lalu lintas antar jaringan

Privasi lalu lintas internetwork mengacu pada pengamanan koneksi dan lalu lintas antara Amazon Pinpoint dan klien serta aplikasi lokal Anda, serta antara Amazon Pinpoint dan sumber daya lain di Wilayah yang sama. AWS AWS Fitur dan praktik berikut dapat membantu Anda memastikan privasi lalu lintas internetwork untuk Amazon Pinpoint.

Lalu lintas antara Amazon Pinpoint dan klien dan aplikasi lokal

Untuk membuat sambungan pribadi antara Amazon Pinpoint dan klien serta aplikasi di jaringan lokal, Anda dapat menggunakannya. AWS Direct Connect Ini memungkinkan Anda untuk menghubungkan jaringan Anda ke suatu AWS Direct Connect lokasi dengan menggunakan kabel Ethernet serat optik standar. Salah satu ujung kabel terhubung ke router Anda. Ujung lainnya terhubung ke AWS Direct Connect router. Untuk informasi selengkapnya, lihat [Apa itu AWS Direct Connect?](#) dalam AWS Direct Connect Panduan Pengguna.

Untuk membantu mengamankan akses ke Amazon Pinpoint melalui API yang dipublikasikan, sebaiknya Anda mematuhi persyaratan Amazon Pinpoint untuk panggilan API. Amazon Pinpoint mengharuskan klien untuk menggunakan Transport Layer Security (TLS) 1.2 atau yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS), seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Sebagian besar sistem modern seperti Java 7 dan versi yang lebih baru support dengan mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal AWS Identity and Access Management (IAM) untuk AWS akun Anda. Atau, Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Lalu lintas antara Amazon Pinpoint dan sumber daya lainnya AWS

Untuk mengamankan komunikasi antara Amazon Pinpoint dan AWS sumber daya lain di AWS Wilayah yang sama, Amazon Pinpoint menggunakan HTTPS dan TLS 1.2 secara default.

Membuat titik akhir VPC antarmuka untuk Amazon Pinpoint

Anda dapat membuat koneksi pribadi antara virtual private cloud (VPC) dan endpoint di Amazon Pinpoint dengan membuat antarmuka VPC endpoint.

Endpoint antarmuka didukung oleh [AWS PrivateLink](#), teknologi yang memungkinkan Anda mengakses Amazon Pinpoint API secara pribadi tanpa gateway internet, perangkat NAT, koneksi VPN, atau. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan Amazon Pinpoint API yang terintegrasi dengannya. AWS PrivateLink

Untuk informasi lebih lanjut, lihat [AWS PrivateLink Panduan](#).

Membuat antarmuka VPC endpoint

Anda dapat membuat titik akhir antarmuka menggunakan konsol VPC Amazon atau () AWS Command Line Interface .AWS CLI Untuk informasi selengkapnya, lihat [Membuat titik akhir antarmuka](#) di AWS PrivateLink Panduan.

Amazon Pinpoint mendukung nama layanan berikut:

- `com.amazonaws.region.pinpoint`
- `com.amazonaws.region.pinpoint-sms-voice-v2`

Jika Anda mengaktifkan DNS pribadi untuk titik akhir antarmuka, Anda dapat membuat permintaan API ke Amazon Pinpoint menggunakan nama DNS default untuk, misalnya Wilayah AWS,. `com.amazonaws.us-east-1.pinpoint` Untuk informasi selengkapnya, lihat [nama host DNS](#) di Panduan.AWS PrivateLink

Untuk daftar semua Wilayah dan titik akhir di mana Amazon Pinpoint saat ini tersedia, [AWS lihat titik akhir layanan](#) di. Referensi Umum Amazon Web

Membuat kebijakan titik akhir VPC

Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC yang mengontrol akses. Kebijakan titik akhir menentukan informasi berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.

- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke layanan menggunakan kebijakan titik akhir](#) di Panduan AWS PrivateLink .

Contoh: kebijakan titik akhir VPC

Kebijakan titik akhir VPC berikut memberikan akses ke tindakan Amazon Pinpoint yang terdaftar untuk semua prinsipal di semua sumber daya.

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

Manajemen identitas dan akses untuk Amazon Pinpoint

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon Pinpoint. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana Amazon Pinpoint bekerja dengan IAM](#)
- [Tindakan Amazon Pinpoint untuk kebijakan IAM](#)

- [Amazon Pinpoint contoh kebijakan berbasis identitas](#)
- [Peran IAM untuk tugas Amazon Pinpoint umum](#)
- [Memecahkan masalah Amazon Pinpoint identitas dan manajemen akses](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amazon Pinpoint.

Pengguna layanan - Jika Anda menggunakan layanan Amazon Pinpoint untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensial dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Amazon Pinpoint untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon Pinpoint, lihat. [Memecahkan masalah Amazon Pinpoint identitas dan manajemen akses](#)

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya Amazon Pinpoint di perusahaan Anda, Anda mungkin memiliki akses penuh ke Amazon Pinpoint. Tugas Anda adalah menentukan fitur dan sumber daya Amazon Pinpoint mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari selengkapnya tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Amazon Pinpoint, lihat. [Bagaimana Amazon Pinpoint bekerja dengan IAM](#)

Administrator IAM - Jika Anda administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Amazon Pinpoint. Untuk melihat contoh kebijakan berbasis identitas Amazon Pinpoint yang dapat Anda gunakan di IAM, lihat. [Amazon Pinpoint contoh kebijakan berbasis identitas](#)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensial identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda adalah contoh

identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat [Autentikasi multi-faktor](#) dalam Panduan Pengguna AWS IAM Identity Center dan [Menggunakan autentikasi multi-faktor \(MFA\) dalam AWS](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci

akses. Untuk informasi selengkapnya, lihat [Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) dalam Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin ke grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat [Menggunakan peran IAM](#) dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat [Set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat

melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.

- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda memanggil suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses maju](#).
- Peran layanan – Peran layanan adalah [peran IAM](#) yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan dalam instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam Akun AWS. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang

dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat [Memilih antara kebijakan yang dikelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Amazon Pinpoint mendukung penggunaan kebijakan berbasis identitas untuk mengontrol akses ke sumber daya Amazon Pinpoint.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Amazon Pinpoint mendukung penggunaan kebijakan berbasis sumber daya untuk mengontrol akses ke sumber daya Amazon Pinpoint.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) dalam Panduan Developer Amazon Simple Storage Service.

Amazon Pinpoint tidak mendukung penggunaan ACL untuk mengontrol akses ke sumber daya Amazon Pinpoint.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- **Batasan izin** – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- **Kebijakan kontrol layanan (SCP)** — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .
- **Kebijakan sesi** – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Amazon Pinpoint mendukung penggunaan jenis kebijakan ini untuk mengontrol akses ke sumber daya Amazon Pinpoint.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

Bagaimana Amazon Pinpoint bekerja dengan IAM

Untuk menggunakan Amazon Pinpoint, pengguna di AWS akun Anda memerlukan izin yang memungkinkan mereka melihat data analitik, membuat proyek, menentukan segmen pengguna, menyebarkan kampanye, dan banyak lagi. Jika Anda mengintegrasikan aplikasi seluler atau web dengan Amazon Pinpoint, pengguna aplikasi Anda juga memerlukan akses ke Amazon Pinpoint.

Akses ini memungkinkan aplikasi Anda mendaftarkan titik akhir dan melaporkan data penggunaan ke Amazon Pinpoint. Untuk memberikan akses ke fitur Amazon Pinpoint, buat kebijakan AWS Identity and Access Management (IAM) yang mengizinkan tindakan Amazon Pinpoint untuk identitas IAM atau sumber daya Amazon Pinpoint.

IAM adalah layanan yang membantu administrator mengontrol akses ke sumber daya dengan aman. AWS Kebijakan IAM mencakup pernyataan yang mengizinkan atau menolak tindakan tertentu oleh pengguna tertentu atau untuk sumber daya tertentu. Amazon Pinpoint menyediakan [serangkaian tindakan](#) yang dapat Anda gunakan dalam kebijakan IAM untuk menentukan izin terperinci bagi pengguna dan sumber daya Amazon Pinpoint. Ini berarti Anda dapat memberikan tingkat akses yang sesuai ke Amazon Pinpoint tanpa membuat kebijakan yang terlalu permisif yang mungkin mengekspos data penting atau membahayakan sumber daya Anda. Misalnya, Anda dapat memberikan akses tidak terbatas ke administrator Amazon Pinpoint, dan memberikan akses hanya-baca kepada individu yang hanya memerlukan akses ke proyek tertentu.

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon Pinpoint, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan dengan Amazon Pinpoint. Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Amazon Pinpoint dan layanan AWS lainnya dengan IAM, [AWS lihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Topik

- [Amazon Pinpoint kebijakan berbasis identitas](#)
- [Amazon Pinpoint kebijakan izin berbasis sumber daya](#)
- [Otorisasi berdasarkan tag Amazon Pinpoint](#)
- [Amazon Pinpoint peran IAM](#)

Amazon Pinpoint kebijakan berbasis identitas

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Amazon Pinpoint mendukung tindakan, sumber daya, dan kunci kondisi tertentu. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat [referensi elemen kebijakan IAM JSON](#) di Panduan Pengguna IAM.

Tindakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Ini berarti bahwa tindakan kebijakan mengontrol apa yang dapat dilakukan pengguna di konsol Amazon Pinpoint. Mereka juga mengontrol apa yang dapat dilakukan pengguna secara terprogram dengan menggunakan AWS SDK, AWS Command Line Interface (AWS CLI), atau Amazon Pinpoint API secara langsung.

Tindakan kebijakan di Amazon Pinpoint menggunakan awalan berikut:

- **mobiletargeting**— Untuk tindakan yang berasal dari Amazon Pinpoint API, yang merupakan API utama untuk Amazon Pinpoint.
- **sms-voice**— Untuk tindakan yang berasal dari Amazon Pinpoint SMS dan Voice API, yang merupakan API tambahan yang menyediakan opsi lanjutan untuk menggunakan dan mengelola saluran SMS dan suara di Amazon Pinpoint.

Misalnya, untuk memberikan izin kepada seseorang untuk melihat informasi tentang semua segmen untuk proyek, yang merupakan tindakan yang sesuai dengan `GetSegments` operasi di Amazon Pinpoint API, sertakan `mobiletargeting:GetSegments` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction`. Amazon Pinpoint mendefinisikan serangkaian tindakannya sendiri yang menggambarkan tugas yang dapat dilakukan pengguna dengannya.

Untuk menentukan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma:

```
"Action": [  
  "mobiletargeting:action1",  
  "mobiletargeting:action2"
```

Anda juga dapat menentukan beberapa tindakan dengan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata Get, sertakan tindakan berikut:

```
"Action": "mobiletargeting:Get*"
```

Namun, sebagai praktik terbaik, Anda harus membuat kebijakan yang mengikuti prinsip hak istimewa paling sedikit. Dengan kata lain, Anda harus membuat kebijakan yang hanya menyertakan izin yang diperlukan untuk melakukan tindakan tertentu.

Untuk daftar tindakan Amazon Pinpoint yang dapat Anda gunakan dalam kebijakan IAM, lihat.

[Tindakan Amazon Pinpoint untuk kebijakan IAM](#)

Sumber daya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Misalnya, `mobiletargeting:GetSegments` tindakan mengambil informasi tentang semua segmen yang terkait dengan proyek Amazon Pinpoint tertentu. Anda mengidentifikasi proyek dengan ARN dalam format berikut:

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

Untuk informasi selengkapnya tentang format ARN, lihat [Amazon Resource Name \(ARN\)](#) di Referensi Umum AWS.

Dalam kebijakan IAM, Anda dapat menentukan ARN untuk jenis sumber daya Amazon Pinpoint berikut:

- Kampanye
- Perjalanan
- Template pesan (disebut sebagai template dalam beberapa konteks)
- Proyek (disebut sebagai aplikasi atau aplikasi dalam beberapa konteks)
- Model pemberi rekomendasi (disebut sebagai pemberi rekomendasi dalam beberapa konteks)
- Segmen

Misalnya, untuk membuat pernyataan kebijakan untuk proyek yang memiliki ID proyek810c7aab86d42fb2b56c8c966example, gunakan ARN berikut:

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

Untuk menentukan semua proyek milik akun tertentu, gunakan wildcard (*):

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

Beberapa tindakan Amazon Pinpoint, seperti tindakan tertentu untuk membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (*):

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

Dalam kebijakan IAM, Anda juga dapat menentukan ARN untuk jenis sumber daya Amazon Pinpoint SMS dan Voice berikut:

- Set Konfigurasi
- Daftar Keluar
- Nomor Telepon
- Kolam
- Id Pengirim

Misalnya, untuk membuat pernyataan kebijakan untuk nomor telepon yang memiliki ID nomor telepon, `phone-12345678901234567890123456789012` gunakan ARN berikut:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/  
phone-12345678901234567890123456789012"
```

Untuk menentukan semua nomor telepon milik akun tertentu, gunakan wildcard (*) sebagai pengganti ID nomor telepon:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

Beberapa tindakan Amazon Pinpoint SMS dan Voice tidak dilakukan pada sumber daya tertentu, seperti untuk mengelola pengaturan tingkat akun seperti batas pengeluaran. Dalam kasus tersebut, Anda harus menggunakan wildcard (*):

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

Beberapa tindakan Amazon Pinpoint API melibatkan banyak sumber daya. Misalnya, `TagResource` tindakan dapat menambahkan tag ke beberapa proyek. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN dengan koma:

```
"Resource": [  
    "resource1",  
    "resource2"
```

Untuk melihat daftar jenis sumber daya Amazon Pinpoint dan ARNnya, lihat [Sumber Daya yang Ditentukan oleh Amazon Pinpoint](#) di Panduan Pengguna IAM. Untuk mempelajari tindakan mana yang dapat Anda tentukan dengan ARN dari setiap jenis sumber daya, lihat [Tindakan yang Ditentukan oleh Amazon Pinpoint](#) di Panduan Pengguna IAM.

Kunci syarat

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat

membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen Condition dalam sebuah pernyataan, atau beberapa kunci dalam elemen Condition tunggal, maka AWS akan mengevaluasinya menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM: variabel dan tag](#) dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Amazon Pinpoint mendefinisikan kumpulan kunci kondisinya sendiri dan juga mendukung beberapa kunci kondisi global. Untuk melihat daftar semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM. Untuk melihat daftar kunci kondisi Amazon Pinpoint, lihat Kunci Kondisi [untuk Amazon Pinpoint](#) di Panduan Pengguna IAM. Untuk mempelajari tindakan dan sumber daya yang dapat digunakan untuk menggunakan kunci kondisi, lihat [Tindakan yang Ditentukan oleh Amazon Pinpoint di Panduan Pengguna IAM](#).

Contoh

Untuk melihat contoh kebijakan berbasis identitas Amazon Pinpoint, lihat. [Amazon Pinpoint contoh kebijakan berbasis identitas](#)

Amazon Pinpoint kebijakan izin berbasis sumber daya

Kebijakan izin berbasis sumber daya adalah dokumen kebijakan JSON yang menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya Amazon Pinpoint dan dalam kondisi apa. Amazon Pinpoint mendukung kebijakan izin berbasis sumber daya untuk kampanye, perjalanan, templat pesan (templat), model pemberi rekomendasi (pemberi rekomendasi), proyek (aplikasi), dan segmen.

Contoh

Untuk melihat contoh kebijakan berbasis sumber daya Amazon Pinpoint, lihat. [the section called “Contoh kebijakan berbasis identitas”](#)

Otorisasi berdasarkan tag Amazon Pinpoint

Anda dapat mengaitkan tag dengan jenis sumber daya Amazon Pinpoint tertentu atau meneruskan tag dalam permintaan ke Amazon Pinpoint. Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/${TagKey}`, `aws:RequestTag/${TagKey}`, atau `aws:TagKeys`.

Untuk informasi tentang menandai sumber daya Amazon Pinpoint, termasuk contoh kebijakan IAM, lihat [Memberi tag pada sumber daya Amazon Pinpoint](#)

Amazon Pinpoint peran IAM

[Peran](#) IAM adalah entitas di dalam akun AWS Anda yang memiliki izin tertentu.

Menggunakan kredensi sementara dengan Amazon Pinpoint

Anda dapat menggunakan kredensial sementara untuk masuk ke federasi, mengasumsikan peran IAM, atau mengasumsikan peran lintas akun. Anda memperoleh kredensial keamanan sementara dengan memanggil AWS Security Token Service (AWS STS) operasi API seperti [AssumeRole](#) atau [GetFederationToken](#)

Amazon Pinpoint mendukung penggunaan kredensial sementara.

Peran terkait layanan

[Peran terkait AWS layanan](#) memungkinkan layanan mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran terkait layanan muncul di akun IAM Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Amazon Pinpoint tidak menggunakan peran terkait layanan.

Peran layanan

Fitur ini memungkinkan layanan untuk menerima [peran layanan](#) atas nama Anda. Peran ini mengizinkan layanan untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran layanan muncul di akun IAM Anda dan dimiliki oleh akun tersebut. Ini berarti administrator IAM dapat mengubah izin untuk peran ini. Namun, melakukan hal itu dapat merusak fungsionalitas layanan.

Amazon Pinpoint mendukung penggunaan peran layanan.

Tindakan Amazon Pinpoint untuk kebijakan IAM

Untuk mengelola akses ke sumber daya Amazon Pinpoint di AWS akun Anda, Anda dapat menambahkan tindakan Amazon Pinpoint AWS Identity and Access Management ke kebijakan (IAM). Dengan menggunakan tindakan dalam kebijakan, Anda dapat mengontrol apa yang dapat dilakukan pengguna di konsol Amazon Pinpoint. Anda juga dapat mengontrol apa yang dapat dilakukan pengguna secara terprogram dengan menggunakan AWS SDK, AWS Command Line Interface (AWS CLI), atau Amazon Pinpoint API secara langsung.

Dalam kebijakan, Anda menentukan setiap tindakan dengan namespace Amazon Pinpoint yang sesuai diikuti dengan titik dua dan nama tindakan, seperti. `GetSegments` Sebagian besar tindakan sesuai dengan permintaan ke Amazon Pinpoint API menggunakan URI dan metode HTTP tertentu. Misalnya, jika Anda mengizinkan `mobiletargeting:GetSegments` tindakan dalam kebijakan pengguna, pengguna diizinkan untuk mengambil informasi tentang semua segmen untuk proyek dengan mengirimkan permintaan HTTP GET ke URI. [/apps/projectId/segments](#) Kebijakan ini juga memungkinkan pengguna untuk melihat informasi tersebut di konsol, dan mengambil informasi tersebut dengan menggunakan AWS SDK atau file. AWS CLI

Setiap tindakan dilakukan pada sumber daya Amazon Pinpoint tertentu, yang Anda identifikasi dalam pernyataan kebijakan dengan Nama Sumber Daya Amazon (ARN). Misalnya, `mobiletargeting:GetSegments` tindakan dilakukan pada proyek tertentu, yang Anda identifikasi dengan ARN, `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Topik ini mengidentifikasi tindakan Amazon Pinpoint yang dapat Anda tambahkan ke kebijakan IAM untuk akun Anda. AWS Untuk melihat contoh yang menunjukkan cara Anda dapat menggunakan tindakan dalam kebijakan untuk mengelola akses ke sumber daya Amazon Pinpoint, lihat. [Amazon Pinpoint contoh kebijakan berbasis identitas](#)

Topik

- [Tindakan API Amazon Pinpoint](#)
- [Amazon Pinpoint SMS dan tindakan API versi suara 1](#)

Tindakan API Amazon Pinpoint

Bagian ini mengidentifikasi tindakan untuk fitur yang tersedia dari Amazon Pinpoint API, yang merupakan API utama untuk Amazon Pinpoint. Untuk mempelajari lebih lanjut tentang API ini, lihat Referensi [API Amazon Pinpoint](#).

Kategori:

- [Analitik dan metrik](#)
- [Kampanye](#)
- [Saluran](#)
- [Titik akhir](#)
- [Aliran acara](#)
- [Peristiwa](#)
- [Lowongan kerja ekspor](#)
- [Impor lowongan kerja](#)
- [Perjalanan](#)
- [Template pesan](#)
- [Pesan](#)
- [Kata sandi satu kali](#)
- [Validasi nomor telepon](#)
- [Proyek](#)
- [Model rekomendasi](#)
- [Segmen](#)
- [Tanda](#)
- [Pengguna](#)

Analitik dan metrik

Izin berikut terkait dengan melihat data analitik di konsol Amazon Pinpoint. Mereka juga terkait dengan pengambilan (kueri) data agregat untuk metrik standar, juga disebut sebagai indikator kinerja utama (KPI), yang berlaku untuk proyek, kampanye, dan perjalanan.

mobiletargeting:GetReports

Lihat data analitik di konsol Amazon Pinpoint. Izin ini juga diperlukan untuk membuat segmen yang berisi atribut khusus menggunakan konsol Amazon Pinpoint. Ini juga diperlukan untuk mendapatkan perkiraan ukuran segmen di konsol Amazon Pinpoint.

- URI - Tidak berlaku
- Metode - Tidak berlaku

- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:*`

mobiletargeting:GetApplicationDateRangeKpi

Ambil (kueri) data agregat untuk metrik aplikasi standar. Ini adalah metrik yang berlaku untuk semua kampanye atau pesan transaksional yang terkait dengan proyek.

- URI — [/apps/projectId/kpis/daterange/kpi-name](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/kpis/daterange/kpi-name`

mobiletargeting:GetCampaignDateRangeKpi

Ambil (kueri) data agregat untuk metrik kampanye standar. Ini adalah metrik yang berlaku untuk kampanye individu.

- URI — [/apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name`

mobiletargeting:GetJourneyDateRangeKpi

Ambil (kueri) data agregat untuk metrik keterlibatan perjalanan standar. Ini adalah metrik keterlibatan yang berlaku untuk perjalanan individu—misalnya, jumlah pesan yang dibuka oleh peserta untuk semua aktivitas dalam perjalanan.

- URI — [/apps/projectId/journeys/journeyId/kpis/daterange/kpi-name](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/kpis/daterange/kpi-name`

mobiletargeting:GetJourneyExecutionMetrics

Ambil (kueri) data agregat untuk metrik eksekusi standar yang berlaku untuk perjalanan individu—misalnya, jumlah peserta yang secara aktif melanjutkan semua aktivitas dalam suatu perjalanan.

- URI — [/apps/projectId/journeys/journeyId/execution-metrics](#)

- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/execution-metrics`

mobiletargeting:GetJourneyExecutionActivityMetrics

Ambil (kueri) data agregat untuk metrik eksekusi standar yang berlaku untuk aktivitas individu dalam suatu perjalanan—misalnya, jumlah peserta yang memulai atau menyelesaikan suatu aktivitas.

- URI — [/apps/projectId/journeys/journeyId/activities/journey-activity-id/execution-metrics](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/activities/journey-activity-id/execution-metrics`

Kampanye

Izin berikut terkait dengan mengelola kampanye di akun Amazon Pinpoint Anda.

mobiletargeting:CreateCampaign

Buat kampanye untuk sebuah proyek.

- URI — [/apps/projectId/campaigns](#)
- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns`

mobiletargeting>DeleteCampaign

Hapus kampanye tertentu.

- URI — [/apps/projectId/campaigns/campaignId](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

mobiletargeting:GetCampaign

Ambil informasi tentang kampanye tertentu.

- URI — [/apps/*projectId*/campaigns/*campaignId*](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
campaigns/*campaignId*

mobiletargeting:GetCampaignActivities

Mengambil informasi tentang kegiatan yang dilakukan oleh kampanye.

- URI — [/apps/*projectId*/campaigns/*campaignId*/activities](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
campaigns/*campaignId*

mobiletargeting:GetCampaigns

Mengambil informasi tentang semua kampanye untuk sebuah proyek.

- URI — [/apps/*projectId*/campaigns](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetCampaignVersion

Mengambil informasi tentang versi kampanye tertentu.

- URI — [/apps/*projectId*/campaigns/*campaignId*/versions/*versionId*](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
campaigns/*campaignId*

mobiletargeting:GetCampaignVersions

Mengambil informasi tentang versi kampanye saat ini dan sebelumnya.

- URI — [/apps/*projectId*/campaigns/*campaignId*/versions](#)

- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

mobiletargeting:UpdateCampaign

Perbarui kampanye tertentu.

- URI — [/apps/projectId/campaigns/campaignId](#)
- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

Saluran

Izin berikut terkait dengan pengelolaan saluran di akun Amazon Pinpoint Anda. Di Amazon Pinpoint, saluran mengacu pada metode yang Anda gunakan untuk menghubungi pelanggan Anda, seperti mengirim email, pesan SMS, atau pemberitahuan push.

mobiletargeting>DeleteAdmChannel

Nonaktifkan saluran Amazon Device Messaging (ADM) untuk sebuah proyek.

- URI — [/apps/projectId/channels/adm](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

mobiletargeting:GetAdmChannel

Ambil informasi tentang saluran ADM untuk sebuah proyek.

- URI — [/apps/projectId/channels/adm](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

mobiletargeting:UpdateAdmChannel

Aktifkan atau perbarui saluran ADM untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/adm](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:DeleteApnsChannel

Nonaktifkan saluran layanan Pemberitahuan Push Apple (APN) untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:GetApnsChannel

Mengambil informasi tentang saluran APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:UpdateApnsChannel

Aktifkan atau perbarui saluran APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting>DeleteApnsSandboxChannel

Nonaktifkan saluran kotak pasir APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_sandbox](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/
apns_sandbox

mobiletargeting:GetApnsSandboxChannel

Ambil informasi tentang saluran kotak pasir APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_sandbox](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

mobiletargeting:UpdateApnsSandboxChannel

Mengaktifkan atau memperbarui saluran kotak pasir APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_sandbox](#)
- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

mobiletargeting>DeleteApnsVoipChannel

Nonaktifkan saluran VoIP APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_voip](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

mobiletargeting:GetApnsVoipChannel

Mengambil informasi tentang saluran VoIP APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_voip](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

mobiletargeting:UpdateApnsVoipChannel

Aktifkan atau perbarui saluran VoIP APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_voip](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/
apns_voip

mobiletargeting:DeleteApnsVoipSandboxChannel

Nonaktifkan saluran kotak pasir VoIP APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_voip_sandbox](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/
apns_voip_sandbox

mobiletargeting:GetApnsVoipSandboxChannel

Mengambil informasi tentang saluran kotak pasir VoIP APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_voip_sandbox](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/
apns_voip_sandbox

mobiletargeting:UpdateApnsVoipSandboxChannel

Aktifkan atau perbarui saluran kotak pasir VoIP APN untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/apns_voip_sandbox](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/
apns_voip_sandbox

mobiletargeting>DeleteBaiduChannel

Nonaktifkan saluran Baidu Cloud Push untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/baidu](#)
- Metode - HAPUS

- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:GetBaiduChannel

Ambil informasi tentang saluran Baidu Cloud Push untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/baidu](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:UpdateBaiduChannel

Aktifkan atau perbarui saluran Baidu Cloud Push untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/baidu](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting>DeleteEmailChannel

Nonaktifkan saluran email untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/email](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:GetEmailChannel

Mengambil informasi tentang saluran email untuk sebuah proyek.

- URI — [/apps/*projectId*/channels/email](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:UpdateEmailChannel

Mengaktifkan atau memperbarui saluran email untuk proyek.

- URI — [/apps/*projectId*/channels/email](#)

- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/email`

mobiletargeting:DeleteGcmChannel

Nonaktifkan channel Firebase Cloud Messaging (FCM) untuk sebuah project. Saluran ini memungkinkan Amazon Pinpoint untuk mengirim pemberitahuan push ke aplikasi Android melalui layanan FCM, yang menggantikan layanan Google Cloud Messaging (GCM).

- URI — [/apps/projectId/channels/gcm](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

mobiletargeting:GetGcmChannel

Ambil informasi tentang saluran FCM untuk sebuah proyek. Saluran ini memungkinkan Amazon Pinpoint untuk mengirim pemberitahuan push ke aplikasi Android melalui layanan FCM, yang menggantikan layanan Google Cloud Messaging (GCM).

- URI — [/apps/projectId/channels/gcm](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

mobiletargeting:UpdateGcmChannel

Aktifkan atau perbarui saluran FCM untuk sebuah proyek. Saluran ini memungkinkan Amazon Pinpoint untuk mengirim pemberitahuan push ke aplikasi Android melalui layanan FCM, yang menggantikan layanan Google Cloud Messaging (GCM).

- URI — [/apps/projectId/channels/gcm](#)
- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

mobiletargeting>DeleteSmsChannel

Nonaktifkan saluran SMS untuk suatu proyek.

- URI — [/apps/projectId/channels/sms](#)
- Metode - HAPUS

- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

mobiletargeting:GetSmsChannel

Ambil informasi tentang saluran SMS untuk suatu proyek.

- URI — [/apps/projectId/channels/sms](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

mobiletargeting:UpdateSmsChannel

Aktifkan atau perbarui saluran SMS untuk suatu proyek.

- URI — [/apps/projectId/channels/sms](#)
- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

mobiletargeting:GetChannels

Mengambil informasi tentang riwayat dan status setiap saluran untuk aplikasi.

- URI — [/apps/application-id/channels](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/channels`

mobiletargeting>DeleteVoiceChannel

Menonaktifkan saluran suara untuk aplikasi dan menghapus pengaturan yang ada untuk saluran tersebut.

- URI — [/apps/application-id/channels/voice](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

mobiletargeting:GetVoiceChannel

Mengambil informasi tentang status dan pengaturan saluran suara untuk suatu aplikasi.

- URI — [/apps/*application-id*/channels/voice](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:UpdateVoiceChannel

Mengaktifkan saluran suara untuk aplikasi atau memperbarui status dan pengaturan saluran suara untuk suatu aplikasi.

- URI — [/apps/*application-id*/channels/voice](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

Titik akhir

Izin berikut terkait dengan pengelolaan titik akhir di akun Amazon Pinpoint Anda. Di Amazon Pinpoint, titik akhir adalah tujuan tunggal untuk pesan Anda. Misalnya, titik akhir dapat berupa alamat email pelanggan, nomor telepon, atau token perangkat seluler.

mobiletargeting>DeleteEndpoint

Hapus titik akhir.

- URI — [/apps/*projectId*/endpoints/*endpointId*](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
endpoints/*endpointId*

mobiletargeting:GetEndpoint

Mengambil informasi tentang titik akhir tertentu.

- URI — [/apps/*projectId*/endpoints/*endpointId*](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
endpoints/*endpointId*

mobiletargeting:RemoveAttributes

Menghapus satu atau lebih atribut, dari jenis atribut yang sama, dari semua titik akhir yang terkait dengan aplikasi.

- URI — [apps/application-id/attributes/attribute-type](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId/
attributes/attribute-type

mobiletargeting:UpdateEndpoint

Buat titik akhir atau perbarui informasi untuk titik akhir.

- URI — [/apps/projectId/endpoints/endpointId](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId/
endpoints/endpointId

mobiletargeting:UpdateEndpointsBatch

Buat atau perbarui titik akhir sebagai operasi batch.

- URI — [/apps/projectId/endpoints](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId

Aliran acara

Izin berikut terkait dengan pengelolaan aliran acara untuk akun Amazon Pinpoint Anda.

mobiletargeting>DeleteEventStream

Hapus aliran acara untuk proyek.

- URI — [/apps/projectId/eventstream/](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream

mobiletargeting:GetEventStream

Mengambil informasi tentang aliran acara untuk sebuah proyek.

- URI — [/apps/*projectId*/eventstream/](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

mobiletargeting:PutEventStream

Buat atau perbarui aliran acara untuk proyek.

- URI — [/apps/*projectId*/eventstream/](#)
- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

Peristiwa

Izin berikut terkait dengan mengelola pekerjaan acara di akun Amazon Pinpoint Anda. Di Amazon Pinpoint, Anda membuat pekerjaan impor untuk membuat segmen berdasarkan definisi titik akhir yang disimpan dalam bucket Amazon S3.

mobiletargeting:PutEvents

Membuat acara baru untuk merekam titik akhir, atau membuat atau memperbarui data titik akhir yang terkait dengan peristiwa yang ada.

- URI — [/apps/*application-id*/events](#)
- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/events`

Lowongan kerja ekspor

Izin berikut terkait dengan mengelola pekerjaan ekspor di akun Amazon Pinpoint Anda. Di Amazon Pinpoint, Anda membuat lowongan ekspor untuk mengirim informasi tentang titik akhir ke bucket Amazon S3 untuk penyimpanan atau analisis.

mobiletargeting:CreateExportJob

Buat pekerjaan ekspor untuk mengekspor definisi titik akhir ke Amazon S3.

- URI — [/apps/*projectId*/jobs/export](#)
- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

mobiletargeting:GetExportJob

Mengambil informasi tentang pekerjaan ekspor tertentu untuk sebuah proyek.

- URI — [/apps/*projectId*/jobs/export/*jobId*](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export/jobId`

mobiletargeting:GetExportJobs

Ambil daftar semua pekerjaan ekspor untuk sebuah proyek.

- URI — [/apps/*projectId*/jobs/export](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

Impor lowongan kerja

Izin berikut terkait dengan mengelola pekerjaan impor di akun Amazon Pinpoint Anda. Di Amazon Pinpoint, Anda membuat pekerjaan impor untuk membuat segmen berdasarkan definisi titik akhir yang disimpan dalam bucket Amazon S3.

mobiletargeting:CreateImportJob

Impor definisi titik akhir dari Amazon S3 untuk membuat segmen.

- URI — [/apps/*projectId*/jobs/import](#)
- Metode – POST

- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId`

mobiletargeting:GetImportJob

Mengambil informasi tentang pekerjaan impor tertentu untuk sebuah proyek.

- URI — [/apps/projectId/jobs/import/jobId](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/import/jobId`

mobiletargeting:GetImportJobs

Mengambil informasi tentang semua pekerjaan impor untuk sebuah proyek.

- URI — [/apps/projectId/jobs/import](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId`

Perjalanan

Izin berikut terkait dengan mengelola perjalanan di akun Amazon Pinpoint Anda.

mobiletargeting:CreateJourney

Buat perjalanan untuk sebuah proyek.

- URI — [/apps/projectId/journeys](#)
- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

mobiletargeting:GetJourney

Ambil informasi tentang perjalanan tertentu.

- URI — [/apps/projectId/journeys/journeyId](#)
- Metode – GET

- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

mobiletargeting:ListJourneys

Ambil informasi tentang semua perjalanan untuk sebuah proyek.

- URI — [/apps/projectId/journeys](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

mobiletargeting:UpdateJourney

Perbarui konfigurasi dan pengaturan lain untuk perjalanan tertentu.

- URI — [/apps/projectId/journeys/journeyId](#)
- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

mobiletargeting:UpdateJourneyState

Batalkan perjalanan aktif.

- URI — [/apps/projectId/journeys/journeyId/state](#)
- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/state`

mobiletargeting>DeleteJourney

Hapus perjalanan tertentu.

- URI — [/apps/projectId/journeys/journeyId](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

Template pesan

Izin berikut terkait dengan membuat dan mengelola templat pesan untuk akun Amazon Pinpoint Anda. Template pesan adalah sekumpulan konten dan pengaturan yang dapat Anda tentukan, simpan, dan gunakan kembali dalam pesan yang Anda kirim untuk salah satu proyek Amazon Pinpoint Anda.

mobiletargeting:ListTemplates

Ambil informasi tentang semua templat pesan yang terkait dengan akun Amazon Pinpoint Anda.

- URI — [/templates](#)
- Metode – GET
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:templates`

mobiletargeting:ListTemplateVersions

Ambil informasi tentang semua versi template pesan tertentu.

- URI — [/templates/template-name/template-type/versions](#)
- Metode – GET
- Sumber ARN - Tidak berlaku

mobiletargeting:UpdateTemplateActiveVersion

Tentukan versi tertentu dari template pesan sebagai versi aktif dari template.

- URI — [/templates/template-name/template-type/active-version](#)
- Metode – GET
- Sumber ARN - Tidak berlaku

mobiletargeting:GetEmailTemplate

Mengambil informasi tentang template pesan untuk pesan yang dikirim melalui saluran email.

- URI — [/templates/template-name/email](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

mobiletargeting:CreateEmailTemplate

Buat template pesan untuk pesan yang dikirim melalui saluran email.

- URI — [/templates/*template-name*/email](#)
- Metode – POST
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:UpdateEmailTemplate

Perbarui template pesan yang ada untuk pesan yang dikirim melalui saluran email.

- URI — [/templates/*template-name*/email](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting>DeleteEmailTemplate

Hapus templat pesan untuk pesan yang dikirim melalui saluran email.

- URI — [/templates/*template-name*/email](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:GetPushTemplate

Mengambil informasi tentang template pesan untuk pesan yang dikirim melalui saluran pemberitahuan push.

- URI — [/templates/*template-name*/push](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:CreatePushTemplate

Buat template pesan untuk pesan yang dikirim melalui saluran notifikasi push.

- URI — [/templates/template-name/push](#)
- Metode – POST
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH

mobiletargeting:UpdatePushTemplate

Perbarui template pesan yang ada untuk pesan yang dikirim melalui saluran pemberitahuan push.

- URI — [/templates/template-name/push](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH

mobiletargeting>DeletePushTemplate

Hapus templat pesan untuk pesan yang dikirim melalui saluran pemberitahuan push.

- URI — [/templates/template-name/push](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH

mobiletargeting:GetSmsTemplate

Ambil informasi tentang template pesan untuk pesan yang dikirim melalui saluran SMS.

- URI — [/templates/template-name/sms](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS

mobiletargeting:CreateSmsTemplate

Buat templat pesan untuk pesan yang dikirim melalui saluran SMS.

- URI — [/templates/*template-name*/sms](#)
- Metode – POST
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:UpdateSmsTemplate

Perbarui templat pesan yang ada untuk pesan yang dikirim melalui saluran SMS.

- URI — [/templates/*template-name*/sms](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting>DeleteSmsTemplate

Hapus templat pesan untuk pesan yang dikirim melalui saluran SMS.

- URI — [/templates/*template-name*/sms](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:GetVoiceTemplate

Mengambil informasi tentang template pesan untuk pesan yang dikirim melalui saluran suara.

- URI — [/templates/*template-name*/voice](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting>CreateVoiceTemplate

Buat templat pesan untuk pesan yang dikirim melalui saluran suara.

- URI — [/templates/*template-name*/voice](#)
- Metode – POST

- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

mobiletargeting:UpdateVoiceTemplate

Perbarui templat pesan yang ada untuk pesan yang dikirim melalui saluran suara.

- URI — [/templates/template-name/voice](#)
- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

mobiletargeting>DeleteVoiceTemplate

Hapus templat pesan untuk pesan yang dikirim melalui saluran suara.

- URI — [/templates/template-name/voice](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

Pesan

Izin berikut terkait dengan pengiriman pesan dan pemberitahuan push dari akun Amazon Pinpoint Anda. Anda dapat menggunakan `SendMessage` dan `SendUsersMessages` operasi untuk mengirim pesan ke titik akhir tertentu tanpa membuat segmen dan kampanye terlebih dahulu.

mobiletargeting:SendMessage

Kirim pesan atau pemberitahuan push ke titik akhir tertentu.

- URI — [/apps/projectId/messages](#)
- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

mobiletargeting:SendUsersMessages

Kirim pesan atau pemberitahuan push ke semua titik akhir yang terkait dengan ID pengguna tertentu.

- URI — [/apps/projectId/users-messages](#)

- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

Kata sandi satu kali

Izin berikut terkait dengan pengiriman dan validasi kata sandi satu kali (OTP) di Amazon Pinpoint.

mobiletargeting:SendOTPMessage

Kirim pesan teks yang berisi kata sandi satu kali.

- URI — [/apps/projectId/otp](#)
- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/otp`

mobiletargeting:VerifyOTPMessage

Periksa validitas kata sandi satu kali (OTP) yang dihasilkan menggunakan operasi `sendotpMessage`.

- URI — [/apps/projectId/verify-otp](#)
- Metode – POST
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/verify-otp`

Validasi nomor telepon

Izin berikut terkait dengan penggunaan layanan validasi nomor telepon di Amazon Pinpoint.

mobiletargeting:PhoneNumberValidate

Ambil informasi tentang nomor telepon.

- URI — [/phone/number/validate](#)
- Metode – POST
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:phone/number/validate`

Proyek

Izin berikut terkait dengan pengelolaan proyek di akun Amazon Pinpoint Anda. Awalnya, proyek disebut sebagai aplikasi. Untuk keperluan operasi ini, aplikasi Amazon Pinpoint sama dengan proyek Amazon Pinpoint.

mobiletargeting:CreateApp

Buat proyek Amazon Pinpoint.

- URI — [/apps](#)
- Metode – POST
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:apps`

mobiletargeting>DeleteApp

Hapus proyek Amazon Pinpoint.

- URI — [/apps/projectId](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId`

mobiletargeting:GetApp

Ambil informasi tentang proyek Amazon Pinpoint.

- URI — [/apps/projectId](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId`

mobiletargeting:GetApps

Ambil informasi tentang semua proyek yang terkait dengan akun Amazon Pinpoint Anda.

- URI — [/apps](#)
- Metode – GET
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:apps`

mobiletargeting:GetApplicationSettings

Ambil pengaturan default untuk proyek Amazon Pinpoint.

- URI — [/apps/projectId/settings](#)

- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId`

mobiletargeting:UpdateApplicationSettings

Perbarui pengaturan default untuk proyek Amazon Pinpoint.

- URI — [/apps/projectId/settings](#)
- Metode – PUT
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId`

Model rekomendasi

Izin berikut terkait dengan pengelolaan konfigurasi Amazon Pinpoint untuk mengambil dan memproses data rekomendasi dari model pemberi rekomendasi. Model pemberi rekomendasi adalah jenis model pembelajaran mesin yang memprediksi dan menghasilkan rekomendasi yang dipersonalisasi dengan menemukan pola dalam data.

mobiletargeting>CreateRecommenderConfiguration

Buat konfigurasi Amazon Pinpoint untuk model pemberi rekomendasi.

- URI — [/recommenders](#)
- Metode – POST
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:recommenders`

mobiletargeting:GetRecommenderConfigurations

Ambil informasi tentang semua konfigurasi model pemberi rekomendasi yang terkait dengan akun Amazon Pinpoint Anda.

- URI — [/recommenders](#)
- Metode – GET
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:recommenders`

mobiletargeting:GetRecommenderConfiguration

Ambil informasi tentang konfigurasi Amazon Pinpoint individual untuk model pemberi rekomendasi.

- URI — [/recommenders/recommenderId](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId

mobiletargeting:UpdateRecommenderConfiguration

Perbarui konfigurasi Amazon Pinpoint untuk model pemberi rekomendasi.

- URI — [/recommenders/recommenderId](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId

mobiletargeting>DeleteRecommenderConfiguration

Hapus konfigurasi Amazon Pinpoint untuk model pemberi rekomendasi.

- URI — [/recommenders/recommenderId](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId

Segmen

Izin berikut terkait dengan pengelolaan segmen di akun Amazon Pinpoint Anda. Di Amazon Pinpoint, segmen adalah grup penerima untuk kampanye Anda yang membagikan atribut tertentu yang Anda tentukan.

mobiletargeting>CreateSegment

Buat segmen. Untuk memungkinkan pengguna membuat segmen dengan mengimpor data titik akhir dari luar Amazon Pinpoint, izinkan tindakan tersebut.

`mobiletargeting>CreateImportJob`

- URI — [/apps/projectId/segments](#)
- Metode – POST
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId

mobiletargeting:DeleteSegment

Hapus segmen.

- URI — [/apps/projectId/segments/segmentId](#)
- Metode - HAPUS
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId/
segments/segmentId

mobiletargeting:GetSegment

Ambil informasi tentang segmen tertentu.

- URI — [/apps/projectId/segments/segmentId](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId/
segments/segmentId

mobiletargeting:GetSegmentExportJobs

Mengambil informasi tentang pekerjaan yang mengekspor definisi titik akhir untuk suatu segmen.

- URI — [/apps/projectId/segments/segmentId/jobs/export](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId/
segments/segmentId/jobs/export

mobiletargeting:GetSegments

Mengambil informasi tentang semua segmen untuk sebuah proyek.

- URI — [/apps/projectId/segments](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:region:accountId:apps/projectId

mobiletargeting:GetSegmentImportJobs

Ambil informasi tentang lowongan yang membuat segmen dengan mengimpor definisi titik akhir dari Amazon S3.

- URI — [/apps/*projectId*/segments/*segmentId*/jobs/import](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
segments/*segmentId*

mobiletargeting:GetSegmentVersion

Ambil informasi tentang versi segmen tertentu.

- URI — [/apps/*projectId*/segments/*segmentId*/versions/*versionId*](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
segments/*segmentId*

mobiletargeting:GetSegmentVersions

Ambil informasi tentang versi segmen saat ini dan sebelumnya.

- URI — [/apps/*projectId*/segments/*segmentId*/versions](#)
- Metode – GET
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
segments/*segmentId*

mobiletargeting:UpdateSegment

Perbarui segmen tertentu.

- URI — [/apps/*projectId*/segments/*segmentId*](#)
- Metode – PUT
- Sumber daya ARN —
arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/
segments/*segmentId*

Tanda

Izin berikut terkait dengan melihat dan mengelola tag untuk sumber daya Amazon Pinpoint.

mobiletargeting:ListTagsForResource

Mengambil informasi tentang tag yang terkait dengan proyek, kampanye, template pesan, atau segmen.

- URI — [/tags/resource-arn](#)
- Metode – GET
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:*`

mobiletargeting:TagResource

Tambahkan satu atau beberapa tag ke proyek, kampanye, templat pesan, atau segmen.

- URI — [/tags/resource-arn](#)
- Metode – POST
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:*`

mobiletargeting:UntagResource

Hapus satu atau beberapa tag dari proyek, kampanye, templat pesan, atau segmen.

- URI — [/tags/resource-arn](#)
- Metode - HAPUS
- Sumber daya ARN — `arn:aws:mobiletargeting:region:accountId:*`

Pengguna

Izin berikut terkait dengan pengelolaan pengguna. Di Amazon Pinpoint, pengguna berhubungan dengan individu yang menerima pesan dari Anda. Satu pengguna mungkin dikaitkan dengan lebih dari satu titik akhir.

mobiletargeting>DeleteUserEndpoints

Hapus semua titik akhir yang terkait dengan ID pengguna.

- URI — [/apps/projectId/users/userId](#)
- Metode - HAPUS
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

mobiletargeting:GetUserEndpoints

Mengambil informasi tentang semua titik akhir yang terkait dengan ID pengguna.

- URI — [/apps/*projectId*/users/*userId*](#)
- Metode – GET
- Sumber daya ARN —
`arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

Amazon Pinpoint SMS dan tindakan API versi suara 1

Bagian ini mengidentifikasi tindakan untuk fitur yang tersedia dari Amazon Pinpoint SMS dan Voice API. Ini adalah API tambahan yang menyediakan opsi lanjutan untuk menggunakan dan mengelola saluran SMS dan suara di Amazon Pinpoint. Untuk mempelajari lebih lanjut tentang API ini, lihat referensi [Amazon Pinpoint SMS dan API suara](#).

sms-voice:CreateConfigurationSet

Buat set konfigurasi untuk mengirim pesan suara.

- URI — `/sms-voice/configuration-sets`
- Metode – POST
- Sumber daya ARN — Tidak tersedia. Gunakan *.

sms-voice>DeleteConfigurationSet

Hapus set konfigurasi untuk mengirim pesan suara.

- URI — `/sms-suara/konfigurasi-sets/ ConfigurationSetName`
- Metode - HAPUS
- Sumber daya ARN — Tidak tersedia. Gunakan *.

sms-voice:GetConfigurationSetEventDestinations

Mengambil informasi tentang set konfigurasi dan tujuan acara yang dikandungnya.

- URI — `/sms-suara/konfigurasi-set/event-tujuan ConfigurationSetName`
- Metode – GET
- Sumber daya ARN — Tidak tersedia. Gunakan *.

sms-voice:CreateConfigurationSetEventDestination

Buat tujuan acara untuk acara suara.

- URI — /sms-suara/konfigurasi-set//event-tujuan *ConfigurationSetName*
- Metode – POST
- Sumber daya ARN — Tidak tersedia. Gunakan *.

sms-voice:UpdateConfigurationSetEventDestination

Perbarui tujuan acara untuk acara suara.

- URI — /sms-voice/configuration-sets//event-destinations/
ConfigurationSetNameEventDestinationName
- Metode – PUT
- Sumber daya ARN — Tidak tersedia. Gunakan *.

sms-voice>DeleteConfigurationSetEventDestination

Menghapus tujuan acara untuk acara suara.

- URI — /sms-voice/configuration-sets//event-destinations/
ConfigurationSetNameEventDestinationName
- Metode - HAPUS
- Sumber daya ARN — Tidak tersedia. Gunakan *.

sms-voice:SendVoiceMessage

Membuat dan mengirim pesan suara.

- URI — /sms-suara/suara/pesan
- Metode – POST
- Sumber daya ARN — Tidak tersedia. Gunakan *.

Amazon Pinpoint contoh kebijakan berbasis identitas

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon Pinpoint. Mereka juga tidak dapat melakukan tugas menggunakan AWS Management

Console, AWS CLI, atau AWS API. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang mereka butuhkan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan pada tab JSON](#) dalam Panduan Pengguna IAM.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol Amazon Pinpoint](#)
- [Contoh: Mengakses satu proyek Amazon Pinpoint](#)
- [Contoh: Melihat sumber daya Amazon Pinpoint berdasarkan tag](#)
- [Contoh: Memungkinkan pengguna untuk melihat izin mereka sendiri](#)
- [Contoh: Menyediakan akses ke tindakan Amazon Pinpoint API](#)
- [Contoh: Menyediakan akses ke Amazon Pinpoint SMS dan tindakan API suara](#)
- [Contoh: Membatasi akses proyek Amazon Pinpoint ke alamat IP tertentu](#)
- [Contoh: Membatasi akses Amazon Pinpoint berdasarkan tag](#)
- [Contoh: Izinkan Amazon Pinpoint mengirim email menggunakan identitas yang telah diverifikasi di Amazon SES](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon Pinpoint di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [Kebijakan yang dikelola AWS](#) atau [Kebijakan yang dikelola AWS untuk fungsi tugas](#) dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat [Kebijakan dan izin dalam IAM](#) dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Kondisi](#) dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [Validasi kebijakan IAM Access Analyzer](#) dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi akses API yang dilindungi MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan dalam IAM](#) dalam Panduan Pengguna IAM.

Menggunakan konsol Amazon Pinpoint

Untuk mengakses konsol Amazon Pinpoint, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon Pinpoint di AWS akun Anda. Jika Anda membuat kebijakan berbasis identitas yang menerapkan izin yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksud untuk entitas (pengguna atau peran) dengan kebijakan tersebut. Untuk memastikan bahwa

entitas tersebut dapat menggunakan konsol Amazon Pinpoint, lampirkan kebijakan ke entitas. Untuk informasi selengkapnya, lihat [Menambah izin untuk pengguna](#) dalam Panduan Pengguna IAM.

Kebijakan contoh berikut menyediakan akses hanya-baca ke konsol Amazon Pinpoint di Wilayah tertentu. AWS Ini mencakup akses hanya-baca ke layanan lain yang bergantung pada konsol Amazon Pinpoint, seperti Amazon Simple Email Service (Amazon SES), IAM, dan Amazon Kinesis.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UseConsole",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "firehose:ListDeliveryStreams",
        "iam:ListRoles",
        "kinesis:ListStreams",
        "s3:List*",
        "ses:Describe*",
        "ses:Get*",
        "ses:List*",
        "sns:ListTopics"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        }
      }
    }
  ]
}
```

Pada contoh kebijakan sebelumnya, ganti *wilayah dengan nama Region*, dan ganti *accountId* dengan ID *akun* Anda. AWS AWS

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Contoh: Mengakses satu proyek Amazon Pinpoint

Anda juga dapat membuat kebijakan hanya-baca yang hanya menyediakan akses ke proyek tertentu. Kebijakan contoh berikut memungkinkan pengguna masuk ke konsol dan melihat daftar proyek. Ini juga memungkinkan pengguna melihat informasi tentang sumber daya terkait untuk AWS layanan lain yang bergantung pada konsol Amazon Pinpoint, seperti Amazon SES, IAM, dan Amazon Kinesis. Namun, kebijakan ini memungkinkan pengguna melihat informasi tambahan hanya tentang proyek yang ditentukan dalam kebijakan. Anda dapat mengubah kebijakan ini untuk mengizinkan akses ke proyek atau AWS Wilayah tambahan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewProject",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetApps",
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/*",
        "arn:aws:mobiletargeting:region:accountId:reports"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ses:Get*",
        "kinesis:ListStreams",
        "firehose:ListDeliveryStreams",
        "iam:ListRoles",
```

```

        "ses:List*",
        "sns:ListTopics",
        "ses:Describe*",
        "s3:List*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "accountId"
        }
    }
}
]
}

```

Pada contoh sebelumnya, ganti *region* dengan *nama Region*, ganti *accountID* dengan ID akun AWS Anda, dan ganti *AWS ProjectID* dengan ID project Amazon Pinpoint yang *ingin* Anda berikan aksesnya.

Demikian pula, Anda dapat membuat kebijakan yang memberikan pengguna di AWS akun Anda dengan akses tulis terbatas ke salah satu proyek Amazon Pinpoint Anda, misalnya proyek yang memiliki ID `810c7aab86d42fb2b56c8c966example` proyek. Dalam hal ini, Anda ingin mengizinkan pengguna untuk melihat, menambah, dan memperbarui komponen proyek, seperti segmen dan kampanye, tetapi tidak menghapus komponen apa pun.

Selain memberikan izin `mobiletargeting:Get` dan `mobiletargeting:List` tindakan, buat kebijakan yang memberikan izin untuk tindakan berikut: `mobiletargeting:Create`; dan `mobiletargeting:Update` `mobiletargeting:Put` Ini adalah izin tambahan yang diperlukan untuk membuat dan mengelola sebagian besar komponen proyek. Sebagai contoh:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LimitedWriteProject",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetApps",
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    },
    {
      "Effect": "Allow",
      "Action": [

```

```

        "mobiletargeting:Get*",
        "mobiletargeting:List*",
        "mobiletargeting:Create*",
        "mobiletargeting:Update*",
        "mobiletargeting:Put*"
    ],
    "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/810c7aab86d42fb2b56c8c966example",
        "arn:aws:mobiletargeting:region:accountId:apps/810c7aab86d42fb2b56c8c966example/*",
        "arn:aws:mobiletargeting:region:accountId:reports"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "ses:Get*",
        "kinesis:ListStreams",
        "firehose:ListDeliveryStreams",
        "iam:ListRoles",
        "ses:List*",
        "sns:ListTopics",
        "ses:Describe*",
        "s3:List*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "accountId"
        }
    }
}
]
}

```

Contoh: Melihat sumber daya Amazon Pinpoint berdasarkan tag

Anda dapat menggunakan kondisi dalam kebijakan berbasis identitas untuk mengontrol akses ke sumber daya Amazon Pinpoint berdasarkan tag. Kebijakan contoh ini menunjukkan cara Anda membuat kebijakan semacam ini untuk memungkinkan melihat sumber daya Amazon Pinpoint. Namun, izin diberikan hanya jika tag `Owner` sumber daya memiliki nilai nama pengguna pengguna

tersebut. Kebijakan ini juga memberi izin yang diperlukan untuk menyelesaikan tindakan ini pada konsol tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListResources",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ViewResourceIfOwner",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "arn:aws:mobiletargeting:*:*:*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "userName"
        },
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:mobiletargeting:region:accountId:*"
        }
      }
    }
  ]
}
```

Anda dapat melampirkan jenis kebijakan ini ke pengguna di akun Anda. Jika pengguna bernama `richard-roe` mencoba melihat sumber daya Amazon Pinpoint, sumber daya harus diberi tag `Owner=richard-roe` atau `owner=richard-roe`. Jika tidak, aksesnya akan ditolak. Kunci tanda syarat `Owner` cocok dengan `Owner` dan `owner` karena nama kunci syarat tidak terpengaruh huruf

besar/kecil. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM JSON: Syarat](#) dalam Panduan Pengguna IAM.

Contoh: Memungkinkan pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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": "*"
    }
  ]
}
```

Contoh: Menyediakan akses ke tindakan Amazon Pinpoint API

Bagian ini memberikan contoh kebijakan yang memungkinkan akses ke fitur yang tersedia dari Amazon Pinpoint API, yang merupakan API utama untuk Amazon Pinpoint. Untuk mempelajari lebih lanjut tentang API ini, lihat Referensi [API Amazon Pinpoint](#).

Akses hanya-baca

Kebijakan contoh berikut memungkinkan akses hanya-baca ke semua sumber daya di akun Amazon Pinpoint Anda di Wilayah tertentu. AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewAllResources",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:Get*",
        "mobiletargeting:List*"
      ],
      "Resource": "arn:aws:mobiletargeting:region:accountId:*"
    }
  ]
}
```

Pada contoh sebelumnya, ganti *wilayah dengan nama Region*, dan ganti accountId dengan ID *akun* Anda. AWS AWS

Akses administrator

Kebijakan contoh berikut memungkinkan akses penuh ke semua tindakan dan sumber daya Amazon Pinpoint di akun Amazon Pinpoint Anda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FullAccess",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:*"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:mobiletargeting:region:accountId:*"
  }
]
}

```

Pada contoh sebelumnya, ganti accountId dengan ID *akun* Anda. AWS

Contoh: Menyediakan akses ke Amazon Pinpoint SMS dan tindakan API suara

Bagian ini memberikan contoh kebijakan yang memungkinkan akses ke fitur yang tersedia dari Amazon Pinpoint SMS dan Voice API. Ini adalah API tambahan yang menyediakan opsi lanjutan untuk menggunakan dan mengelola saluran SMS dan suara di Amazon Pinpoint. Untuk mempelajari lebih lanjut tentang API ini, lihat referensi [Amazon Pinpoint SMS dan API suara](#).

Akses hanya-baca

Kebijakan contoh berikut memungkinkan akses hanya-baca ke semua tindakan dan sumber daya Amazon Pinpoint SMS dan Voice API di akun Anda: AWS

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMSVoiceReadOnly",
      "Effect": "Allow",
      "Action": [
        "sms-voice:Get*",
        "sms-voice:List*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sms-voice:region:accountId:*"
        }
      }
    }
  ]
}

```

Akses administrator

Kebijakan contoh berikut memungkinkan akses penuh ke semua tindakan dan sumber daya Amazon Pinpoint SMS dan Voice API di akun Anda AWS :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMSVoiceFullAccess",
      "Effect": "Allow",
      "Action": [
        "sms-voice:*",
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sms-voice:region:accountId:*"
        }
      }
    }
  ]
}
```

Contoh: Membatasi akses proyek Amazon Pinpoint ke alamat IP tertentu

*Contoh kebijakan berikut memberikan izin kepada pengguna mana pun untuk melakukan tindakan Amazon Pinpoint apa pun pada proyek tertentu (*ProjectID*). Namun, permintaan harus berasal dari kisaran alamat IP yang ditentukan dalam kondisi.*

Kondisi dalam pernyataan ini mengidentifikasi `54.240.143.*` kisaran alamat Internet Protocol versi 4 (IPv4) yang diizinkan, dengan satu pengecualian: `54.240.143.188` ConditionBlok menggunakan `IpAddress` dan `NotIpAddress` kondisi dan kunci `aws:SourceIp` kondisi, yang merupakan kunci kondisi AWS-wide. Untuk informasi selengkapnya tentang kunci kondisi ini, lihat [Menentukan kondisi dalam Panduan Pengguna IAM kebijakan](#). Nilai `aws:SourceIp` IPv4 menggunakan notasi CIDR standar. Untuk informasi selengkapnya, lihat [operator kondisi alamat IP](#) di Panduan Pengguna IAM.

```

{
  "Version": "2012-10-17",
  "Id": "AMZPinpointPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "mobiletargeting:*",
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "54.240.143.0/24"
        },
        "NotIpAddress": {
          "aws:SourceIp": "54.240.143.188/32"
        }
      }
    }
  ]
}

```

Contoh: Membatasi akses Amazon Pinpoint berdasarkan tag

Contoh kebijakan berikut memberikan izin untuk melakukan tindakan Amazon Pinpoint pada proyek tertentu (ProjectID). Namun, izin diberikan hanya jika permintaan berasal dari pengguna yang namanya merupakan nilai dalam tag Owner sumber daya untuk proyek, sebagaimana ditentukan dalam kondisi.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ModifyResourceIfOwner",
      "Effect": "Allow",
      "Action": "mobiletargeting:*",
      "Resource": [
        "arn:aws:mobiletargeting:region:accountId:apps/projectId",
        "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
      ]
    }
  ]
}

```

```

    ],
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/Owner": "userName"
      }
    }
  }
]
}

```

Contoh: Izinkan Amazon Pinpoint mengirim email menggunakan identitas yang telah diverifikasi di Amazon SES

Saat Anda memverifikasi identitas email (seperti alamat email atau domain) melalui konsol Amazon Pinpoint, identitas tersebut secara otomatis dikonfigurasi sehingga dapat digunakan oleh Amazon Pinpoint dan Amazon SES. Namun, jika Anda memverifikasi identitas email melalui Amazon SES, dan Anda ingin menggunakan identitas itu dengan Amazon Pinpoint, Anda harus menerapkan kebijakan untuk identitas itu.

Contoh kebijakan berikut memberikan izin Amazon Pinpoint untuk mengirim email menggunakan identitas email yang telah diverifikasi melalui Amazon SES.

```

{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "PinpointEmail",
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "ses:*",
      "Resource": "arn:aws:ses:region:accountId:identity/emailId",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:mobiletargeting:region:accountId:apps/*"
        }
      }
    }
  ]
}

```

```
    ]
  }
}
```

Jika Anda menggunakan Amazon Pinpoint di Wilayah AWS GovCloud (AS-Barat), gunakan contoh kebijakan berikut sebagai gantinya:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "PinpointEmail",
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "ses:*",
      "Resource": "arn:aws-us-gov:ses:us-gov-west-1:accountId:identity/emailId",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws-us-gov:mobiletargeting:us-gov-west-1:accountId:apps/*"
        }
      }
    }
  ]
}
```

Peran IAM untuk tugas Amazon Pinpoint umum

[Peran IAM](#) adalah identitas AWS Identity and Access Management (IAM) yang dapat Anda buat di AWS akun Anda dan memberikan izin tertentu. Peran IAM adalah AWS identitas dengan kebijakan izin yang menentukan apa yang dapat dan tidak dapat dilakukan identitas. AWS Namun, alih-alih dikaitkan secara unik dengan satu orang, peran dapat diasumsikan oleh siapa saja yang membutuhkannya.

Juga, peran tidak memiliki kredensi jangka panjang standar yang terkait dengannya. Sebaliknya, ia menyediakan kredensial keamanan sementara untuk sesi. Anda dapat menggunakan peran IAM untuk

mendelegasikan akses ke pengguna, aplikasi, aplikasi, atau layanan yang biasanya tidak memiliki akses ke sumber daya Anda AWS .

Untuk alasan ini, Anda dapat menggunakan peran IAM untuk mengintegrasikan Amazon Pinpoint dengan layanan dan sumber daya AWS tertentu untuk akun Anda. Misalnya, Anda mungkin ingin mengizinkan Amazon Pinpoint mengakses definisi titik akhir yang disimpan di bucket Amazon Simple Storage Service (Amazon S3) dan ingin digunakan untuk segmen. Atau Anda mungkin ingin mengizinkan Amazon Pinpoint untuk mengalirkan data peristiwa ke aliran Amazon Kinesis untuk akun Anda. Demikian pula, Anda mungkin ingin menggunakan peran IAM untuk memungkinkan web atau aplikasi seluler mendaftarkan titik akhir atau melaporkan data penggunaan untuk proyek Amazon Pinpoint, tanpa AWS menyimpan kunci di aplikasi (di mana mereka bisa sulit diputar dan pengguna berpotensi mengekstraknya).

Untuk skenario ini, Anda dapat mendelegasikan akses ke Amazon Pinpoint dengan menggunakan peran IAM. Bagian ini menjelaskan dan memberikan contoh tugas Amazon Pinpoint umum yang menggunakan peran IAM untuk bekerja dengan layanan lain. AWS Untuk informasi tentang penggunaan peran IAM dengan aplikasi web dan seluler secara lebih spesifik, lihat [Menyediakan akses ke pengguna yang diautentikasi secara eksternal \(federasi identitas\) di Panduan Pengguna IAM](#).

Topik

- [Peran IAM untuk mengimpor titik akhir atau segmen](#)
- [Peran IAM untuk mengekspor titik akhir atau segmen](#)
- [Peran IAM untuk mengambil rekomendasi dari Amazon Personalize](#)
- [Peran IAM untuk streaming acara ke Kinesis](#)
- [Peran IAM untuk mengirim email dengan Amazon SES](#)

Peran IAM untuk mengimpor titik akhir atau segmen

Dengan Amazon Pinpoint, Anda dapat menentukan segmen pengguna dengan mengimpor definisi titik akhir dari bucket Amazon Simple Storage Service (Amazon S3) di akun Anda. AWS Sebelum mengimpor, Anda harus mendelegasikan izin yang diperlukan ke Amazon Pinpoint. Untuk melakukan ini, Anda membuat peran AWS Identity and Access Management (IAM) dan melampirkan kebijakan berikut ke peran:

- Kebijakan yang `AmazonS3ReadOnlyAccess` AWS dikelola. Kebijakan ini dibuat dan dikelola oleh AWS, dan memberikan akses hanya-baca ke bucket Amazon S3 Anda.

- Kebijakan kepercayaan yang memungkinkan Amazon Pinpoint untuk mengambil peran.

Setelah membuat peran, Anda dapat menggunakan Amazon Pinpoint untuk mengimpor segmen dari bucket Amazon S3. Untuk informasi tentang membuat bucket, membuat file titik akhir, dan mengimpor segmen menggunakan konsol, lihat [Mengimpor segmen](#) di Panduan Pengguna Amazon Pinpoint. Untuk contoh cara mengimpor segmen secara terprogram menggunakan AWS SDK for Java, lihat [Mengimpor segmen](#) di panduan ini.

Membuat peran IAM ()AWS CLI

Selesaikan langkah-langkah berikut untuk membuat peran IAM dengan menggunakan AWS Command Line Interface (AWS CLI). Jika Anda belum menginstal AWS CLI, lihat [Menginstal AWS CLI di](#) Panduan AWS Command Line Interface Pengguna.

Untuk membuat peran IAM dengan menggunakan AWS CLI

1. Buat file JSON yang berisi kebijakan kepercayaan untuk peran Anda, dan simpan file secara lokal. Anda dapat menggunakan kebijakan kepercayaan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
          "arn:aws:mobiletargeting:region:accountId:apps/application-id"
        }
      }
    }
  ]
}
```

Pada contoh sebelumnya, lakukan hal berikut:

- Ganti *wilayah* dengan AWS Wilayah tempat Anda menggunakan Amazon Pinpoint.
 - Ganti *accountID* dengan ID unik untuk akun Anda. AWS
 - Ganti *application-id* dengan ID unik proyek.
2. Pada baris perintah, gunakan [create-role](#) perintah untuk membuat peran dan lampirkan kebijakan kepercayaan:

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document file://PinpointImportTrustPolicy.json
```

Mengikuti `file://` awalan, tentukan jalur ke file JSON yang berisi kebijakan kepercayaan.

Setelah Anda menjalankan perintah ini, Anda melihat output yang mirip dengan yang berikut di terminal Anda:

```
{
  "Role": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
            "Service": "pinpoint.amazonaws.com"
          },
          "Condition": {
            "StringEquals": {
              "aws:SourceAccount": "accountId"
            },
            "ArnLike": {
              "aws:SourceArn":
                "arn:aws:mobiletargeting:region:accountId:apps/application-id"
            }
          }
        }
      ]
    },
    "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
```

```
"CreateDate": "2016-12-20T00:44:37.406Z",
"RoleName": "PinpointSegmentImport",
"Path": "/",
"Arn": "arn:aws:iam::accountId:role/PinpointSegmentImport"
}
}
```

- Gunakan [attach-role-policy](#) perintah untuk melampirkan kebijakan AmazonS3ReadOnlyAccess AWS terkelola ke peran:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport
```

Peran IAM untuk mengekspor titik akhir atau segmen

Anda dapat memperoleh daftar titik akhir dengan membuat pekerjaan ekspor. Saat membuat pekerjaan ekspor, Anda harus menentukan ID proyek, dan Anda dapat menentukan ID segmen secara opsional. Amazon Pinpoint kemudian mengekspor daftar titik akhir yang terkait dengan proyek atau segmen ke bucket Amazon Simple Storage Service (Amazon S3). File yang dihasilkan berisi daftar titik akhir yang diformat JSON dan atributnya, seperti saluran, alamat, status opt-in/opt-out, tanggal pembuatan, dan ID titik akhir.

Untuk membuat pekerjaan ekspor, Anda harus mengonfigurasi peran IAM yang memungkinkan Amazon Pinpoint menulis ke bucket Amazon S3. Proses mengkonfigurasi peran terdiri dari dua langkah:

- Buat kebijakan IAM yang memungkinkan entitas (dalam hal ini, Amazon Pinpoint) untuk menulis ke bucket Amazon S3 tertentu.
- Buat peran IAM dan lampirkan kebijakan padanya.

Topik ini berisi prosedur untuk menyelesaikan kedua langkah ini. Prosedur ini mengasumsikan bahwa Anda telah membuat bucket Amazon S3, dan folder di bucket tersebut, untuk menyimpan segmen yang diekspor. Untuk informasi tentang membuat bucket, lihat [Membuat bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Prosedur ini juga mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi AWS Command Line Interface (AWS CLI). Untuk informasi tentang pengaturan AWS CLI, lihat [Menginstal AWS CLI di](#) Panduan AWS Command Line Interface Pengguna.

Langkah 1: Buat kebijakan IAM

Kebijakan IAM mendefinisikan izin untuk entitas, seperti identitas atau sumber daya. Untuk membuat peran untuk mengekspor titik akhir Amazon Pinpoint, Anda harus membuat kebijakan yang memberikan izin untuk menulis ke folder tertentu di bucket Amazon S3 tertentu. Contoh kebijakan berikut mengikuti praktik keamanan pemberian hak istimewa paling sedikit—yaitu, hanya memberikan izin yang diperlukan untuk melakukan satu tugas.

Untuk membuat kebijakan IAM

1. Di editor teks, buat file baru. Tempel kode berikut ke file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUserToSeeBucketListInTheConsole",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": [ "arn:aws:s3:::*" ]
    },
    {
      "Sid": "AllowRootAndHomeListingOfBucket",
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [ "arn:aws:s3:::example-bucket" ],
      "Condition": {
        "StringEquals": {
          "s3:delimiter": [ "/" ],
          "s3:prefix": [
            "",
            "Exports/"
          ]
        }
      }
    },
    {
      "Sid": "AllowListingOfUserFolder",
```

```

    "Action": [
      "s3:ListBucket"
    ],
    "Effect": "Allow",
    "Resource": [ "arn:aws:s3:::example-bucket" ],
    "Condition": {
      "StringLike": {
        "s3:prefix": [
          "Exports/*"
        ]
      }
    }
  },
  {
    "Sid": "AllowAllS3ActionsInUserFolder",
    "Action": [ "s3:*" ],
    "Effect": "Allow",
    "Resource": [ "arn:aws:s3:::example-bucket/Exports/*" ]
  }
]
}

```

Pada kode sebelumnya, ganti semua instance *example-bucket* dengan nama bucket Amazon S3 yang berisi folder tempat Anda ingin mengekspor informasi segmen. Juga, ganti semua contoh *Ekspor* dengan nama folder itu sendiri.

Setelah selesai, simpan file sebagai `s3policy.json`.

2. Dengan menggunakan AWS CLI, navigasikan ke direktori tempat `s3policy.json` file berada. Kemudian masukkan perintah berikut untuk membuat kebijakan:

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
file://s3policy.json
```

Jika kebijakan berhasil dibuat, Anda akan melihat output yang mirip dengan berikut ini:

```

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

Salin Nama Sumber Daya Amazon (ARN) kebijakan

(arn:aws:iam::123456789012:policy/s3ExportPolicy dalam contoh sebelumnya). Di bagian selanjutnya, Anda harus menyediakan ARN ini saat Anda membuat peran.

Note

Jika Anda melihat pesan yang menyatakan bahwa akun Anda tidak diizinkan untuk melakukan CreatePolicy operasi, Anda harus melampirkan kebijakan ke pengguna yang memungkinkan Anda membuat kebijakan dan peran IAM baru. Untuk informasi lebih lanjut, lihat [Menambahkan dan menghapus izin identitas IAM](#) dalam Panduan Pengguna IAM.

Langkah 2: Buat peran IAM

Setelah membuat kebijakan IAM, Anda dapat membuat peran dan melampirkan kebijakan tersebut. Setiap peran IAM berisi kebijakan kepercayaan — seperangkat aturan yang menentukan entitas mana yang diizinkan untuk mengambil peran tersebut. Di bagian ini, Anda membuat kebijakan kepercayaan yang memungkinkan Amazon Pinpoint untuk mengambil peran tersebut. Selanjutnya, Anda membuat peran itu sendiri, lalu lampirkan kebijakan yang Anda buat di bagian sebelumnya.

Untuk membuat peran IAM

1. Di editor teks, buat file baru. Tempel kode berikut ke file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      }
    }
  ]
}
```

```

    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "accountId"
      },
      "ArnLike": {
        "aws:SourceArn":
          "arn:aws:mobiletargeting:region:accountId:apps/applicationId"
      }
    }
  }
]
}

```

Simpan file sebagai `trustpolicy.json`.

2. Dengan menggunakan AWS CLI, navigasikan ke direktori tempat `trustpolicy.json` file berada. Kemudian masukkan perintah berikut untuk membuat peran baru:

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
file://trustpolicy.json
```

3. Di baris perintah, masukkan perintah berikut untuk melampirkan kebijakan yang Anda buat di bagian sebelumnya ke peran yang baru saja Anda buat:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
s3ExportPolicy --role-name s3ExportRole
```

Pada perintah sebelumnya, ganti `arn:aws:iam:ExportPolicy :123456789012:policy/s3` dengan ARN kebijakan yang Anda buat di bagian sebelumnya.

Peran IAM untuk mengambil rekomendasi dari Amazon Personalize

Anda dapat mengonfigurasi Amazon Pinpoint untuk mengambil data rekomendasi dari solusi Personalisasi Amazon yang telah digunakan sebagai kampanye Personalisasi Amazon. Anda dapat menggunakan data ini untuk mengirim rekomendasi yang dipersonalisasi kepada penerima pesan berdasarkan atribut dan perilaku masing-masing penerima. Untuk mempelajari lebih lanjut, lihat [Model pembelajaran mesin](#) di Panduan Pengguna Amazon Pinpoint.

Sebelum Anda dapat mengambil data rekomendasi dari kampanye Amazon Personalize, Anda harus membuat AWS Identity and Access Management peran (IAM) yang memungkinkan Amazon Pinpoint untuk mengambil data dari kampanye. Amazon Pinpoint dapat membuat peran ini untuk Anda secara otomatis saat Anda menggunakan konsol untuk menyiapkan model pemberi rekomendasi di Amazon Pinpoint. Atau, Anda dapat membuat peran ini secara manual.

Untuk membuat peran secara manual, gunakan IAM API untuk menyelesaikan langkah-langkah berikut:

1. Buat kebijakan IAM yang memungkinkan entitas (dalam hal ini, Amazon Pinpoint) untuk mengambil data rekomendasi dari kampanye Amazon Personalize.
2. Buat IAM role dan lampirkan kebijakan IAM ke IAM role tersebut.

Topik ini menjelaskan cara menyelesaikan langkah-langkah ini dengan menggunakan AWS Command Line Interface (AWS CLI). Ini mengasumsikan bahwa Anda telah membuat solusi Amazon Personalize dan menerapkannya sebagai kampanye Amazon Personalize. Untuk informasi tentang membuat dan menerapkan kampanye, lihat [Membuat kampanye di Panduan](#) Pengembang Personalisasi Amazon.

Topik ini juga mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi AWS CLI Untuk informasi tentang pengaturan AWS CLI, lihat [Menginstal AWS CLI di](#) Panduan AWS Command Line Interface Pengguna.

Langkah 1: Buat kebijakan IAM

Kebijakan IAM mendefinisikan izin untuk entitas, seperti identitas atau sumber daya. Untuk membuat peran yang memungkinkan Amazon Pinpoint mengambil data rekomendasi dari kampanye Amazon Personalize, Anda harus terlebih dahulu membuat kebijakan IAM untuk peran tersebut. Kebijakan ini perlu mengizinkan Amazon Pinpoint untuk:

- Ambil informasi konfigurasi untuk solusi yang diterapkan oleh campaign (`DescribeSolution`).
- Periksa status kampanye (`DescribeCampaign`).
- Mengambil data rekomendasi dari kampanye (`GetRecommendations`).

Dalam prosedur berikut, kebijakan contoh memungkinkan akses ini untuk solusi Amazon Personalize tertentu yang digunakan oleh kampanye Amazon Personalize tertentu.

Untuk membuat kebijakan IAM

1. Di editor teks, buat file baru. Tempel kode berikut ke file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RetrieveRecommendationsOneCampaign",
      "Effect": "Allow",
      "Action": [
        "personalize:DescribeSolution",
        "personalize:DescribeCampaign",
        "personalize:GetRecommendations"
      ],
      "Resource": [
        "arn:aws:personalize:region:accountId:solution/solutionId",
        "arn:aws:personalize:region:accountId:campaign/campaignId"
      ]
    }
  ]
}
```

Pada contoh sebelumnya, ganti teks yang *dicetak miring* dengan informasi Anda:

- *region* — Nama AWS Wilayah yang menjadi tuan rumah solusi dan kampanye Amazon Personalize.
 - *AccountID* — ID Anda. Akun AWS
 - *SolutionId* — ID sumber daya unik untuk solusi Amazon Personalize yang digunakan oleh kampanye.
 - *Campaignid* — ID sumber daya unik untuk kampanye Amazon Personalize untuk mengambil data rekomendasi.
2. Setelah selesai, simpan file sebagai `RetrieveRecommendationsPolicy.json`.
 3. Dengan menggunakan antarmuka baris perintah, arahkan ke direktori tempat Anda menyimpan `RetrieveRecommendationsPolicy.json` file.
 4. Masukkan perintah berikut untuk membuat kebijakan dan beri nama `RetrieveRecommendationsPolicy`. Untuk menggunakan nama yang berbeda, ubah *RetrieveRecommendationsPolicy* ke nama yang Anda inginkan.

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document
file://RetrieveRecommendationsPolicy.json
```

Note

Jika Anda menerima pesan bahwa akun Anda tidak diizinkan untuk melakukan CreatePolicy operasi, Anda harus melampirkan kebijakan kepada pengguna yang memungkinkan Anda membuat kebijakan dan peran IAM baru untuk akun Anda. Untuk informasi lebih lanjut, lihat [Menambahkan dan menghapus izin identitas IAM](#) dalam Panduan Pengguna IAM.

5. Salin Nama Sumber Daya Amazon (ARN) kebijakan (arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy dalam contoh sebelumnya). Anda memerlukan ARN ini untuk membuat peran IAM di bagian berikutnya.

Langkah 2: Buat peran IAM

Setelah membuat kebijakan IAM, Anda dapat membuat peran IAM dan melampirkan kebijakan tersebut.

Setiap peran IAM berisi kebijakan kepercayaan, yang merupakan seperangkat aturan yang menentukan entitas mana yang diizinkan untuk mengambil peran tersebut. Di bagian ini, Anda membuat kebijakan kepercayaan yang memungkinkan Amazon Pinpoint untuk mengambil peran tersebut. Selanjutnya, Anda membuat peran itu sendiri. Kemudian, Anda melampirkan kebijakan ke peran tersebut.

Untuk membuat peran IAM

1. Di editor teks, buat file baru. Tempel kode berikut ke file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
```

```

        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "accountId"
            },
            "ArnLike": {
                "AWS:SourceArn":
                "arn:aws:mobiletargeting:region:accountId:apps/*"
            }
        }
    ]
}

```

2. Simpan file sebagai RecommendationsTrustPolicy.json.
3. Dengan menggunakan antarmuka baris perintah, arahkan ke direktori tempat Anda menyimpan RecommendationsTrustPolicy.json file.
4. Masukkan perintah berikut untuk membuat peran baru dan beri nama PinpointRoleforPersonalize. Untuk menggunakan nama yang berbeda, ubah *PinpointRoleforPersonalize* ke nama yang Anda inginkan.

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-document file://RecommendationsTrustPolicy.json
```

5. Masukkan perintah berikut untuk melampirkan kebijakan yang Anda buat di bagian sebelumnya ke peran yang baru saja Anda buat:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

Pada perintah sebelumnya, ganti *arn:aws:iam:*
RetrieveRecommendationsPolicy :123456789012:policy/ dengan ARN dari kebijakan yang Anda buat di bagian sebelumnya. Juga, ganti *PinpointRoleforPersonalize* dengan nama peran yang Anda tentukan di langkah 4, jika Anda menentukan nama yang berbeda untuk peran tersebut.

Peran IAM untuk streaming acara ke Kinesis

Amazon Pinpoint dapat secara otomatis mengirim data penggunaan aplikasi, atau data peristiwa, dari aplikasi Anda ke aliran data Amazon Kinesis atau aliran pengiriman Amazon Data Firehose

di akun Anda. AWS Sebelum Amazon Pinpoint dapat mulai streaming data acara, Anda harus mendelegasikan izin yang diperlukan ke Amazon Pinpoint.

Jika Anda menggunakan konsol untuk mengatur streaming acara, Amazon Pinpoint secara otomatis membuat peran AWS Identity and Access Management (IAM) dengan izin yang diperlukan. Untuk informasi selengkapnya, lihat [Streaming peristiwa Amazon Pinpoint ke Amazon Kinesis di Panduan Pengguna](#) Amazon Pinpoint.

Jika Anda ingin membuat peran secara manual, lampirkan kebijakan berikut ke peran:

- Kebijakan izin yang memungkinkan Amazon Pinpoint mengirim data peristiwa ke aliran Anda.
- Kebijakan kepercayaan yang memungkinkan Amazon Pinpoint untuk mengambil peran.

Setelah membuat peran, Anda dapat mengonfigurasi Amazon Pinpoint untuk secara otomatis mengirim acara ke streaming Anda. Untuk informasi selengkapnya, lihat [Streaming acara Amazon Pinpoint ke Kinesis](#) dalam panduan ini.

Membuat peran IAM (AWS CLI)

Selesaikan langkah-langkah berikut untuk membuat peran IAM secara manual dengan menggunakan AWS Command Line Interface (AWS CLI). Untuk mempelajari cara membuat peran menggunakan konsol Amazon Pinpoint, lihat [Streaming peristiwa Amazon Pinpoint ke Kinesis](#) di Panduan Pengguna Amazon Pinpoint.

Jika Anda belum menginstal AWS CLI, lihat [Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna](#). Anda juga harus membuat aliran Kinesis atau aliran Firehose. Untuk informasi tentang cara membuat sumber daya ini, lihat [Membuat dan Mengelola Aliran](#) di Panduan Pengembang Amazon Kinesis Data Streams atau [Membuat aliran pengiriman Amazon Data Firehose](#) di Panduan Pengembang [Amazon Data Firehose](#).

Untuk membuat peran IAM dengan menggunakan AWS CLI

1. Buat file baru. Tempelkan kebijakan berikut ke dalam dokumen dan buat perubahan berikut:
 - Ganti *wilayah* dengan AWS Wilayah tempat Anda menggunakan Amazon Pinpoint.
 - Ganti *accountID* dengan ID unik untuk akun Anda. AWS
 - Ganti *applicationId* dengan ID unik proyek.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "pinpoint.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "accountId"
      },
      "ArnLike": {
        "aws:SourceArn":
"arn:aws:mobiletargeting:region:accountId:apps/applicationId"
      }
    }
  }
]
}

```

Setelah selesai, simpan file sebagai `PinpointEventStreamTrustPolicy.json`.

- Gunakan `create-role` perintah untuk membuat peran dan lampirkan kebijakan kepercayaan:

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-document file://PinpointEventStreamTrustPolicy.json
```

- Buat file baru yang berisi kebijakan izin untuk peran Anda.

Jika Anda mengonfigurasi Amazon Pinpoint untuk mengirim data ke aliran Kinesis, tempelkan kebijakan berikut ke dalam file dan ganti yang berikut ini:

- Ganti *wilayah* dengan AWS Wilayah tempat Anda menggunakan Amazon Pinpoint.
- Ganti *accountId* dengan ID unik untuk akun Anda. AWS
- Ganti *streamName* dengan nama aliran Kinesis Anda.

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Action": [

```

```

        "kinesis:PutRecords",
        "kinesis:DescribeStream"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:kinesis:region:accountId:stream/streamName"
    ]
}
}

```

Atau, jika Anda mengonfigurasi Amazon Pinpoint untuk mengirim data ke aliran Firehose, tempelkan kebijakan berikut ke dalam file dan ganti yang berikut ini:

- Ganti *wilayah* dengan AWS Wilayah tempat Anda menggunakan Amazon Pinpoint.
- Ganti *accountId* dengan ID unik untuk akun Anda. AWS
- Ganti *delivery-stream-name* dengan nama aliran Firehose Anda.

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "firehose:PutRecordBatch",
      "firehose:DescribeDeliveryStream"
    ],
    "Resource": [
      "arn:aws:firehose:region:accountId:deliverystream/delivery-stream-name"
    ]
  }
}

```

Setelah selesai, simpan file sebagai `PinpointEventStreamPermissionsPolicy.json`.

4. Gunakan `put-role-policy` perintah untuk melampirkan kebijakan izin ke peran:

```

aws iam put-role-policy --role-name PinpointEventStreamRole --policy-
name PinpointEventStreamPermissionsPolicy --policy-document file://
PinpointEventStreamPermissionsPolicy.json

```

Peran IAM untuk mengirim email dengan Amazon SES

Amazon Pinpoint menggunakan sumber daya Amazon SES Anda untuk mengirim email untuk kampanye atau perjalanan Anda. Sebelum Amazon Pinpoint dapat menggunakan sumber daya Amazon SES Anda untuk mengirim email, Anda harus memberikan izin yang diperlukan ke Amazon Pinpoint. Akun Anda harus memiliki `iam:UpdateAssumeRolePolicy` izin `iam:PutRolePolicy` dan untuk memperbarui atau membuat peran IAM.

Konsol Amazon Pinpoint dapat secara otomatis membuat peran AWS Identity and Access Management (IAM) dengan izin yang diperlukan. Untuk informasi selengkapnya, lihat [Membuat peran pengiriman orkestrasi email di Panduan Pengguna Amazon Pinpoint](#).

Jika Anda ingin membuat peran secara manual, lampirkan kebijakan berikut ke peran:

- Kebijakan izin yang memberikan Amazon Pinpoint akses ke sumber daya Amazon SES Anda.
- Kebijakan kepercayaan yang memungkinkan Amazon Pinpoint untuk mengambil peran.

Setelah membuat peran, Anda dapat mengonfigurasi Amazon Pinpoint untuk menggunakan sumber daya Amazon SES Anda.

Anda dapat menguji kebijakan IAM dengan simulator kebijakan IAM. Untuk informasi selengkapnya, lihat [Menguji kebijakan IAM dengan simulator kebijakan IAM](#) di Panduan Pengguna [IAM](#).

Membuat peran IAM ()AWS Management Console

Selesaikan langkah-langkah berikut untuk membuat peran IAM secara manual untuk kampanye atau perjalanan Anda untuk mengirim email.

1. Buat kebijakan izin baru dengan mengikuti petunjuk dalam [Membuat kebijakan menggunakan editor JSON](#) di [Panduan Pengguna IAM](#).
 - Pada [langkah 5](#), gunakan kebijakan izin berikut untuk peran IAM.
 - Ganti *partisi* dengan partisi tempat sumber daya berada. Untuk standar Wilayah AWS, partisi adalah `aws`. Jika Anda memiliki sumber daya di partisi lain, maka partisi-nya adalah `aws-partitionname`. Misalnya, partisi untuk sumber daya di AWS GovCloud (AS-Barat) adalah `aws-us-gov`.
 - Ganti *wilayah* dengan nama Wilayah AWS yang menghosting proyek Amazon Pinpoint.
 - Ganti *accountID* dengan ID unik untuk Anda. Akun AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PinpointUsesSESForEmailSends",
      "Effect": "Allow",
      "Action": [
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource": [
        "arn:partition:ses:region:accountId:identity/*",
        "arn:partition:ses:region:accountId:configuration-set/*"
      ]
    }
  ]
}
```

2. Buat kebijakan kepercayaan baru dengan mengikuti petunjuk dalam [Membuat peran menggunakan kebijakan kepercayaan khusus](#) di [Panduan Pengguna IAM](#).

a. Pada [langkah 4](#), gunakan kebijakan kepercayaan berikut.

- Ganti *accountId* dengan ID unik untuk Anda. Akun AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPinpoint",
      "Effect": "Allow",
      "Principal": {
        "Service": "pinpoint.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        }
      }
    }
  ]
}
```



```
    }  
  ]  
}
```

- b. Pada [langkah 11](#), tambahkan kebijakan izin yang Anda buat di langkah sebelumnya.

Memecahkan masalah Amazon Pinpoint identitas dan manajemen akses

Gunakan informasi berikut untuk mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon Pinpoint dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di Amazon Pinpoint](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses sumber daya Amazon Pinpoint saya](#)

Saya tidak berwenang untuk melakukan tindakan di Amazon Pinpoint

Jika AWS Management Console memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk Anda.

Contoh kesalahan berikut terjadi ketika mateojackson pengguna mencoba menggunakan konsol untuk melihat detail tentang proyek tetapi tidak memiliki `mobiletargeting:GetApp` izin.

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

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya untuk mengizinkan dia mengakses sumber daya `my-example-project` menggunakan tindakan `mobiletargeting:GetApp`.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon Pinpoint.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amazon Pinpoint. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

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

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses sumber daya Amazon Pinpoint saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah Amazon Pinpoint mendukung fitur-fitur ini, lihat [Bagaimana Amazon Pinpoint bekerja dengan IAM](#)
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(federasi identitas\)](#) dalam Panduan Pengguna IAM.

- Untuk mempelajari perbedaan antara penggunaan kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.

Pencatatan dan pemantauan di Amazon Pinpoint

Pencatatan dan pemantauan adalah bagian penting untuk menjaga keandalan, ketersediaan, dan kinerja proyek Amazon Pinpoint Anda dan jenis sumber daya Amazon Pinpoint lainnya. Anda harus mencatat dan mengumpulkan data pemantauan dari semua bagian proyek dan sumber daya Amazon Pinpoint Anda agar lebih mudah men-debug kegagalan multipoint jika terjadi. AWS menyediakan beberapa alat yang dapat membantu Anda mencatat dan mengumpulkan data ini, dan menanggapi potensi insiden:

AWS CloudTrail

Amazon Pinpoint terintegrasi dengan AWS CloudTrail, yang merupakan layanan yang menyediakan catatan tindakan yang diambil di Amazon Pinpoint oleh pengguna, peran, atau layanan lain. AWS Ini termasuk tindakan dari konsol Amazon Pinpoint dan panggilan terprogram ke operasi Amazon Pinpoint API. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan mana yang dibuat ke Amazon Pinpoint. Untuk setiap permintaan, Anda dapat mengidentifikasi waktu permintaan itu dibuat, tempat alamat IP itu dibuat, siapa yang membuat permintaan, dan detail tambahan lainnya. Untuk informasi selengkapnya, lihat [Mencatat panggilan API Amazon Pinpoint dengan AWS CloudTrail](#) dalam panduan ini.

Amazon CloudWatch

Anda dapat menggunakan Amazon CloudWatch untuk mengumpulkan, melihat, dan menganalisis beberapa metrik penting yang terkait dengan akun dan proyek Amazon Pinpoint Anda. Anda juga dapat menggunakan CloudWatch untuk membuat alarm yang memberi tahu Anda jika nilai untuk metrik memenuhi kondisi tertentu dan berada dalam atau melebihi ambang batas yang Anda tentukan. Jika Anda membuat alarm, CloudWatch kirimkan notifikasi ke topik Amazon Simple Notification Service (Amazon SNS) yang Anda tentukan. Untuk informasi selengkapnya, lihat [Memantau Amazon Pinpoint dengan amazon CloudWatch](#) di Panduan Pengguna Amazon Pinpoint.

AWS Health Dasbor

Dengan menggunakan AWS Health dasbor, Anda dapat memeriksa dan memantau status lingkungan Amazon Pinpoint Anda. Untuk memeriksa status layanan Amazon Pinpoint secara keseluruhan, gunakan Dashboard AWS Service Health. Untuk memeriksa, memantau, dan melihat data historis tentang peristiwa atau masalah apa pun yang mungkin memengaruhi AWS lingkungan Anda secara lebih spesifik, gunakan Dasbor AWS Personal Health. Untuk mempelajari selengkapnya tentang dasbor ini, lihat [Panduan AWS Health Pengguna](#).

AWS Trusted Advisor

AWS Trusted Advisor memeriksa AWS lingkungan Anda dan memberikan rekomendasi untuk peluang untuk mengatasi kesenjangan keamanan, meningkatkan ketersediaan dan kinerja sistem, dan menghemat uang. Semua AWS pelanggan memiliki akses ke serangkaian Trusted Advisor pemeriksaan inti. Pelanggan yang memiliki paket dukungan Bisnis atau Perusahaan memiliki akses ke Trusted Advisor pemeriksaan tambahan.

Banyak dari pemeriksaan ini dapat membantu Anda menilai postur keamanan sumber daya Amazon Pinpoint Anda sebagai bagian dari AWS akun Anda secara keseluruhan. Misalnya, rangkaian Trusted Advisor pemeriksaan inti meliputi yang berikut:

- Konfigurasi logging untuk AWS akun Anda, untuk setiap AWS Wilayah yang didukung.
- Akses izin untuk bucket Amazon Simple Storage Service (Amazon S3), yang mungkin berisi file yang Anda impor ke Amazon Pinpoint untuk membangun segmen.
- Penggunaan AWS Identity and Access Management pengguna, grup, dan peran untuk mengontrol akses ke sumber daya Amazon Pinpoint.
- Konfigurasi IAM dan setelan kebijakan yang mungkin membahayakan keamanan AWS lingkungan Anda dan sumber daya Amazon Pinpoint.

Untuk informasi selengkapnya, lihat [AWS Trusted Advisor](#) di AWS Support Panduan Pengguna.

Validasi kepatuhan untuk Amazon Pinpoint

Auditor pihak ketiga menilai keamanan dan kepatuhan Amazon Pinpoint sebagai bagian dari AWS beberapa program kepatuhan. Ini termasuk AWS Sistem dan Kontrol Organisasi (SOC), FedRAMP, HIPAA, ISO/IEC 27001:2013 untuk kontrol manajemen keamanan, ISO/IEC 27017:2015 untuk kontrol khusus cloud, ISO/IEC 27018:2014 untuk perlindungan data pribadi, ISO/IEC 9001:2015 untuk sistem manajemen mutu, dan lainnya.

Untuk daftar AWS layanan yang berada dalam ruang lingkup untuk program kepatuhan tertentu, lihat [AWS layanan dalam lingkup oleh AWS layanan program kepatuhan](#) . Untuk informasi umum, lihat [program AWS kepatuhan program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga dengan menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh laporan di Laporan AWS Pengunduhan Artefak](#) .

Tanggung jawab kepatuhan Anda saat menggunakan Amazon Pinpoint ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan memulai cepat keamanan dan kepatuhan Panduan memulai](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS
- [Arsitektur untuk whitepaper keamanan dan kepatuhan HIPAA - Whitepaper](#) ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi yang sesuai dengan HIPAA. AWS
- [AWS sumber daya AWS kepatuhan](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Mengevaluasi sumber daya dengan aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Amazon Pinpoint adalah layanan yang memenuhi syarat AWS HIPAA ketika pelanggan menggunakan saluran komunikasi yang tepat. Jika Anda ingin menggunakan Amazon Pinpoint untuk menjalankan beban kerja yang berisi Informasi Kesehatan yang Dilindungi (PHI) sebagaimana didefinisikan oleh HIPAA dan undang-undang dan peraturan terkait, Anda harus menggunakan saluran email, saluran pemberitahuan push, atau saluran SMS untuk mengirim pesan yang berisi PHI. Jika Anda menggunakan saluran SMS untuk mengirim pesan yang berisi PHI, Anda harus mengirim pesan tersebut dari [kode pendek khusus](#) yang Anda minta untuk AWS akun Anda untuk tujuan eksplisit mengirim pesan yang akan atau mungkin berisi PHI. Saluran suara tidak memenuhi syarat AWS HIPAA; jangan gunakan saluran suara untuk mengirim pesan yang berisi PHI.

Ketahanan di Amazon Pinpoint

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya, lihat tentang arsitektur referensi, lihat Panduan Arsitektur [Tangguh Amazon Pinpoint](#).

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [infrastruktur AWS global](#).

Keamanan infrastruktur di Amazon Pinpoint

Sebagai layanan terkelola, Amazon Pinpoint dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon Pinpoint melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Meskipun Anda dapat melakukan panggilan API ini dari lokasi jaringan mana pun, Amazon Pinpoint mendukung kebijakan akses berbasis sumber daya. Kebijakan ini dapat mencakup pembatasan berdasarkan alamat IP sumber. Untuk mempelajari lebih lanjut tentang jenis kebijakan ini, lihat [Mengelola akses menggunakan kebijakan](#).

Selain itu, Anda dapat mengonfigurasi dan menggunakan berbagai fitur AWS keamanan untuk mengontrol akses ke sumber daya Amazon Pinpoint dari aplikasi seluler atau web apa pun yang Anda integrasikan dengan Amazon Pinpoint. Ini termasuk pembatasan panggilan API untuk tugas seperti menambahkan titik akhir, memperbarui data titik akhir, mengirimkan data peristiwa, dan melaporkan data penggunaan.

Untuk menggunakan fitur-fitur ini, sebaiknya gunakan SDK AWS Seluler atau AWS Amplify JavaScript pustaka untuk mengintegrasikan aplikasi seluler dan web dengan Amazon Pinpoint. Untuk aplikasi Android atau iOS, kami sarankan Anda menggunakan AWS Mobile SDK for Android atau AWS Mobile SDK for iOS, masing-masing. Untuk aplikasi seluler atau web JavaScript berbasis, sebaiknya gunakan AWS Amplify JavaScript Library for the Web atau AWS Amplify JavaScript Library for React Native. Untuk mempelajari lebih lanjut tentang sumber daya ini, lihat [Memulai SDK AWS seluler](#), [Memulai library AWS Amplify untuk web](#), dan [Memulai library AWS Amplify](#) untuk react native.

Analisis konfigurasi dan kerentanan di Amazon Pinpoint

Sebagai layanan terkelola, Amazon Pinpoint dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper [Amazon Web Services: Tinjauan proses keamanan](#). Ini berarti AWS mengelola dan melakukan tugas dan prosedur keamanan dasar untuk mengoperasikan, menambal, memperbarui, dan memelihara infrastruktur dasar untuk akun dan sumber daya Amazon Pinpoint Anda. Prosedur ini telah ditinjau dan disertifikasi oleh pihak ketiga yang sesuai.

Untuk informasi selengkapnya, lihat sumber daya berikut:

- [Validasi kepatuhan untuk Amazon Pinpoint](#)
- [Model tanggung jawab bersama](#)
- [Amazon Web Services: Ikhtisar proses keamanan](#) (whitepaper)

Praktik terbaik keamanan untuk Amazon Pinpoint

Gunakan akun AWS Identity and Access Management (IAM) untuk mengontrol akses ke operasi Amazon Pinpoint API, terutama operasi yang membuat, memodifikasi, atau menghapus sumber

daya. Amazon Pinpoint Untuk Amazon Pinpoint API, sumber daya tersebut mencakup proyek, kampanye, dan perjalanan. Untuk Amazon Pinpoint SMS dan Voice API, sumber daya tersebut termasuk nomor telepon, kumpulan, dan set konfigurasi.

- Buat pengguna individu untuk setiap orang yang mengelola Amazon Pinpoint sumber daya, termasuk Anda sendiri. Jangan gunakan kredensial AWS root untuk mengelola sumber daya Amazon Pinpoint.
- Beri setiap pengguna set izin minimum yang diperlukan untuk melakukan tugas-tugasnya.
- Gunakan grup IAM untuk mengelola izin secara efektif bagi beberapa pengguna.
- Putar kredensial IAM Anda secara rutin.

Untuk informasi selengkapnya tentang Amazon Pinpoint keamanan, lihat [Keamanan di Amazon Pinpoint](#). Untuk informasi selengkapnya tentang IAM, lihat [AWS Identity and Access Management](#). Untuk informasi tentang praktik terbaik IAM, lihat [Praktik terbaik IAM](#).

Kuota Amazon Pinpoint

Bagian berikut mencantumkan dan menjelaskan kuota, yang sebelumnya disebut sebagai batas, yang berlaku untuk sumber daya dan operasi Amazon Pinpoint. Beberapa kuota dapat ditingkatkan, sementara yang lain tidak bisa. Untuk menentukan apakah Anda dapat meminta kenaikan kuota, lihat kolom atau pernyataan yang Memenuhi Syarat untuk Meningkatkan di setiap bagian.

Topik

- [Kuota proyek](#)
- [Kuota permintaan API](#)
- [Kuota kampanye](#)
- [Kuota email](#)
- [Kuota titik akhir](#)
- [Kuota impor titik akhir](#)
- [Kuota konsumsi acara](#)
- [Kuota perjalanan](#)
- [Kuota Lambda](#)
- [Kuota pembelajaran mesin](#)
- [Kuota templat pesan](#)
- [Kuota pemberitahuan push](#)
- [Kuota pesan dalam aplikasi](#)
- [Kuota segmen](#)
- [Kuota SMS](#)
- [Kuota 10DLC](#)
- [Kuota suara](#)
- [Meminta peningkatan kuota](#)

Kuota proyek

Tabel berikut mencantumkan kuota yang terkait dengan proyek di Amazon Pinpoint.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Proyek	Di masing-masing Wilayah AWS, Anda dapat memiliki hingga 100 proyek.	Tidak

Kuota permintaan API

Amazon Pinpoint mengimplementasikan kuota yang membatasi ukuran dan jumlah permintaan yang dapat Anda buat ke Amazon Pinpoint API dari akun Anda. AWS

Ukuran maksimum payload pemanggilan (permintaan dan respons) adalah 7 MB, kecuali ditentukan lain untuk jenis sumber daya tertentu. Untuk menentukan apakah sumber daya memiliki kuota yang berbeda, lihat bagian yang sesuai dari topik ini untuk jenis sumber daya tersebut.

Jumlah maksimum permintaan bervariasi menurut jenis kuota dan operasi API. Amazon Pinpoint mengimplementasikan dua jenis kuota untuk permintaan API:

- Kuota tarif — Juga disebut sebagai batas tarif, jenis kuota ini menentukan jumlah maksimum permintaan yang dapat Anda buat per detik untuk operasi tertentu. Ini mengontrol tingkat permintaan yang dikirim atau diterima per akun.
- Kuota burst — Juga disebut sebagai batas burst atau kapasitas burst, jenis kuota ini menentukan jumlah maksimum permintaan yang secara bersamaan dalam penerbangan untuk akun.

Tabel berikut mencantumkan kuota laju dan burst untuk Amazon Pinpoint API.

Operasi	Kuota burst/rate default (Permintaan per detik)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10
CreateImportJob	300

Operasi	Kuota burst/rate default (Permintaan per detik)
CreatePushTemplate	10
CreateSegment	25
CreateSmsTemplate	10
CreateVoiceTemplate	10
DeleteCampaign	25
DeleteEndpoint	5
DeleteSegment	25
GetEndpoint	10
PhoneNumberValidate	20
PutEvents	15
SendMessages	4.000
SendUsersMessages	6.000
UpdateCampaign	25
UpdateEmailTemplate	10
UpdateEndpoint	10
UpdateEndpointsBatch	2
UpdateInAppTemplate	10
UpdatePushTemplate	10
UpdateSegment	25
UpdateSmsTemplate	10

Operasi	Kuota burst/rate default (Permintaan per detik)
UpdateVoiceTemplate	10
Semua operasi lainnya	300

Tabel berikut mencantumkan kuota impor file untuk `CreateImportJob`.

Operasi	Kuota default	Memenuhi syarat untuk kenaikan
Jumlah maksimum file impor	10.000 file per pekerjaan impor	Tidak


Jika Anda melebihi salah satu kuota ini, Amazon Pinpoint membatasi permintaan tersebut—yaitu, ia menolak permintaan yang valid dan mengembalikan kesalahan. `TooManyRequests Throttling` didasarkan pada jumlah total permintaan yang Anda buat dari akun Anda untuk operasi tertentu secara spesifik. Wilayah AWS Selain itu, keputusan pelambatan dihitung secara independen untuk setiap operasi. Misalnya, jika Amazon Pinpoint membatasi permintaan untuk `SendMessage` operasi, permintaan bersamaan untuk operasi dapat diselesaikan dengan sukses. `UpdateEndpoint`

Kuota kampanye

Kuota berikut berlaku untuk sumber daya [Kampanye](#) Amazon Pinpoint API.

Kuota berikut berlaku per Wilayah AWS dan beberapa dapat ditingkatkan. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota dalam Panduan Pengguna Service Quotas](#).

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Kampanye aktif	200 per akun	Tidak

 **Note**
Kampanye aktif adalah kampanye

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
	yang belum selesai atau gagal. Kampanye aktif memiliki statusSCHEDULED ,EXECUTING , atauPENDING_N EXT_RUN .	
Ukuran segmen maksimum	Untuk segmen impor: 100.000.000 per kampanye Untuk segmen dinamis: tidak terbatas	Tidak


Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Kampanye berbasis acara	<p>Setiap proyek dapat menyertakan hingga 25 kampanye yang dikirim saat peristiwa terjadi.</p> <p>Kampanye yang menggunakan pemicu berbasis peristiwa harus menggunakan segmen dinamis. Mereka tidak dapat menggunakan segmen yang diimpor.</p> <p>Jika Anda mengintegrasikan aplikasi dengan Amazon Pinpoint menggunakan SDK AWS Seluler, pesan dari kampanye berbasis acara hanya akan dikirim ke pelanggan yang aplikasinya menjalankan AWS Mobile SDK for Android versi 2.7.2 atau yang lebih baru, atau versi 2.6.30 atau yang lebih baru. AWS Mobile SDK for iOS</p> <p>Jika Amazon Pinpoint tidak dapat mengirimkan pesan dari kampanye berbasis acara dalam waktu lima menit, Amazon Pinpoint akan menghapus pesan dan tidak mencoba mengirimkannya kembali.</p>	Tidak

Kuota email

Kuota di bagian berikut berlaku untuk saluran email.

Kuota pesan email


Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Ukuran maksimum pesan, termasuk lampiran	10 MB per pesan	Tidak
Jumlah identitas terverifikasi	10.000 identitas	Tidak

 **Note**

Identitas mengacu pada alamat email atau domain, atau kombinasi keduanya. Setiap email yang Anda kirim menggunakan Amazon Pinpoint harus dikirim dari identitas terverifikasi.

Kuota pengirim dan penerima email



Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Alamat pengirim	Semua alamat atau domain pengirim harus diverifikasi.	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Alamat penerima	<p>Jika akun Anda ada di kotak pasir, semua alamat email atau domain penerima harus diverifikasi.</p> <p>Jika akun Anda keluar dari kotak pasir, Anda dapat mengirim ke alamat yang valid.</p>	Ya
Jumlah penerima per pesan	50 penerima per pesan	Tidak
Jumlah identitas yang dapat Anda verifikasi	<p>10.000 identitas per Wilayah AWS</p> <div data-bbox="591 926 1029 1528" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Identitas mengacu pada alamat email atau domain, atau kombinasi keduanya. Setiap email yang Anda kirim menggunakan Amazon Pinpoint harus dikirim dari identitas terverifikasi.</p> </div>	Tidak

Kuota pengiriman email

Kuota pengiriman, tarif pengiriman, dan batas kotak pasir dibagi antara dua layanan di Wilayah yang sama. Jika Anda menggunakan Amazon SES di us-east-1, dan Anda telah dihapus dari kotak

pasir dan kuota/tarif pengiriman Anda meningkat, maka semua perubahan itu berlaku untuk akun Pinpoint Anda di us-east-1.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
<p>Jumlah email yang dapat dikirim per periode 24 jam (kuota pengiriman)</p>	<p>Jika akun Anda ada di kotak pasir, 200 email per periode 24 jam.</p> <p>Jika akun Anda berada di luar kotak pasir, kuota bervariasi berdasarkan kasus penggunaan spesifik Anda.</p> <div data-bbox="591 810 1029 1314" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Kuota ini didasarkan pada jumlah penerima, berbeda dengan jumlah pesan unik yang dikirim. Penerima adalah alamat email apa pun di baris Kepada:.</p> </div>	<p>Ya</p>
<p>Jumlah email yang dapat dikirim setiap detik (tingkat pengiriman)</p>	<p>Jika akun Anda ada di kotak pasir, 1 email per detik.</p> <p>Jika akun Anda berada di luar kotak pasir, tarifnya bervariasi berdasarkan kasus penggunaan spesifik Anda.</p> <div data-bbox="591 1703 1029 1885" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Tarif ini didasarkan pada jumlah penerima,</p> </div>	<p>Ya</p>

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
	<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> sebagai lawan dari jumlah pesan unik yang dikirim. Penerima adalah alamat email apa pun di baris Kepada:. </div>	

Kuota titik akhir

Kuota berikut berlaku untuk sumber daya [Endpoints](#) dari Amazon Pinpoint API.

Jumlah maksimum atribut yang didukung per titik akhir adalah 250, dan ukuran titik akhir maksimum adalah 15 KB. Jumlah atribut ini mungkin dibatasi, bagaimanapun, oleh ukuran total titik akhir, yang mencakup semua atribut. Jika Anda mengalami kesalahan saat menambahkan atribut ke template Anda, pertimbangkan untuk mengurangi jumlah data di setiap atribut atau mengurangi jumlah atribut.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Ukuran titik akhir	Ukuran maksimal 15 KB	Tidak
Atribut yang ditetapkan ke <code>Attributes</code> , <code>Metrics</code> , dan <code>UserAttributes</code> parameter secara kolektif	250 untuk semua parameter atribut per aplikasi	Tidak
Atribut yang ditetapkan ke <code>Attributes</code> parameter	250 untuk semua parameter atribut per aplikasi	Tidak
Atribut yang ditetapkan ke <code>Metrics</code> parameter	250 untuk semua parameter atribut per aplikasi	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Atribut yang ditetapkan ke <code>UserAttributes</code> parameter	250 untuk semua parameter atribut per aplikasi	Tidak
Panjang nama atribut	50 karakter	Tidak
Panjang nilai atribut	100 karakter	Tidak
<code>EndpointBatchItem</code> objek dalam <code>EndpointBatchRequest</code> muatan	100 per muatan. Ukuran muatan tidak boleh melebihi 7 MB.	Tidak
Titik akhir dengan ID pengguna yang sama	15 titik akhir unik per ID pengguna	Tidak
Nilai yang ditetapkan untuk atribut <code>Attributes</code> parameter	50 per atribut	Tidak
Nilai yang ditetapkan untuk atribut <code>UserAttributes</code> parameter	50 per atribut	Tidak

Kuota impor titik akhir

Kuota berikut berlaku untuk mengimpor titik akhir ke Amazon Pinpoint.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Lowongan kerja impor aktif	10 per akun Impor pekerjaan hanya dihitung terhadap kuota ini jika sedang berjalan. Setelah	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
	pekerjaan impor selesai, itu tidak lagi dihitung terhadap kuota ini.	
Ukuran impor	1 GB per pekerjaan impor Misalnya, jika setiap titik akhir berukuran 4 KB atau kurang, Anda dapat mengimpor 250.000 titik akhir.	Tidak

Kuota konsumsi acara

Kuota berikut berlaku untuk menelan peristiwa menggunakan SDK AWS Seluler dan sumber daya [Acara](#) dari Amazon Pinpoint API.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum jenis acara kustom	1.500 per aplikasi	Tidak
Jumlah maksimum kunci atribut kustom	500 per aplikasi	Tidak
Jumlah maksimum nilai atribut kustom per tombol atribut	100.000. Jumlah apa pun yang melebihi 100.000 masih terdaftar, tetapi tidak akan tersedia di konsol analitik Amazon Pinpoint.	Tidak
Jumlah karakter maksimum per tombol atribut	50	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah karakter maksimum per nilai atribut	200. Jika jumlah karakter melebihi 200 acara dijatuhkan.	Tidak
Jumlah maksimum tombol metrik kustom	500 per aplikasi	Tidak
Jumlah maksimum kejadian dalam permintaan	100 per permintaan	Tidak
Ukuran permintaan maksimum	4 MB	Tidak
Ukuran maksimum kejadian individu	1.000 KB	Tidak
Jumlah maksimum kunci atribut dan kunci metrik untuk setiap acara	40 per permintaan	Tidak

Kuota perjalanan

Kuota berikut berlaku untuk perjalanan.

Kuota berikut berlaku per Wilayah AWS dan beberapa dapat ditingkatkan. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota dalam Panduan Pengguna Service Quotas](#).

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum perjalanan aktif	50 per akun	Tidak
Jumlah maksimum aktif EventTriggeredJourneys	20 per akun	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum aktivitas perjalanan	40 per perjalanan	Tidak
Ukuran segmen maksimum	Untuk segmen impor: 100.000.000 per perjalanan. Untuk segmen dinamis: tidak terbatas	Tidak
Aktivitas pusat kontak maksimal	3 per perjalanan	Tidak
Jumlah maksimum aturan hari tertutup	20 per saluran	Tidak
Panjang maksimum nama aturan hari tertutup	150 karakter	Tidak
Jumlah hari maksimum antara waktu mulai dan berakhir untuk aturan hari tertutup	7 hari	Tidak
Jumlah maksimum aturan jam buka	4 per hari	Tidak

Kuota Lambda

Kuota berikut berlaku untuk konfigurasi Amazon Pinpoint untuk mengambil dan memproses data dari Lambda

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Ukuran maksimum dari muatan permohonan	6 MB	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
(permintaan dan respon) untuk fungsi Lambda		
Jumlah waktu maksimum untuk menunggu fungsi Lambda memproses data	15 detik	Tidak
Jumlah maksimum atribut peristiwa per titik akhir	5	Tidak
Jumlah maksimum karakter untuk nama atribut acara	128 karakter	Tidak
Jumlah maksimum karakter untuk nilai atribut peristiwa	128 karakter	Tidak
Jumlah maksimum hari perjalanan dapat dijalankan	540 hari	Tidak

Kuota pembelajaran mesin

Kuota berikut berlaku untuk konfigurasi Amazon Pinpoint untuk mengambil dan memproses data dari model machine learning (ML).

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum konfigurasi model	1 per template pesan 100 per akun	Tidak
Jumlah rekomendasi maksimum	5 per titik akhir atau pengguna	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum atribut yang direkomendasikan per titik akhir atau pengguna	1, jika nilai atribut tidak diproses oleh AWS Lambda fungsi 10, jika nilai atribut diproses oleh AWS Lambda fungsi	Tidak
Panjang maksimum nama atribut yang direkomendasikan	50 karakter untuk nama atribut 25 karakter untuk nama tampilan atribut (nama yang muncul di pencari Atribut di konsol)	Tidak
Panjang maksimum nilai atribut yang direkomendasikan yang diambil dari Amazon Personalize	100 karakter	Tidak
Ukuran maksimum dari muatan permohonan (permintaan dan respon) untuk fungsi Lambda	6 MB	Tidak
Jumlah waktu maksimum untuk menunggu fungsi Lambda memproses data	15 detik	Tidak
Jumlah maksimum upaya untuk menjalankan fungsi Lambda	3 upaya	Tidak

Bergantung pada cara Anda mengonfigurasi Amazon Pinpoint untuk menggunakan model ML, kuota tambahan mungkin berlaku. [Untuk mempelajari kuota Amazon Personalisasi, lihat Kuota di Panduan](#)

[Pengembang Amazon Personalisasi](#). Untuk mempelajari AWS Lambda kuota, lihat [Kuota di Panduan AWS Lambda](#) Pengembang.

Kuota templat pesan

Kuota berikut berlaku untuk template pesan untuk akun Amazon Pinpoint Anda.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum template pesan	20.000 per akun	Tidak
Jumlah maksimum versi	5.000 per templat	Tidak
Jumlah karakter maksimum dalam templat email	600.000 karakter	Tidak
Jumlah maksimum karakter dalam templat dalam aplikasi	200.000 karakter	Tidak
Jumlah karakter maksimum di bagian templat default dari templat notifikasi push	4.000 karakter	Tidak
Jumlah maksimum karakter di bagian templat khusus ADM dari templat notifikasi push	6.000 karakter	Tidak
Jumlah maksimum karakter di bagian template khusus APN dari template notifikasi push	4.000 karakter	Tidak
Jumlah maksimum karakter di bagian templat khusus Baidu dari templat notifikasi push	4.000 karakter	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum karakter di bagian templat khusus FCM dari templat notifikasi push	4.000 karakter	Tidak
Jumlah karakter maksimum dalam templat SMS	1.600 karakter	Tidak
Jumlah karakter maksimum dalam templat suara	10.000 karakter	Tidak

Kuota pemberitahuan push

Kuota berikut berlaku untuk pesan yang dikirimkan Amazon Pinpoint melalui saluran notifikasi push.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum notifikasi push yang dapat dikirim per detik dalam kampanye	25.000 notifikasi per detik	Ya
Ukuran payload pesan Amazon Device Messaging (ADM)	6 KB per pesan	Tidak
Ukuran payload pesan layanan Push Notification Apple (APN)	4 KB per pesan	Tidak
Ukuran payload pesan kotak pasir APN	4 KB per pesan	Tidak
Ukuran payload pesan Baidu Cloud Push	4 KB per pesan	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Ukuran payload pesan Firebase Cloud Messaging (FCM)	4 KB per pesan	Tidak

Kuota pesan dalam aplikasi

Kuota berikut berlaku untuk pesan dalam aplikasi yang Anda kelola dengan Amazon Pinpoint.

Kuota berikut berlaku per Wilayah AWS dan beberapa dapat ditingkatkan. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota dalam Panduan Pengguna Service Quotas](#).

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum kali Anda dapat memanggil GetInAppMessages API per detik.	5.000 permintaan per detik	Ya
Kampanye perpesanan dalam aplikasi	Setiap proyek dapat menyertakan hingga 25 kampanye yang menggunakan saluran pesan dalam aplikasi.	Ya, lihat Meminta kenaikan kuota pada Panduan Pengguna Service Quotas

Kuota segmen

Kuota berikut berlaku untuk sumber daya [Segmen](#) dari Amazon Pinpoint API.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum dimensi yang dapat digunakan untuk membuat segmen	100 per segmen	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum grup segmen per segmen	5	Tidak
Jumlah maksimum segmen sumber per segmen	5	Tidak
Kedalaman maksimum segmen sumber. Misalnya jika segmen memiliki segmen sumber yang juga memiliki segmen sumber, rantai kedalaman tidak lebih panjang dari batas ini.	5	Tidak

Kuota SMS

Kuota berikut berlaku untuk saluran SMS.

Untuk informasi selengkapnya tentang biaya SMS, lihat [Memahami laporan penagihan dan penggunaan SMS untuk Amazon Pinpoint](#) di Panduan Pengguna Amazon Pinpoint.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Ambang batas pengeluaran	USD \$1,00 per akun	Ya , tetapi batas pengeluaran bervariasi menurut wilayah. Anda harus menentukan wilayah di mana Anda memerlukan peningkatan.
Jumlah pesan SMS yang dapat dikirim setiap detik (kecepatan pengiriman)	Bervariasi tergantung pada negara tujuan dan nomor telepon asal. Untuk informasi selengkapnya, lihat Batas	Ya , bagaimanapun, Anda mungkin perlu mendapatkan nomor telepon yang mendukung throughput yang

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
	<p>bagian pesan per detik di Panduan Pengguna Amazon Pinpoint.</p>	<p>lebih tinggi. Jika Anda tidak yakin jenis nomor mana yang akan digunakan, hubungi AWS Support atau Manajer AWS Akun Anda untuk informasi lebih lanjut</p> <p>Jika Anda menggunakan ID Pengirim alfanumerik untuk mengirim pesan, Anda mungkin dapat meningkatkan tingkat throughput Anda. Untuk mengetahui apakah peningkatan throughput tersedia untuk ID Pengirim Anda, buka permintaan ID Pengirim di Konsol Pusat Dukungan. Dalam permintaan Anda, sertakan ID Pengirim yang ada, negara tempat Anda menggunakan ID Pengirim tersebut, dan tingkat throughput yang ingin Anda minta.</p>
<p>Jumlah pesan SMS yang dapat dikirim ke satu penerima setiap detik</p>	<p>1 pesan per detik</p>	<p>Tidak</p>
<p>Jumlah topik Amazon SNS untuk SMS dua arah</p>	<p>100.000 per akun</p>	<p>Ya</p>
<p>Jumlah Kata Kunci untuk SMS Dua Arah</p>	<p>30 Kata kunci per angka</p>	<p>Ya</p>

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Nomor SMS dan Nomor Suara	25 per akun dan wilayah	Ya
Jumlah nomor telepon khusus	25 per akun	Ya
Jumlah daftar opt-out Catatan: Daftar opt-out Default yang diperlukan dihitung terhadap kuota ini.	25 per akun	Ya
Jumlah set konfigurasi	25 per akun	Ya
Jumlah tujuan acara	5 per set konfigurasi	Tidak
Jumlah nomor telepon tujuan yang diverifikasi saat berada di kotak pasir SMS	10 per akun	Ya
Jumlah kumpulan nomor telepon	25 per akun	Ya
Jumlah identitas originasi yang dapat dikaitkan dengan kumpulan nomor telepon	100 per kolam nomor telepon	Ya

Kuota 10DLC

Kuota berikut berlaku untuk pesan SMS yang dikirim menggunakan nomor telepon 10DLC. Nomor 10DLC hanya dapat digunakan untuk mengirim pesan ke penerima di Amerika Serikat.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Max 10DLC perusahaan per Akun AWS	25	Ya

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Kampanye Max 10DLC per perusahaan 10DLC	10	Ya
Jumlah maksimum 10DLC per kampanye 10DLC	49	Tidak

Kuota suara


Kuota berikut berlaku untuk saluran suara.

Note

Ketika akun Anda dihapus dari kotak pasir, Anda secara otomatis memenuhi syarat untuk kuota maksimum yang ditunjukkan pada tabel berikut.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah pesan suara yang dapat dikirim selama periode 24 jam	Jika akun Anda ada di kotak pasir: 20 pesan	Tidak
Jumlah pesan suara yang dapat dikirim ke satu penerima selama periode 24 jam	5 pesan	Tidak
Jumlah pesan suara yang dapat dikirim per menit	Jika akun Anda ada di kotak pasir: 5 panggilan per menit Jika akun Anda keluar dari kotak pasir: 20 panggilan per menit	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah pesan suara yang dapat dikirim dari satu nomor telepon asal per detik	1 pesan per detik	Tidak
Panjang pesan suara	Jika akun Anda ada di kotak pasir: 30 detik Jika akun Anda keluar dari kotak pasir: 5 menit	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Kemampuan untuk mengirim pesan suara ke nomor telepon internasional	<p>Jika akun Anda berada di kotak pasir, Anda dapat mengirim pesan ke penerima hanya di negara berikut:</p> <ul style="list-style-type: none">• Australia• Kanada• Germany• Hong Kong• Israel• Jepang• Meksiko• Singapura• Sweden• Amerika Serikat• Britania Raya <p>Jika akun Anda berada di luar kotak pasir, Anda dapat mengirim pesan ke penerima di negara mana pun.</p> <div data-bbox="592 1402 1029 1808" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Note</p><p>Panggilan internasional dikenakan biaya tambahan, yang bervariasi menurut negara atau wilayah tujuan.</p></div>	Tidak

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah karakter dalam pesan suara	3.000 karakter yang dapat ditagih, dalam kata-kata yang diucapkan Total 6.000 karakter, termasuk karakter yang dapat ditagih dan tag SSML	Tidak
Jumlah set konfigurasi	10.000 set konfigurasi suara per AWS Wilayah	Tidak

Meminta peningkatan kuota

Jika nilai di kolom Memenuhi Syarat untuk Peningkatan di salah satu tabel sebelumnya adalah Ya, Anda dapat meminta kenaikan untuk kuota tersebut.

Untuk meminta peningkatan kuota

- Masuk ke AWS Management Console <https://console.aws.amazon.com/>.
- Buat kasus AWS Support baru di <https://console.aws.amazon.com/support/home#/case/create>.
- Pada panel Kasus dukungan Anda, pilih Buat kasus.
- Pilih peningkatan batas Mencari layanan? tautan.
- Dalam peningkatan kuota Layanan, untuk Layanan, pilih salah satu opsi berikut:
 - Untuk meminta peningkatan kuota yang terkait dengan saluran email, pilih Pinpoint Email.
 - Untuk meminta kenaikan kuota batas pengeluaran SMS atau tarif pengiriman SMS, pilih Pinpoint SMS. Untuk semua peningkatan kuota SMS lainnya, pilih Pinpoint
 - Untuk meminta peningkatan kuota yang terkait dengan saluran suara, pilih Pinpoint Voice.
 - Untuk meminta peningkatan kuota yang terkait dengan fitur Amazon Pinpoint lainnya, pilih Pinpoint.
- Tergantung pada Layanan yang Anda pilih, Anda mungkin diminta untuk memasukkan yang berikut:

- (Opsional) Untuk Menyediakan tautan ke situs atau aplikasi yang akan mengirim pesan SMS, memberikan informasi tentang situs web, aplikasi, atau layanan yang akan mengirim pesan SMS.
 - (Opsional) Untuk jenis pesan apa yang ingin Anda kirim, pilih jenis pesan yang akan dikirim menggunakan kode panjang Anda:
 - One Time Password (Kata Sandi Satu Kali) – Pesan yang menyediakan kata sandi yang digunakan pelanggan Anda untuk melakukan autentikasi dengan situs web atau aplikasi Anda.
 - Promotional (Promosi) – Pesan tidak penting yang mempromosikan bisnis atau layanan Anda, seperti penawaran atau pengumuman khusus.
 - Transactional (Transaksional) – Pesan informasi penting yang mendukung transaksi pelanggan, seperti konfirmasi pesanan atau pemberitahuan akun. Pesan transaksional tidak boleh berisi konten promosi atau pemasaran.
 - (Opsional) Untuk AWS Wilayah mana Anda akan mengirim pesan, pilih wilayah tempat Anda akan mengirim pesan.
 - (Opsional) Untuk negara mana Anda berencana untuk mengirim pesan, masukkan negara atau wilayah tempat Anda ingin membeli kode pendek.
 - (Opsional) Dalam Bagaimana pelanggan Anda memilih untuk menerima pesan dari Anda, berikan detail tentang proses keikutsertaan Anda.
 - (Opsional) Di kolom Harap berikan templat pesan yang Anda rencanakan untuk digunakan untuk mengirim pesan ke pelanggan Anda, sertakan templat yang akan Anda gunakan.
7. Di Requests (Permintaan), lakukan hal-hal berikut:
- Untuk Wilayah pilih Anda Wilayah AWS.
 - Untuk Resource Type (Jenis Sumber Daya), pilih General Limits (Batas Umum). Bidang Jenis Sumber Daya hanya ada untuk beberapa Layanan.
 - Untuk Kuota pilih kuota yang akan diubah.
 - Untuk nilai kuota baru masukkan nilai baru untuk kuota.
 - Untuk meminta kenaikan kuota yang sama dalam tambahan Wilayah AWS, pilih Tambahkan permintaan lain, lalu pilih tambahan Wilayah AWS dan isi permintaan baru.
8. Pilih kuota yang ingin Anda tingkatkan, lalu masukkan nilai baru yang Anda inginkan untuk kuota tersebut.
9. Di bawah Deskripsi kasus, jelaskan mengapa Anda meminta kenaikan kuota.

10. Di bawah opsi Kontak, untuk bahasa kontak pilihan, pilih bahasa yang Anda inginkan untuk digunakan saat berkomunikasi dengan tim AWS Support.
11. Untuk metode Kontak, pilih metode komunikasi pilihan Anda dengan tim AWS Support.
12. Pilih Kirim.

Tim AWS Support memberikan tanggapan awal atas permintaan Anda dalam waktu 24 jam.

Untuk mencegah sistem kami digunakan untuk mengirim konten yang tidak diinginkan atau berbahaya, kami harus mempertimbangkan setiap permintaan dengan hati-hati. Jika kami dapat melakukannya, kami akan mengabulkan permintaan Anda dalam periode 24 jam ini. Namun, jika kami perlu mendapatkan informasi tambahan dari Anda, mungkin diperlukan waktu lebih lama untuk menyelesaikan permintaan Anda.

Kami mungkin tidak dapat mengabulkan permintaan Anda jika kasus penggunaan Anda tidak selaras dengan kebijakan kami.

Riwayat dokumen untuk Amazon Pinpoint

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengembang Amazon Pinpoint setelah Desember 2018. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

- Pembaruan dokumentasi terbaru: November 16, 2023

Perubahan	Deskripsi	Tanggal
Header Email	Anda dapat menambahkan header email ke pesan email Anda. Untuk informasi selengkapnya, lihat Mengirim email dengan header berhenti berlangganan .	7 Mei 2024
Orkestrasi Email	Amazon Pinpoint telah memperbarui cara menggunakan sumber daya Amazon SES Anda untuk mengirim email. Untuk informasi selengkapnya, lihat peran IAM untuk mengirim email dengan Amazon SES .	April 30, 2024
Amazon Pinpoint telah memperbarui dokumentasi panduan pengguna mereka	Topik manajemen sumber daya SMS dan Suara sekarang dialihkan ke Panduan Pengguna SMS Amazon Pinpoint. Untuk informasi selengkapnya, lihat Panduan Pengguna SMS Amazon Pinpoint .	Februari 8, 2024
Kuota Amazon Pinpoint	Menambahkan kuota untuk jumlah maksimum aturan hari	Desember 19, 2023

tertutup, Panjang maksimum nama aturan hari tertutup, Jumlah hari maksimum antara waktu mulai dan berakhir untuk aturan hari tertutup dan Jumlah maksimum aturan jam buka. Untuk informasi selengkapnya, lihat [Kuota Amazon Pinpoint](#).

[Amazon Pinpoint telah memperbarui dokumentasi panduan pengguna mereka](#)

Untuk mendapatkan informasi terbaru mengenai cara membuat, mengonfigurasi, dan mengelola sumber daya SMS dan suara Amazon Pinpoint Anda, lihat panduan pengguna SMS [Amazon Pinpoint](#) yang baru.

16 November 2023

[Kuota Amazon Pinpoint](#)

Memperbarui kuota untuk UpdateEndpointsBatch,, UpdateEndpoint, PutEvents DeleteEndpoint, dan GetEndpoint. Untuk informasi selengkapnya, lihat [Kuota Amazon Pinpoint](#).

September 22, 2023

Kuota Amazon Pinpoint	Memperbarui kuota untuk CreateEmailTemplate, CreateSmsTemplate,, CreatePushTemplate, CreateInAppTemplate, CreateVoiceTemplate, UpdateEmailTemplate, UpdateSmsTemplate, UpdatePushTemplate UpdateInAppTemplate, UpdateVoiceTemplate dan CreateImportJob. Untuk informasi selengkapnya, lihat Kuota Amazon Pinpoint .	12 September 2023
Metrik eksekusi perjalanan dan kampanye	Metrik analitik baru telah ditambahkan untuk perjalanan dan kampanye. Untuk informasi selengkapnya, lihat Metrik eksekusi perjalanan dan kampanye .	April 25, 2023
Membuat titik akhir VPC antarmuka untuk Amazon Pinpoint	Amazon Pinpoint sekarang mendukung titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat Membuat titik akhir VPC antarmuka untuk Amazon Pinpoint .	11 April 2023
Enkripsi dalam perjalanan	Mulai 2023-03—22 Amazon Pinpoint tidak akan lagi mendukung TLS 1.0 tetapi Anda masih dapat menggunakan TLS 1.2 atau yang lebih baru. Untuk informasi selengkapnya, lihat Enkripsi bergerak .	20 Maret 2023

Kuota Amazon Pinpoint	Memperbarui proses untuk meminta peningkatan kuota untuk kampanye, perjalanan, dan pesan dalam aplikasi. Untuk informasi selengkapnya, lihat Kuota Amazon Pinpoint .	16 Desember 2022
Ketersediaan regional	Amazon Pinpoint sekarang tersedia di Wilayah ini: Wilayah AS Timur (Ohio).	5 Oktober 2022
Pembaruan contoh peran IAM	Memperbarui beberapa contoh peran IAM di seluruh dokumen agar lebih selaras dengan praktik terbaik keamanan.	Mei 27, 2022
SMS dan API Suara, versi 2	Amazon Pinpoint sekarang menyertakan API khusus untuk mengirim SMS dan pesan Suara. API ini mencakup fitur-fitur baru, seperti set konfigurasi, kumpulan, dan daftar opt-out, yang berguna bagi pelanggan yang mengirim SMS dan pesan suara secara transaksional. Untuk informasi selengkapnya, lihat Menggunakan Amazon Pinpoint SMS dan Voice API .	1 April 2022

Pembuatan dan validasi kata sandi satu kali	Amazon Pinpoint sekarang menyertakan fitur yang menghasilkan kata sandi satu kali (OTP) dan mengirimkannya ke pengguna Anda sebagai pesan SMS. Ini juga mencakup API untuk memvalidasi kode OTP ketika pengguna Anda memasukkannya ke dalam aplikasi atau situs Anda. Untuk informasi selengkapnya, lihat Mengirim dan memvalidasi Kata Sandi Satu Kali (OTP) .	26 November 2021
Pesan dalam aplikasi	Menambahkan informasi tentang mengintegrasikan kemampuan pesan dalam aplikasi Amazon Pinpoint dengan aplikasi Anda.	November 10, 2021
Contoh kode	Menambahkan pustaka contoh kode untuk operasi Amazon Pinpoint umum.	3 November 2021
Kuota proyek	Jumlah maksimum proyek Amazon Pinpoint tetap 100, tetapi kuota ini sekarang dapat ditingkatkan dengan membuka permintaan Peningkatan Batas Layanan dengan. AWS Support	11 Oktober 2021

Pembaruan kebijakan Lambda.	Kebijakan izin Lambda tertentu sekarang harus menyertakan suatu AWS : SourceAccount kondisi. Memperbarui kebijakan sampel di segmen Membuat saluran khusus di Amazon Pinpoint dan Menyesuaikan dengan AWS Lambda topik untuk memenuhi persyaratan ini.	7 Oktober 2021
UpdateEndpoint	Amazon Pinpoint UpdateEndpoint API sekarang dicatat oleh. CloudTrail	16 November 2020
Atribut khusus	Amazon Pinpoint sekarang mendukung 250 atribut dalam template pesan email. Lihat Kuota .	18 September 2020
Ketersediaan regional	Amazon Pinpoint sekarang tersedia di Wilayah ini: Wilayah Asia Pasifik (Tokyo), Wilayah Eropa (London), dan Wilayah Kanada (Tengah). Perhatikan bahwa Amazon Pinpoint SMS dan Voice API tidak tersedia di Wilayah ini.	10 September 2020
Ketersediaan regional	Amazon Pinpoint sekarang tersedia di Wilayah Asia Pasifik (Tokyo). Perhatikan bahwa Amazon Pinpoint SMS dan Voice API tidak mendukung Voice di Wilayah ini.	2 September 2020

Acara kampanye	Menambahkan informasi tentang <code>delivery_type</code> parameter acara kampanye baru ke acara Kampanye .	Agustus 2, 2020
Ketersediaan regional	Amazon Pinpoint sekarang tersedia di Wilayah Asia Pasifik (Seoul). Perhatikan bahwa Amazon Pinpoint API tidak mendukung Voice atau SMS di Wilayah ini.	31 Juli 2020
Ketersediaan regional	Amazon Pinpoint sekarang tersedia di Wilayah. AWS GovCloud (US)	30 April 2020
Saluran khusus	Informasi terbaru tentang membuat saluran kustom dengan menggunakan fungsi Lambda atau webhook .	23 April 2020
Pembelajaran mesin	Menambahkan informasi tentang mengambil rekomendasi yang dipersonalisasi dari model pemberi rekomendasi, dan secara opsional meningkatkan rekomendasi tersebut dengan menggunakan fungsi. AWS Lambda	4 Maret 2020
Keamanan	Menambahkan Bab Keamanan , yang menyediakan informasi tentang berbagai kontrol keamanan dan fitur Amazon Pinpoint.	4 Februari 2020

Perjalanan	Menambahkan informasi tentang penggunaan perjalanan Amazon Pinpoint untuk mengembangkan alur kerja otomatis yang melakukan aktivitas pesan untuk proyek. Juga menambahkan informasi tentang kueri data analitik untuk subset metrik yang berlaku untuk perjalanan.	31 Oktober 2019
Analitik	Menambahkan prosedur yang menjelaskan cara menanyakan data analitik untuk kampanye dan pesan transaksional, serta menambahkan informasi tentang penggunaan hasil kueri .	17 Oktober 2019
Analitik	Menambahkan informasi tentang kueri data analitik untuk subset metrik yang berlaku untuk email transaksional dan pesan SMS.	6 September 2019
Contoh kode	Contoh kode tambahan yang dapat Anda gunakan untuk mengirim pemberitahuan push transaksional menggunakan semua layanan yang didukung Amazon Pinpoint.	30 Juli 2019

Analitik	Menambahkan informasi tentang kueri data analitik untuk subset metrik yang berlaku untuk proyek (aplikasi) dan kampanye.	24 Juli 2019
Segmen	Menambahkan tutorial yang menjelaskan solusi untuk mengimpor data pelanggan ke Amazon Pinpoint dari sistem eksternal, seperti Salesforce atau Marketo.	14 Mei 2019
Ketersediaan regional	Amazon Pinpoint sekarang tersedia di Wilayah AWS Asia Pasifik (Mumbai) dan Asia Pasifik (Sydney).	25 April 2019
Menggunakan tukang pos dengan Amazon Pinpoint	Menambahkan tutorial yang menjelaskan cara menggunakan Postman untuk berinteraksi dengan Amazon Pinpoint API.	8 April 2019
Menandai	Menambahkan informasi tentang menandai sumber daya Amazon Pinpoint .	27 Februari 2019
Pendaftaran SMS	Menambahkan chapter Tutorial , dan menambahkan tutorial yang menjelaskan cara membuat solusi yang menangani pendaftaran pengguna SMS .	27 Februari 2019

[Contoh kode](#)

Menambahkan [contoh kode](#) dalam beberapa bahasa pemrograman yang menunjukkan cara mengirim [email](#), [SMS](#), dan pesan [suara](#) secara terprogram.

6 Februari 2019

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengembang Amazon Pinpoint hingga Desember 2018.

Perubahan	Deskripsi	Tanggal
Ketersediaan wilayah	Amazon Pinpoint sekarang tersedia di Wilayah AWS AS Barat (Oregon) dan Eropa (Frankfurt).	21 Desember 2018
Saluran suara	Anda dapat menggunakan saluran suara Amazon Pinpoint baru untuk membuat pesan suara dan mengirimkannya ke pelanggan Anda melalui telepon. Saat ini, Anda hanya dapat mengirim pesan suara dengan menggunakan Amazon Pinpoint SMS dan Voice API.	15 November 2018
Ketersediaan Eropa (Irlandia)	Amazon Pinpoint sekarang tersedia di Wilayah AWS Eropa (Irlandia).	25 Oktober 2018
API Acara	Gunakan Amazon Pinpoint API untuk merekam peristiwa	7 Agustus 2018

Perubahan	Deskripsi	Tanggal
	dan mengaitkannya dengan titik akhir.	
Contoh kode untuk mendefinisikan dan mencari titik akhir	Contoh kode ditambahkan yang menunjukkan cara mendefinisikan, memperbaiki, menghapus, dan mencari titik akhir. Contoh disediakan untuk AWS CLI, AWS SDK for Java, dan Amazon Pinpoint API. Untuk informasi selengkapnya, lihat Mendefinisikan audiens Anda ke Amazon Pinpoint dan Mengakses data pemirsa di Amazon Pinpoint .	7 Agustus 2018
Izin ekspor titik akhir	Konfigurasi kebijakan IAM yang memungkinkan Anda mengekspor titik akhir Amazon Pinpoint ke bucket Amazon S3.	1 Mei 2018
Verifikasi nomor telepon untuk SMS	Gunakan Amazon Pinpoint API untuk memverifikasi nomor telepon guna menentukan apakah itu adalah tujuan yang valid untuk pesan SMS.	23 April 2018
Topik yang diperbarui untuk integrasi Amazon Pinpoint	Integrasikan Amazon Pinpoint dengan Android, iOS, atau JavaScript aplikasi Anda menggunakan AWS SDK atau pustaka.	Maret 23, 2018

Perubahan	Deskripsi	Tanggal
AWS CloudTrail penebangan	Menambahkan informasi tentang mencatat panggilan Amazon Pinpoint API dengan CloudTrail .	6 Februari 2018
Service Quotas yang diperbarui	Diperbarui Kuota dengan informasi tambahan tentang kuota email.	19 Januari 2018
Beta publik untuk ekstensi Amazon Pinpoint	Gunakan AWS Lambda fungsi untuk menyesuaikan segmen atau membuat saluran pesan khusus .	28 November 2017
Kuota payload notifikasi push	Kuota termasuk ukuran payload untuk pesan push seluler .	25 Oktober 2017
Service Quotas yang diperbarui	Menambahkan informasi saluran SMS dan email ke Kuota .	Oktober 9, 2017
ADM dan dorongan seluler Baidu	Perbarui kode aplikasi Anda untuk menangani pemberitahuan push dari saluran push seluler Baidu dan ADM.	27 September 2017
ID pengguna dan peristiwa otentikasi dengan kumpulan pengguna Amazon Cognito.	Jika Anda menggunakan kumpulan pengguna Amazon Cognito untuk mengelola login pengguna di aplikasi seluler Anda, Amazon Cognito menetapkan ID pengguna ke titik akhir, dan melaporkan peristiwa autentikasi ke Amazon Pinpoint.	26 September 2017

Perubahan	Deskripsi	Tanggal
ID Pengguna	Tetapkan ID pengguna ke titik akhir untuk memantau penggunaan aplikasi dari pengguna individu. Contoh disediakan untuk AWS Mobile SDK dan SDK for Java .	31 Agustus 2017
Acara otentikasi	Laporkan peristiwa autentikasi untuk mempelajari seberapa sering pengguna melakukan autentikasi dengan aplikasi Anda. Contoh disediakan di Melaporkan peristiwa dalam aplikasi Anda .	31 Agustus 2017
Acara sampel yang diperbarui	Contoh peristiwa termasuk peristiwa yang Amazon Pinpoint stream untuk aktivitas email dan SMS.	Juni 08, 2017
Manajemen sesi Android	Kelola sesi di aplikasi Android dengan menggunakan kelas yang disediakan oleh aplikasi AWS Mobile Hub contoh.	20 April 2017
Sampel acara monetisasi yang diperbarui	Kode sampel diperbarui untuk melaporkan peristiwa monetisasi.	31 Maret 2017
Aliran acara	Anda dapat mengonfigurasi Amazon Pinpoint untuk mengirim acara aplikasi dan kampanye ke aliran Kinesis .	24 Maret 2017

Perubahan	Deskripsi	Tanggal
Izin	Lihat Bagaimana Amazon Pinpoint bekerja dengan IAM untuk informasi tentang pemberian akses ke Amazon Pinpoint AWS untuk pengguna di akun Anda dan pengguna aplikasi seluler Anda.	Januari 12, 2017
Amazon Pinpoint ketersediaan umum	Rilis ini memperkenalkan Amazon Pinpoint.	1 Desember 2016

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.