aws

Guida per gli sviluppatori

Amazon Pinpoint

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Pinpoint: Guida per gli sviluppatori

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

..... x
Che cos'è Amazon Pinpoint? 1
Usa Amazon Pinpoint per inviare messaggi ai segmenti di pubblico e analizzare i dati 1
Definizione di segmenti di destinatari 1
Pianifica campagne di messaggistica 2
Invio di messaggi transazionali 2
Utilizza i report di analisi e metriche 2
Usa gli endpoint per definire il tuo pubblico 3
Aggiungi endpoint 4
Esempi 5
Informazioni correlate 10
Associa gli utenti agli endpoint 11
Esempi 12
Informazioni correlate 16
Aggiungi un batch di endpoint 16
Esempi 17
Informazioni correlate 24
Importazione degli endpoint 25
Prima di iniziare 25
Esempi 25
Informazioni correlate 38
Esportazione degli endpoint da Amazon Pinpoint a bucket Amazon S3 38
Prima di iniziare 39
Esempi 39
Informazioni correlate 50
Cerca gli endpoint in un progetto Amazon Pinpoint 50
Esempi 51
Informazioni correlate 56
Elenca l'endpoint IDs 56
Gestisci il numero massimo di endpoint 59
Eliminare gli endpoint 60
Esempi 60
Crea o importa segmenti 64
Crea segmenti 64

Crea segmenti con AWS SDK per Java ... 64
Importa segmenti ... 68
Importa un segmento con AWS SDK per Java ... 68
Personalizza i segmenti ... 71
Dati eventi ... 72
Creazione di una funzione Lambda ... 73
Assegna un criterio di funzione Lambda ... 75
Assegnare una funzione Lambda a una campagna ... 78
Crea campagne Amazon Pinpoint in modo programmatico ... 79
Crea una campagna con l'SDK for Java ... 79
Crea una campagna di A/B prova ... 82
Gestione dei tag ... 85
Usa i tag nelle politiche IAM ... 86
Aggiunge tag a risorse ... 87
Aggiungi tag utilizzando l'API ... 87
Aggiungi tag utilizzando il AWS CLI ... 88
Visualizza i tag per le risorse ... 90
Visualizza i tag utilizzando l'API ... 90
Visualizza i tag utilizzando il AWS CLI ... 91
Aggiorna o sovrascrivi i tag ... 92
Rimuove i tag dalle risorse ... 92
Rimuovi i tag utilizzando l'API ... 92
Rimuovi i tag utilizzando il AWS CLI ... 93
Integra con la tua applicazione ... 94
Lavorare con AWS SDKs ... 94
Connect l'applicazione frontend utilizzando Amplify ... 96
Approfondimenti ... 96
Registra gli endpoint nella tua app ... 96
Prima di iniziare ... 97
AWS mobile SDKs ... 97
AWS Amplify ... 97
Passaggi successivi ... 97
Segnala gli eventi nella tua app ... 97
Prima di iniziare ... 99
AWS cellulare SDKs ... 99
Web e React Native ... 97

API Eventi di Amazon Pinpoint 99
Passaggi successivi 99
Invia messaggi transazionali dalla tua app 101
Invia e-mail transazionali 101
Scegli un metodo per inviare email 102
Scegli tra Amazon Pinpoint e Amazon SES 102
Inviare e-mail utilizzando l'API 102
Aggiungi intestazioni di annullamento dell'iscrizione alle e-mail 117
Invio di messaggi SMS 119
Invio di messaggi vocali 132
Usa l'API SMS and Voice 141
Genera password monouso 143
Risposta `SendOtpMessage` 146
Convalida i messaggi OTP 147
Risposta `VerifyOtpMessage` 147
Esempi di codice OTP in Amazon Pinpoint 148
Genera un ID di riferimento 148
Invia codici OTP 149
Convalida i codici OTP 150
Personalizza i messaggi in-app 152
Recupera i messaggi in-app per un endpoint 152
`GetInAppMessages`Esempio JSON di risposta API 155
Oggetto `InAppMessageCampaigns` 156
Oggetto `InAppMessage` 158
Oggetto `HeaderConfig` 159
Oggetto `BodyConfig` 160
Oggetto `InAppMessageContent` 160
Oggetto `Schedule` 162
Oggetto `InAppMessageButton` 162
Oggetto `DefaultButtonConfig` 163
Oggetto `OverrideButtonConfig` 165
Convalida del numero di telefono 166
Casi d'uso per la convalida dei numeri di telefono di Amazon Pinpoint 166
Convalida un numero di telefono utilizzando il AWS CLI 167
Risposta di convalida del numero di telefono 168
Crea un canale personalizzato 172

Usa un webhook 172
Utilizza una funzione Lambda 172
Assegna una funzione Lambda o un webhook a una singola campagna 174
Crea e configura una funzione Lambda per una campagna Amazon Pinpoint 175
Funzione Lambda di esempio 175
Formato di risposta della funzione Lambda per Amazon Pinpoint 179
Concedi l'autorizzazione a richiamare la funzione 181
Trasmetti i dati degli eventi delle app 184
Configura lo streaming dei dati degli eventi 185
Prerequisiti 185
AWS CLI 186
AWS SDK per Java 186
Flusso di dati sugli eventi delle app da Amazon Pinpoint 187
Esempio di evento dell'app 187
Attributi di eventi dell’app 188
Flusso di dati sugli eventi della campagna da Amazon Pinpoint 193
Esempio di evento della campagna 193
Attributi degli eventi di campagna 194
Dati sugli eventi di viaggio da Amazon Pinpoint 201
Esempio di evento di viaggio 202
Attributi di eventi di viaggio 203
Flusso di dati sugli eventi di posta elettronica da Amazon Pinpoint 207
Esempi di eventi via e-mail 208
Attributi di eventi e-mail 214
Flusso di dati di eventi SMS da Amazon Pinpoint 221
Esempio di evento SMS 222
Attributi di eventi SMS 223
Eliminare un flusso di eventi 233
AWS CLI 233
AWS SDK per Java 233
Interroga i dati di analisi 234
Interrogazione di metriche in Amazon Pinpoint 235
Policy IAM 236
Metriche standard per progetti, campagne e viaggi 240
Parametri delle applicazioni per le campagne 241
Parametri delle applicazioni per la posta elettronica 247

Parametri delle applicazioni per gli SMS ... 259
Parametri della campagna ... 267
Parametri di coinvolgimento del viaggio ... 280
Parametri di esecuzione del viaggio ... 288
Parametri di esecuzione delle attività di viaggio ... 290
Metriche relative all'esecuzione di percorsi e campagne ... 296
Interroga i dati della campagna ... 298
Prerequisiti ... 299
Esegui una query sui dati per una campagna ... 300
Esegui query sui dati per più campagne ... 305
Interroga i dati di messaggistica transazionale ... 312
Prerequisiti ... 312
Interroga i dati per le e-mail transazionali ... 313
Interroga i dati per gli SMS transazionali ... 318
Usa i risultati delle query JSON ... 323
Struttura JSON ... 324
Oggetti e campi JSON ... 330
Registra le chiamate API con CloudTrail ... 332
Informazioni su Amazon Pinpoint in CloudTrail ... 332
Azioni API nei CloudTrail file di registro ... 333
Azioni dell'API di posta elettronica nei CloudTrail file di registro ... 338
Azioni SMS e API vocali v1 supportate nei CloudTrail file di registro ... 339
CloudTrail esempi di immissione di log ... 339
Usa un modello di raccomandazione ... 345
Aggiungi consigli ai messaggi ... 345
Richiama una funzione Lambda per un modello di raccomandazione ... 348
Dati dell’evento di input ... 348
Dati e requisiti di risposta ... 350
Assegna una policy per elaborare i dati di raccomandazione ... 355
Autorizza Amazon Pinpoint a richiamare una funzione Lambda ... 357
Configura un modello di raccomandazione ... 358
Eliminazione dei dati del progetto Amazon Pinpoint ... 359
Elimina tutti i dati del progetto Amazon Pinpoint ... 359
Esempi di codice ... 361
Amazon Pinpoint ... 362
Nozioni di base ... 362

API SMS e Voce di Amazon Pinpoint ... 449
Nozioni di base ... 450
Sicurezza ... 460
Protezione dei dati ... 461
Crittografia dei dati ... 463
Riservatezza del traffico Internet ... 464
Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint ... 464
Gestione dell'identità e degli accessi ... 466
Destinatari ... 466
Autenticazione con identità ... 467
Gestione dell'accesso con policy ... 470
Come Amazon Pinpoint funziona con IAM ... 473
Azioni delle policy Amazon Pinpoint ... 480
Esempi di policy basate su identità ... 511
Ruoli IAM per attività comuni ... 525
Risoluzione dei problemi ... 542
Registrazione di log e monitoraggio ... 544
Convalida della conformità ... 546
Resilienza ... 547
Sicurezza dell'infrastruttura ... 547
Analisi della configurazione e delle vulnerabilità ... 548
Best practice di sicurezza ... 549
Quote ... 550
Quote per i progetti ... 550
Quote di richiesta API ... 551
Quote per le campagne ... 553
Quote per l'e-mail ... 555
Quote per i messaggi e-mail ... 556
Quote per mittenti e destinatari di e-mail ... 556
Quote di invio e-mail ... 557
Quote per gli endpoint ... 559
Quote per l'importazione degli endpoint ... 560
Quote per l'inserimento di eventi ... 561
Quote per i viaggi ... 562
Quote di Lambda ... 563
Quote di machine learning ... 564

Quote per i modelli di messaggio 565
Quote per le notifiche push 567
Quote per i messaggi in-app 567
Quote per i segmenti 568
Quote per SMS 569
Quote 10DLC 569
Quote per i messaggi vocali 569
Richiesta di aumento delle quote 569
Cronologia dei documenti 572
Aggiornamenti precedenti 582

Avviso di fine del supporto: il 30 ottobre 2026, AWS terminerà il supporto per Amazon Pinpoint. Dopo il 30 ottobre 2026, non potrai più accedere alla console Amazon Pinpoint o alle risorse Amazon Pinpoint (endpoint, segmenti, campagne, percorsi e analisi). Per ulteriori informazioni, consulta la pagina relativa alla fine del supporto di Amazon Pinpoint. Nota: per quanto APIs riguarda gli SMS, i comandi vocali, i messaggi push su dispositivi mobili, l'OTP e la convalida del numero di telefono non sono interessati da questa modifica e sono supportati da AWS End User Messaging.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.

Che cos'è Amazon Pinpoint?

Amazon Pinpoint è un AWS servizio che puoi utilizzare per interagire con i tuoi clienti attraverso più canali di messaggistica. Puoi utilizzare Amazon Pinpoint per inviare notifiche push, e-mail, messaggi di testo (SMS) o messaggi vocali.

Le informazioni contenute in questa guida per lo sviluppatore è destinato agli sviluppatori di applicazioni. Questa guida contiene informazioni su come usare le funzionalità di Amazon Pinpoint a livello di programmazione. Contiene anche informazioni di particolare interesse per gli sviluppatori di app per dispositivi mobili, come procedure per l'integrazione di caratteristiche di analisi e messaggistica con l'applicazione.

Amazon Pinpoint è disponibile in diverse AWS regioni in Nord America, Europa, Asia e Oceania. Per ulteriori informazioni su Regioni AWS, consulta Managing Regioni AWS in. Riferimenti generali di Amazon Web Services Per un elenco di tutte le regioni in cui Amazon Pinpoint è attualmente disponibile, consulta Endpoint e quote di Amazon Pinpoint e Endpoint dei servizi AWS nella Riferimenti generali di Amazon Web Services. Per ulteriori informazioni sul numero di zone di disponibilità presenti in ciascuna regione, consulta Infrastruttura globale AWS.

Per ulteriori informazioni su Amazon Pinpoint, consulta le seguenti guide:

- Documentazione di riferimento dell'API Amazon Pinpoint
- API SMS e Voce di Amazon Pinpoint
- Guida per l'utente di Amazon Pinpoint

Usa Amazon Pinpoint per inviare messaggi ai segmenti di pubblico e analizzare i dati

Puoi utilizzare Amazon Pinpoint per definire segmenti di pubblico, inviare campagne di messaggistica e messaggi transazionali e utilizzare metriche per analizzare il comportamento degli utenti.

Definizione di segmenti di destinatari

Per raggiungere gli utenti appropriati per i tuoi messaggi, puoi definire segmenti di destinatari. Un segmento specifica quali utenti riceveranno i messaggi inviati da una campagna. Puoi definire segmenti dinamici in base ai dati segnalati dall'applicazione, ad esempio il sistema operativo o il tipo

di dispositivo mobile. È anche possibile importare segmenti statici definiti utilizzando un altro servizio o un'altra applicazione.

Pianifica campagne di messaggistica

Per coinvolgere i destinatari, puoi creare una campagna di messaggistica. Una campagna prevede l'invio di messaggi personalizzati secondo una pianificazione definita. Puoi creare campagne per inviare messaggi push su dispositivi mobili, e-mail o SMS.

Per sperimentare diverse strategie per le campagne, configura la tua campagna come test A/B e analizza i risultati con gli strumenti di analisi di Amazon Pinpoint.

Invio di messaggi transazionali

Tieni costantemente informati i tuoi clienti inviando messaggi push e SMS transazionali per dispositivi mobili, come messaggi di attivazione di nuovi account, conferme d'ordine e notifiche di reimpostazione della password, direttamente a utenti specifici. È possibile inviare messaggi transazionali mediante la REST API Amazon Pinpoint.

Utilizza i report di analisi e metriche

Acquisisci informazioni approfondite sui destinatari e sull'efficacia delle tue campagne utilizzando gli strumenti di analisi disponibili in Amazon Pinpoint. Puoi visualizzare le tendenze relative al livello di coinvolgimento, alle attività di acquisto, ai dati demografici e molto altro dei tuoi utenti. È anche possibile monitorare il traffico dei messaggi visualizzando i parametri, ad esempio il numero totale di messaggi che sono stati inviati o aperti per una campagna o un'applicazione. Tramite l'API Amazon Pinpoint, l'applicazione è in grado di trasmettere i dati personalizzati resi disponibili da Amazon Pinpoint per l'analisi ed è quindi possibile eseguire query sui dati di analisi in base a determinate metriche standard.

Per analizzare o archiviare dati di analisi al di fuori di Amazon Pinpoint, puoi configurare Amazon Pinpoint per lo streaming dei dati su Amazon Kinesis.

Usa gli endpoint per rappresentare il tuo pubblico in Amazon Pinpoint

In Amazon Pinpoint, ogni membro del pubblico è rappresentato da uno o più endpoint. Quando usi Amazon Pinpoint per inviare un messaggio, lo indirizzi verso endpoint che rappresentano i membri del tuo pubblico di destinazione. Ogni definizione di endpoint include una destinazione del messaggio, come un token del dispositivo, un indirizzo e-mail o un numero di telefono. Include anche dati relativi agli utenti e ai loro dispositivi. Prima di analizzare, segmentare o coinvolgere il pubblico, devi aggiungere endpoint al tuo progetto Amazon Pinpoint.

I dati degli endpoint crescono e cambiano di pari passo con il tuo pubblico. Per visualizzare le informazioni più recenti di Amazon Pinpoint sul tuo pubblico, puoi cercare gli endpoint singolarmente oppure puoi esportare tutti gli endpoint da un progetto Amazon Pinpoint. Esaminando i dati degli endpoint, puoi visualizzare le seguenti informazioni sui tuoi utenti:

- Il loro dispositivo e la loro piattaforma.
- Il loro fuso orario.
- Le versioni della tua app installate sul loro dispositivo.
- La loro ubicazione in città e in campagna.
- Altri attributi e metriche personalizzati che registri.

La console Amazon Pinpoint fornisce anche l'analisi dei dati demografici e degli attributi personalizzati acquisiti negli endpoint.

I seguenti argomenti spiegano come lavorare con gli endpoint in Amazon Pinpoint. Per informazioni sull'aggiunta automatica di endpoint tramite Android, iOS o JavaScript client, consultaRegistra gli endpoint Amazon Pinpoint nella tua applicazione.

Argomenti

- Aggiungi endpoint ad Amazon Pinpoint
- Associa gli utenti agli endpoint Amazon Pinpoint
- Aggiungi un batch di endpoint ad Amazon Pinpoint
- Importazione degli endpoint in Amazon Pinpoint
- Esportazione degli endpoint da Amazon Pinpoint a bucket Amazon S3

- Cerca gli endpoint in un progetto Amazon Pinpoint
- Elenca gli endpoint IDs con Amazon Pinpoint
- Gestisci il numero massimo di endpoint in Amazon Pinpoint
- Eliminazione degli endpoint da Amazon Pinpoint in modo programmatico

Aggiungi endpoint ad Amazon Pinpoint

Un endpoint rappresenta una destinazione alla quale puoi inviare messaggi, ad esempio un dispositivo mobile, un numero di telefono o un indirizzo e-mail. Prima di inviare messaggi a un membro del gruppo di destinatari, è necessario definire uno o più endpoint per quella persona.

Man mano che aggiungi endpoint in Amazon Pinpoint, viene creato un repository sempre più ricco di dati relativi al pubblico. Questi dati sono composti da:

- Endpoint aggiunti da te o aggiornati usando l'API Amazon Pinpoint.
- Endpoint aggiunti o aggiornati dal codice client man mano che si aggiungono utenti alla tua applicazione.

Per definire un endpoint, devi specificare il canale e l'indirizzo. Il canale è il tipo di piattaforma usata per inviare messaggi all'endpoint, ad esempio un servizio di notifica push, SMS o e-mail. L'indirizzo specifica dove inviare il messaggio all'endpoint, ad esempio un token di dispositivo, un numero di telefono o un indirizzo e-mail.

Per aggiungere ulteriori dettagli sui destinatari, puoi arricchire gli endpoint con attributi personalizzati o standard. Questi attributi includono dati sugli utenti, le loro preferenze, i loro dispositivi, le versioni del client che utilizzano o le loro posizioni. L'aggiunta di dati di questo tipo agli endpoint ti consente di:

- Visualizzare grafici relativi al pubblico nella console Amazon Pinpoint.
- Segmentare il gruppo di destinatari in base agli attributi degli endpoint, in modo da inviare messaggi al target appropriato.
- Personalizzare i messaggi incorporando variabili dei messaggi che vengono sostituite con i valori degli attributi degli endpoint.

Un'applicazione mobile o JavaScript client registra automaticamente gli endpoint se integri Amazon Pinpoint utilizzando AWS la libreria Mobile SDKs o Amplify. AWS JavaScript Il client registra un endpoint per ogni nuovo utente e aggiorna gli endpoint per gli utenti già registrati. Per registrare gli

endpoint da un dispositivo mobile o da un client, consulta. JavaScript Registra gli endpoint Amazon Pinpoint nella tua applicazione

Esempi

Gli esempi seguenti illustrano come aggiungere un endpoint a un progetto Amazon Pinpoint. L'endpoint rappresenta un membro del gruppo di destinatari che abita a Seattle e usa un iPhone. Questa persona può ricevere messaggi tramite il servizio Apple Push Notification ()APNs. L'indirizzo dell'endpoint è il token del dispositivo fornito da. APNs

AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l' AWS CLI.

Example Comando per l'aggiornamento di endpoint

Per aggiungere o aggiornare un endpoint, usa il comando update-endpoint:

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

Dove:

- application-id è l'ID del progetto Amazon Pinpoint in cui vuoi aggiungere o aggiornare di un endpoint.
- example-endpoint è l'ID da assegnare a un nuovo endpoint oppure l'ID di un endpoint esistente da aggiornare.
- endpoint-request-file.json è il percorso del file di un file JSON locale che contiene l'input per il parametro. `--endpoint-request`

Example File di richiesta endpoint

Il comando `update-endpoint` di esempio usa un file JSON come argomento per il parametro `--endpoint-request`. Questo file contiene una definizione di endpoint come la seguente:

```
{
  "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
  }
}
```

Per gli attributi che puoi utilizzare per definire un endpoint, consulta EndpointRequestlo schema nell'Amazon Pinpoint API Reference.

AWS SDK per Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK per Java.

Example Codice

Per aggiungere un endpoint, inizializza un oggetto `EndpointRequest` e passalo al metodo `updateEndpoint` del client `AmazonPinpoint`:

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

public class AddExampleEndpoint {

 public static void main(String[] args) {

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

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

  String applicationId = args[0];

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

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

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

  // Adds custom metrics to the endpoint.
  wangXiulansIphoneEndpoint.addMetricsEntry("technology_interest_level", 9.0);
```

```
  wangXiulansIphoneEndpoint.addMetricsEntry("music_interest_level", 6.0);
  wangXiulansIphoneEndpoint.addMetricsEntry("travel_interest_level", 4.0);

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

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

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

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

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

 }
}
```

HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta PUT per l'endpoint

Per aggiungere una richiesta di endpoint, invia una richiesta PUT alla risorsa Endpoint all'URI seguente:

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

Dove:

- application-id è l'ID del progetto Amazon Pinpoint in cui vuoi aggiungere o aggiornare di un endpoint.
- endpoint-id è l'ID che vuoi assegnare a un nuovo endpoint oppure l'ID di un endpoint esistente che vuoi aggiornare.

Nella richiesta, includi le intestazioni richieste e fornisci il codice EndpointRequestJSON come corpo:

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

{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
```

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

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

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

Informazioni correlate

Per ulteriori informazioni sulla risorsa Endpoint nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta Endpoint nella documentazione di riferimento dell'API Amazon Pinpoint.

Per ulteriori informazioni sulla personalizzazione dei messaggi con variabili, consulta la sezione relativa alle variabili dei messaggi nella Guida per l'utente di Amazon Pinpoint.

Per informazioni sulle quote applicabili agli endpoint, ad esempio il numero di attributi che è possibile assegnare, consulta the section called “Quote per gli endpoint”.

Associa gli utenti agli endpoint Amazon Pinpoint

Un endpoint può includere attributi che definiscono un utente, che rappresenta una persona nel gruppo di destinatari. Ad esempio, un utente può rappresentare una persona che ha installato la tua app per dispositivi mobili o che ha un account sul tuo sito Web.

Per definire un utente è necessario specificare un ID utente univoco e, facoltativamente, gli attributi utente personalizzati. Se una persona usa la tua app su più dispositivi o se è contattabile tramite messaggio a più indirizzi, puoi assegnare lo stesso ID utente a più endpoint. In questo caso, Amazon Pinpoint sincronizza gli attributi utente negli endpoint. Quindi, se aggiungi un attributo utente a un endpoint, Amazon Pinpoint aggiunge tale attributo a tutti gli endpoint che condividono lo stesso ID utente.

Puoi aggiungere attributi utente per tenere traccia dei dati applicabili a una persona che non variano in base al dispositivo usato da quella persona. Ad esempio, è possibile aggiungere attributi per il nome, l'età o lo stato dell'account di una persona.

> **Tip**
>
> Se la tua applicazione utilizza i pool di utenti di Amazon Cognito per gestire l'autenticazione degli utenti, Amazon Cognito può aggiungere automaticamente IDs utenti e attributi ai tuoi endpoint. Per il valore dell'ID utente degli endpoint, Amazon Cognito assegna il valore `sub` assegnato all'utente nel pool di utenti. Per ulteriori informazioni sull'aggiunta di utenti con , consulta Utilizzo dell'analisi dei dati di Amazon Pinpoint con i pool di utenti di Amazon Cognito nella Guida per gli sviluppatori di Amazon Cognito.

Dopo aver aggiunto le definizioni degli utenti agli endpoint, puoi segmentare il gruppo di destinatari in altri modi. Puoi definire un segmento in base agli attributi utente oppure puoi definire un segmento importando un elenco di utenti. IDs Quando invii un messaggio a un segmento basato sugli utenti, le potenziali destinazioni includono tutti gli endpoint associati a ogni utente nel segmento.

Puoi anche definire in altre opzioni per l'invio di messaggi ai destinatari. Puoi utilizzare una campagna per inviare messaggi a un segmento di utenti oppure puoi inviare un messaggio direttamente a un elenco di utenti IDs. Per personalizzare il messaggio, puoi includere variabili dei messaggi che vengono sostituite con i valori degli attributi utente.

Esempi

Gli esempi seguenti illustrano come aggiungere una definizione utente a un endpoint.

AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l' AWS CLI.

Example Comando per l'aggiornamento di endpoint

Per aggiungere un utente a un endpoint, usa il comando update-endpoint. Per il parametro `--endpoint-request` è possibile definire un nuovo endpoint, che può includere un utente. In alternativa, per aggiornare un endpoint esistente, è possibile specificare solo gli attributi da modificare. L'esempio seguente aggiunge un utente a un endpoint esistente specificando solo gli attributi utente:

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

Dove:

- *`application-id`*è l'ID del progetto Amazon Pinpoint in cui stai aggiungendo o aggiornando un endpoint.
- *`endpoint-id`*è l'ID che stai assegnando a un nuovo endpoint o è l'ID di un endpoint esistente che stai aggiornando.
- *`endpoint-request-file.json`*è il percorso del file di un file JSON locale che contiene l'input per il parametro. `--endpoint-request`

Example File di richiesta endpoint

Il comando `update-endpoint` di esempio usa un file JSON come argomento per il parametro `--endpoint-request`. Questo file contiene una definizione utente come la seguente:

```
{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":["Wang"],
```

```
            "LastName":["Xiulan"],
            "Gender":["Female"],
            "Age":["39"]
        }
    }
}
```

Per gli attributi che puoi utilizzare per definire un utente, consulta l'`User`oggetto nello EndpointRequestschema nell'Amazon Pinpoint API Reference.

AWS SDK per Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK per Java.

Example Codice

Per aggiungere un utente a un endpoint, inizializza un EndpointRequest oggetto e passalo al `updateEndpoint` metodo del client. `AmazonPinpoint` Puoi usare questo oggetto per definire un nuovo endpoint, che può includere un utente. In alternativa, per aggiornare un endpoint esistente, è possibile aggiornare solo le proprietà da modificare. L'esempio seguente aggiunge un utente a un endpoint esistente aggiungendo un EndpointUser oggetto all'oggetto: EndpointRequest

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

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

```
            wangXiList.add("running");
            wangXiList.add("swimming");

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

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

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

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

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

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

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

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

Per l'esempio completo dell'SDK, consulta AddExampleUser.java on. GitHub

HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta Put per endpoint con definizione utente

Per aggiungere un utente a un endpoint, invia una richiesta PUT alla risorsa Endpoint all'URI seguente:

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

Dove:

- *application-id*è l'ID del progetto Amazon Pinpoint in cui stai aggiungendo o aggiornando un endpoint.
- *endpoint-id*è l'ID che stai assegnando a un nuovo endpoint o è l'ID di un endpoint esistente che stai aggiornando.

Nella richiesta, includi le intestazioni richieste e fornisci il codice EndpointRequestJSON come corpo. Il corpo della richiesta può definire un nuovo endpoint, che può includere un utente. In alternativa, per aggiornare un endpoint esistente, è possibile specificare solo gli attributi da modificare. L'esempio seguente aggiunge un utente a un endpoint esistente specificando solo gli attributi utente:

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

{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":"Wang",
```

```
            "LastName":"Xiulan",
            "Gender":"Female",
            "Age":"39"
        }
    }
}
```

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

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

Informazioni correlate

Per ulteriori informazioni sulla risorsa Endpoint nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta Endpoint nella documentazione di riferimento dell'API Amazon Pinpoint.

Per ulteriori informazioni sulla personalizzazione dei messaggi con variabili, consulta la sezione relativa alle variabili dei messaggi nella Guida per l'utente di Amazon Pinpoint.

Per informazioni su come definire un segmento importando un elenco di utenti IDs, consulta Importazione di segmenti nella Guida per l'utente di Amazon Pinpoint.

Per informazioni sull'invio di un messaggio diretto a un massimo di 100 utenti IDs, consulta Messaggi degli utenti nell'Amazon Pinpoint API Reference.

Per informazioni sulle quote applicabili agli endpoint, ad esempio il numero di attributi utente che è possibile assegnare, consulta the section called “Quote per gli endpoint”.

Aggiungi un batch di endpoint ad Amazon Pinpoint

È possibile aggiungere o aggiornare più endpoint in un'unica operazione fornendo gli endpoint in batch. Ogni richiesta batch può includere fino a 100 definizioni di endpoint.

Se vuoi aggiungere o aggiornare più di 100 endpoint in un'unica operazione, consulta Importazione degli endpoint in Amazon Pinpoint.

Esempi

Gli esempi seguenti illustrano come aggiungere due endpoint contemporaneamente includendoli in una richiesta batch.

AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l' AWS CLI.

Example Comando per l'aggiornamento di endpoint in batch

Per inviare una richiesta di endpoint in batch, usa il comando `update-endpoints-batch`:

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

Dove:

- *`application-id`*è l'ID del progetto Amazon Pinpoint in cui stai aggiungendo o aggiornando gli endpoint.
- *`endpoint_batch_request_file.json`*è il percorso del file di un file JSON locale che contiene l'input per il parametro. `--endpoint-batch-request`

Example File di richiesta di batch di endpoint

Il comando `update-endpoints-batch` di esempio usa un file JSON come argomento per il parametro `--endpoint-request`. Questo file contiene un batch di definizioni di endpoint come il seguente:

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

Per gli attributi che puoi utilizzare per definire un batch di endpoint, consulta EndpointBatchRequestlo schema nell'Amazon Pinpoint API Reference.

AWS SDK per Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK per Java.

Example Codice

Per inviare una richiesta di endpoint in batch, inizializza un oggetto `EndpointBatchRequest` e passalo al metodo `updateEndpointsBatch` del client `AmazonPinpoint`. L'esempio seguente popola un oggetto `EndpointBatchRequest` con due oggetti `EndpointBatchItem`:

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

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

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

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class AddExampleEndpoints {

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

                                Usage:    <appId>

                                Where:
                                   appId - The ID of the application.

                                """;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

                        // Create an EndpointBatchItem object for Mary Major.
                        EndpointBatchItem maryMajorsSmsEndpoint =
EndpointBatchItem.builder()
                                        .channelType(ChannelType.SMS)
                                        .address("+16145550100")
                                        .id("example_endpoint_2")
```

```
                                        .attributes(myMapMary)
                                        .user(maryMajor)
                                        .build();

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

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

                        // Updates the endpoints with Amazon Pinpoint.
                        UpdateEndpointsBatchResponse result =
pinpoint.updateEndpointsBatch(batchRequest);
                        System.out.format("Update endpoints batch result: %s\n",
result.messageBody().message());

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

Per l'esempio completo dell'SDK, consulta AddExampleEndpoints .java on. GitHub

HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta Put per endpoint

Per inviare una richiesta di endpoint in batch, invia una richiesta PUT alla risorsa Endpoints all'URI seguente:

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

*application-id*Dov'è l'ID del progetto Amazon Pinpoint in cui stai aggiungendo o aggiornando gli endpoint.

Nella richiesta, includi le intestazioni richieste e fornisci il codice EndpointBatchRequestJSON come corpo:

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

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

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

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

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

Informazioni correlate

Per ulteriori informazioni sulla risorsa Endpoint nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta Endpoint nella documentazione di riferimento dell'API Amazon Pinpoint.

Importazione degli endpoint in Amazon Pinpoint

È possibile aggiungere o aggiornare numerosi endpoint importandoli da un bucket Amazon S3. L'importazione di endpoint è utile se disponi di record relativi al pubblico all'esterno di Amazon Pinpoint e vuoi aggiungere queste informazioni a un progetto Amazon Pinpoint. In questo caso, è necessario:

1. Creare definizioni di endpoint basate sui dati dei destinatari.
2. Salvare queste definizioni di endpoint in uno o più file e caricare i file in un bucket Amazon S3.
3. Aggiungere gli endpoint a un progetto Amazon Pinpoint importandoli da un bucket.

Ogni processo di importazione può trasferire fino a 1 GB di dati. In un processo tipico, in cui ogni endpoint è da 4 KB o meno, è possibile importare circa 250.000 endpoint. È possibile eseguire fino a due processi di importazione simultanei per account. AWS Se hai bisogno di più larghezza di banda per i tuoi lavori di importazione, puoi inviare una richiesta di aumento della quota di servizio a. Supporto Per ulteriori informazioni, consulta Richiesta di aumento delle quote.

Prima di iniziare

Prima di importare endpoint, devi disporre delle seguenti risorse nel tuo account AWS :

- Un bucket Amazon S3. Per creare un bucket, consulta Creazione di un bucket nella Guida per l'utente di Amazon Simple Storage Service.
- Un ruolo AWS Identity and Access Management (IAM) che concede le autorizzazioni di lettura di Amazon Pinpoint per il tuo bucket Amazon S3. Per creare il ruolo, consulta Ruolo IAM per l'importazione di endpoint o segmenti.

Esempi

Gli esempi seguenti illustrano come aggiungere definizioni di endpoint a un bucket Amazon S3 e quindi importare tali endpoint in un progetto Amazon Pinpoint.

File con definizioni di endpoint

I file che aggiungi al tuo bucket Amazon S3 possono contenere definizioni di endpoint in formato CSV o in formato JSON delimitato da nuove righe. Per gli attributi che puoi utilizzare per definire i tuoi endpoint, consulta lo schema EndpointRequestJSON nell'Amazon Pinpoint API Reference.

CSV

È possibile importare gli endpoint definiti in un file CSV, come nell'esempio seguente:

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

La prima riga è l'intestazione, che contiene gli attributi degli endpoint. Specifica gli attributi nidificati usando la notazione punto, come in `Location.Country`.

Le righe successive definiscono gli endpoint fornendo valori per ognuno degli attributi nell'intestazione.

Per includere una virgola o le virgolette in un valore, racchiudi il valore tra virgolette, come in `"aaa,bbb"`.

Le interruzioni di riga non sono supportate all'interno di un valore nel file CSV.

JSON

È possibile importare gli endpoint definiti in un file JSON delimitato da nuove righe, come nell'esempio seguente:

```
{"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"}}
```

In questo formato, ogni riga è un oggetto JSON completo che contiene una singola definizione di endpoint.

Richieste al processo di importazione

I seguenti esempi illustrano come aggiungere definizioni di endpoint a Amazon S3 caricando un file locale in un bucket. Quindi, gli esempi importano le definizioni di endpoint in un progetto Amazon Pinpoint.

AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l' AWS CLI.

Example Comando S3 CP

Per caricare un file locale in un bucket Amazon S3, utilizza il comando Amazon S3: `cp`

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

Dove:

- ./endpoints-file è il percorso di un file locale che contiene le definizioni di endpoint.
- bucket-name/prefix/ è il nome del bucket Amazon S3 e, facoltativamente, un prefisso che consente di organizzare gli oggetti nel bucket in modo gerarchico. Ad esempio, un prefisso utile potrebbe essere `pinpoint/imports/endpoints/`.

Example Comando per la creazione di un processo di importazione

Per importare le definizioni di endpoint da un bucket Amazon S3, usa il comando `create-import-job`:

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

Dove:

- application-id è l'ID del progetto Amazon Pinpoint per cui stai importando gli endpoint.

- bucket- name/prefix/key è la posizione in Amazon S3 che contiene uno o più oggetti da importare. La posizione può terminare con la chiave per un singolo oggetto oppure con un prefisso che qualifica più oggetti.
- iam-import-role-arnè l'Amazon Resource Name (ARN) di un ruolo IAM che concede ad Amazon Pinpoint l'accesso in lettura al bucket.
- format può essere `JSON` o `CSV`, a seconda del formato utilizzato per definire gli endpoint. Se la posizione Amazon S3 include più oggetti di formati misti, Amazon Pinpoint importa solo gli oggetti che corrispondono al formato specificato.
- RegisterEndpoints può essere `true` o `false`. Se impostato su true, il processo di importazione registra gli endpoint con Amazon Pinpoint quando vengono importate le definizioni di endpoint.

  RegisterEndpoints DefineSegments e combinazioni

RegisterEndpoints	DefineSegments	Descrizione
true	true	Amazon Pinpoint importerà gli endpoint e creerà un segmento che li contiene.
true	false	Amazon Pinpoint importerà gli endpoint e non creerà un segmento.
false	true	Amazon Pinpoint importerà gli endpoint e creerà un segmento che li contiene. Gli endpoint non verranno salvati e non sovrascri veranno gli endpoint esistenti.
false	false	Amazon Pinpoint rifiuterà questa richiesta.

La risposta include i dettagli sul processo di importazione:

```
{
```

```
    "ImportJobResponse": {
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
            "S3Url": "s3://bucket-name/prefix/key"
        },
        "Id": "d5ecad8e417d498389e1d5b9454d4e0c",
        "JobStatus": "CREATED",
        "Type": "IMPORT"
    }
}
```

La risposta fornisce l'ID processo con l'attributo `Id`. Puoi usare questo ID per verificare lo stato corrente del processo di importazione.

Example Comando Get per processo di importazione

Per verificare lo stato corrente di un processo di importazione, usa il comando `get-import-job`:

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

Dove:

- application-id è l'ID del progetto Amazon Pinpoint per cui è stato avviato il processo di importazione.
- job-id è l'ID del processo di importazione da verificare.

La risposta a questo comando fornisce lo stato corrente del processo di importazione:

```
{
    "ImportJobResponse": {
        "ApplicationId": "application-id",
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-24T21:26:45.308Z",
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
```

```
            "DefineSegment": false,
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
            "S3Url": "s3://s3-bucket-name/prefix/endpoint-definitions.json"
        },
        "FailedPieces": 0,
        "Id": "job-id",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 3,
        "Type": "IMPORT"
    }
}
```

La risposta fornisce lo stato del processo con l'attributo `JobStatus`.

AWS SDK per Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK per Java.

Example Codice

Per caricare un file con le definizioni di endpoint in Amazon S3, usa il metodo `putObject` del client `AmazonS3`.

Per importare endpoint in un progetto Amazon Pinpoint, inizializza un oggetto `CreateImportJobRequest`. Passa quindi questo oggetto al metodo `createImportJob` del client `AmazonPinpoint`.

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

import com.amazonaws.AmazonServiceException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.CreateImportJobRequest;
import com.amazonaws.services.pinpoint.model.CreateImportJobResult;
import com.amazonaws.services.pinpoint.model.Format;
import com.amazonaws.services.pinpoint.model.GetImportJobRequest;
```

```
import com.amazonaws.services.pinpoint.model.GetImportJobResult;
import com.amazonaws.services.pinpoint.model.ImportJobRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.List;
import java.util.concurrent.TimeUnit;

public class ImportEndpoints {

    public static void main(String[] args) {

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

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

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

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

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

    }

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

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

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

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

```
   }

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

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

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

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

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

       try {

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

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

           // Checks the job status until the job completes or fails.
           do {
               getImportJobResult = pinpointClient.getImportJob(new
GetImportJobRequest()
                       .withJobId(jobId)
                       .withApplicationId(applicationId));
```

```
                jobStatus =
getImportJobResult.getImportJobResponse().getJobStatus();
                System.out.format("Import job %s . . .\n", jobStatus.toLowerCase());
                TimeUnit.SECONDS.sleep(3);
            } while (!jobStatus.equals("COMPLETED") && !jobStatus.equals("FAILED"));

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

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

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

    }

}
```

HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta PUT Object di S3

Per aggiungere le definizioni di endpoint a un bucket, utilizza l'operazione oggetto PUT di Amazon S3 e fornisci le definizioni di endpoint come corpo:

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

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

Dove:

- /prefix/key è il prefisso e il nome della chiave per l'oggetto che conterrà le definizioni di endpoint dopo il caricamento. È possibile utilizzare il prefisso per organizzare gerarchicamente gli oggetti. Ad esempio, un prefisso utile potrebbe essere `pinpoint/imports/endpoints/`.
- bucket-name è il nome del bucket Amazon S3 a cui aggiungi le definizioni di endpoint.

Example Richiesta POST per processo di importazione

Per importare le definizioni di endpoint da un bucket Amazon S3, invia una richiesta POST alla risorsa Processi di importazione. Nella richiesta, includi le intestazioni richieste e fornisci il codice ImportJobRequestJSON come corpo:

```
POST /v1/apps/application_id/jobs/import HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T214912Z
```

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

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

Dove:

- application-id è l'ID del progetto Amazon Pinpoint per cui stai importando gli endpoint.
- bucket- name/prefix/key è la posizione in Amazon S3 che contiene uno o più oggetti da importare. La posizione può terminare con la chiave per un singolo oggetto oppure con un prefisso che qualifica più oggetti.
- iam-import-role-arnè l'Amazon Resource Name (ARN) di un ruolo IAM che concede ad Amazon Pinpoint l'accesso in lettura al bucket.
- format può essere `JSON` o `CSV`, a seconda del formato utilizzato per definire gli endpoint. Se la posizione Amazon S3 include più file di formati misti, Amazon Pinpoint importa solo i file corrispondenti al formato specificato.

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

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

```
}
```

La risposta fornisce l'ID processo con l'attributo `Id`. Puoi usare questo ID per verificare lo stato corrente del processo di importazione.

Example Richiesta GET per processo di importazione

Per controllare lo stato corrente di un processo di importazione, invia una richiesta `GET` alla risorsa Import Job (Processo di importazione):

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

Dove:

- application_id è l'ID del progetto Amazon Pinpoint per cui è stato avviato il processo di importazione.
- job_id è l'ID del processo di importazione da verificare.

Se la richiesta va a buon fine, riceverai una risposta analoga alla seguente:

```
{
    "ApplicationId": "application_id",
    "Id": "70a51b2cf442447492d2c8e50336a9e8",
    "JobStatus": "COMPLETED",
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-05T22:04:49.213Z",
    "CompletionDate": "2018-06-05T22:04:58.034Z",
    "Type": "IMPORT",
    "TotalFailures": 0,
```

```
    "TotalProcessed": 3,
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key.json",
        "RoleArn": "iam-import-role-arn",
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

La risposta fornisce lo stato del processo con l'attributo `JobStatus`.

Informazioni correlate

Per ulteriori informazioni sulla risorsa Processi di importazione nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta l'articolo relativo ai processi di importazione nella documentazione di riferimento dell'API Amazon Pinpoint.

Esportazione degli endpoint da Amazon Pinpoint a bucket Amazon S3

Per ottenere tutte le informazioni di cui Amazon Pinpoint dispone sul pubblico, è possibile esportare le definizioni di endpoint appartenenti a un progetto. Quando esegui l'esportazione, Amazon Pinpoint inserisce le definizioni di endpoint in un bucket Amazon S3 specificato. L'esportazione degli endpoint è utile per:

- Visualizzare i dati più recenti sugli endpoint nuovi ed esistenti che l'applicazione client ha registrato in Amazon Pinpoint.
- Sincronizzare i dati degli endpoint in Amazon Pinpoint con il sistema CRM (Customer Relationship Management) in uso.
- Creare report sui dati dei clienti o analizzarli.

> **Note**
>
> I contenuti distribuiti ai bucket Amazon S3 potrebbero contenere informazioni dei clienti. Se devi eliminare i dati degli endpoint che hai esportato in un bucket Amazon S3, devi farlo in

Amazon S3 Per ulteriori informazioni sulla rimozione di dati sensibili, consulta Come faccio a svuotare un bucket S3? o Come faccio a eliminare un bucket S3? .

Prima di iniziare

Prima di esportare endpoint, devi disporre delle seguenti risorse nel tuo account AWS :

- Un bucket Amazon S3. Per creare un bucket, consulta Creazione di un bucket nella Guida per l'utente di Amazon Simple Storage Service.
- Un ruolo AWS Identity and Access Management (IAM) che concede le autorizzazioni di scrittura di Amazon Pinpoint per il tuo bucket Amazon S3. Per creare il ruolo, consulta Ruolo IAM per l'esportazione di endpoint o segmenti.

Esempi

Gli esempi seguenti illustrano come esportare gli endpoint da un progetto Amazon Pinpoint e quindi scaricare questi endpoint dal bucket Amazon S3.

AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l' AWS CLI.

Example Comando per la creazione di un processo di esportazione

Per esportare gli endpoint nel tuo progetto Amazon Pinpoint, usa il comando `create-export-job`:

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

Dove:

- *`application-id`* è l'ID del progetto Amazon Pinpoint contenente gli endpoint.

- *bucket-name/prefix/* è il nome del bucket Amazon S3 e, facoltativamente, un prefisso che consente di organizzare gli oggetti nel bucket in modo gerarchico. Ad esempio, un prefisso utile potrebbe essere `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* è il nome della risorsa Amazon (ARN) di un ruolo IAM che concede ad Amazon Pinpoint l'accesso in lettura al bucket.

La risposta a questo comando fornisce dettagli sul processo di esportazione:

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

La risposta fornisce l'ID processo con l'attributo `Id`. Puoi usare questo ID per verificare lo stato corrente del processo di esportazione.

Example Comando per ottenere un processo di esportazione

Per verificare lo stato corrente di un processo di esportazione, usa il comando `get-export-job`:

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

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint da cui sono stati esportati gli endpoint.
- *job-id* è l'ID del processo da verificare.

La risposta a questo comando fornisce lo stato corrente del processo di esportazione:

```
{
```

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

La risposta fornisce lo stato del processo con l'attributo `JobStatus`. Quando il valore di stato del processo è `COMPLETED`, puoi recuperare gli endpoint esportati dal bucket Amazon S3.

Example Comando S3 CP

Per scaricare gli endpoint esportati, usa il comando Amazon S3 cp:

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

Dove:

- *`bucket-name/prefix/key`* è la posizione del file .gz aggiunto da Amazon Pinpoint al bucket durante l'esportazione degli endpoint. Questo file contiene le definizioni degli endpoint esportati. Ad esempio, nell'URL `https://PINPOINT-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/Exports/example.csv`, `PINPOINT-EXAMPLE-BUCKET` è il nome del bucket e `Exports/example.csv` è la chiave. Per ulteriori informazioni sulle Chiavi, consulta Chiavi nella Guida per l'utente di Amazon S3.
- */local/directory/* è il percorso della directory locale in cui scaricare gli endpoint.

AWS SDK per Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK per Java.

Example Codice

Per esportare endpoint da un progetto Amazon Pinpoint, inizializza un oggetto `CreateExportJobRequest`. Passa quindi questo oggetto al metodo `createExportJob` del client `AmazonPinpoint`.

Per scaricare gli endpoint esportati da Amazon Pinpoint, usa il metodo `getObject` del client `AmazonS3`.

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

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

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

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

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

                This program performs the following steps:

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

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

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

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

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

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

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

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

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

```
            String iamExportRoleArn) {

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

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

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

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

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

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

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

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

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

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

            return objectKeys;

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

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

        GetExportJobResponse getExportJobResult;
        String status;

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

```
                    .build();

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

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

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

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

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

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

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

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

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

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

Per l'esempio completo dell'SDK, consulta .java on. ExportEndpoints GitHub

HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta POST al processo di esportazione

Per esportare gli endpoint nel progetto Amazon Pinpoint, invia una richiesta `POST` alla risorsa Processi di esportazione:

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

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

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint contenente gli endpoint.

- *bucket-name/prefix* è il nome del bucket Amazon S3 e, facoltativamente, un prefisso che consente di organizzare gli oggetti nel bucket in modo gerarchico. Ad esempio, un prefisso utile potrebbe essere `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* è il nome della risorsa Amazon (ARN) di un ruolo IAM che concede ad Amazon Pinpoint l'accesso in lettura al bucket.

La risposta a questa richiesta fornisce dettagli sul processo di esportazione:

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

La risposta fornisce l'ID processo con l'attributo `Id`. Puoi usare questo ID per verificare lo stato corrente del processo di esportazione.

Example Richiesta GET al processo di esportazione

Per controllare lo stato corrente di un processo di esportazione, invia una richiesta `GET` alla risorsa Export Job (Processo di esportazione):

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

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint da cui sono stati esportati gli endpoint.

- *job-id* è l'ID del processo da verificare.

La risposta a questa richiesta fornisce lo stato corrente del processo di esportazione:

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

La risposta fornisce lo stato del processo con l'attributo `JobStatus`. Quando il valore di stato del processo è `COMPLETED`, puoi recuperare gli endpoint esportati dal bucket Amazon S3.

Informazioni correlate

Per trovare l'ID dell'endpoint per un endpoint specifico, devi determinare a quale segmento appartiene l'endpoint e quindi esportare il segmento da Amazon Pinpoint. I dati esportati includono l'ID endpoint per ciascun endpoint. È possibile esportare un segmento in un file utilizzando la console Amazon Pinpoint. Per ulteriori informazioni sull'esportazione di segmenti, consulta Esportazione di segmenti nella Guida per l'utente di Amazon Pinpoint.

Per ulteriori informazioni sulla risorsa Processi di esportazione nell'API Amazon Pinpoint, compresi i metodi HTTP supportati e i parametri della richiesta, consulta l'articolo relativo ai processi di esportazione nella documentazione di riferimento dell'API Amazon Pinpoint.

Cerca gli endpoint in un progetto Amazon Pinpoint

È possibile cercare i dettagli relativi a qualsiasi endpoint aggiunto a un progetto Amazon Pinpoint. Questi dettagli possono includere l'indirizzo di destinazione per i messaggi, il canale di

messaggistica, i dati relativi al dispositivo dell'utente, i dati relativi alla posizione dell'utente e qualsiasi attributo personalizzato registrato negli endpoint.

Per cercare un endpoint, è necessario l'ID endpoint. Se non conosci l'ID, puoi recuperare i dati dell'endpoint tramite l'esportazione. Per esportare gli endpoint, consulta the section called "Esportazione degli endpoint da Amazon Pinpoint a bucket Amazon S3".

Esempi

Gli esempi seguenti illustrano come cercare un singolo endpoint specificando il relativo ID.

AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l' AWS CLI.

Example Comando get-endpoint

Per cercare un endpoint, usa il comando `get-endpoint`:

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

Dove:

- *application-id* è l'ID del progetto Amazon Pinpoint contenente l'endpoint.
- *endpoint-id* è l'ID dell'endpoint da cercare.

La risposta a questo comando è la definizione JSON dell'endpoint, come nell'esempio seguente:

```
{
    "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 per Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK per Java.

Example Codice

Per cercare un endpoint, inizializza un oggetto `GetEndpointRequest`. Passa quindi questo oggetto al metodo `getEndpoint` del client `AmazonPinpoint`:

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

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

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

                Usage:   <appId> <endpoint>
```

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

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

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

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

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

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

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

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

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

```
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

Per stampare i dati dell'endpoint in un formato leggibile, questo esempio usa la libreria Google GSON per convertire l'oggetto `EndpointResponse` in una stringa JSON.

HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta GET per endpoint

Per cercare un endpoint, invia una richiesta `GET` alla risorsa 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
```

Dove:

- *`application-id`* è l'ID del progetto Amazon Pinpoint contenente l'endpoint.
- *`endpoint-id`* è l'ID dell'endpoint da cercare.

La risposta a questa richiesta è la definizione JSON dell'endpoint, come nell'esempio seguente:

```
{
    "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"
}
```

Informazioni correlate

Per ulteriori informazioni sulla risorsa Endpoint nell'API Amazon Pinpoint, consulta Endpoint nella documentazione di riferimento dell'API Amazon Pinpoint.

Elenca gli endpoint IDs con Amazon Pinpoint

Per aggiornare o eliminare un endpoint, è necessario l'ID endpoint. Quindi, se desideri eseguire queste operazioni su tutti gli endpoint di un progetto Amazon Pinpoint, il primo passaggio consiste nell'elencare tutti gli IDs endpoint che appartengono a quel progetto. Quindi, puoi modificarli per, ad esempio, aggiungere un attributo IDs a livello globale o eliminare tutti gli endpoint del progetto.

L'esempio seguente utilizza AWS SDK per Java e fa quanto segue:

1. Richiama il `exportEndpointsToS3` metodo di esempio dal codice di esempio in Export endpoints from Amazon Pinpoint. Questo metodo esporta le definizioni di endpoint da un progetto Amazon Pinpoint. Le definizioni di endpoint vengono aggiunte come file gzip in un bucket Amazon S3.
2. Download dei file gzip esportati.
3. Legge i file gzip e ottiene l'ID endpoint della definizione JSON di ogni endpoint.
4. Stampa l'endpoint sulla console IDs .
5. Esegue la pulizia eliminando i file aggiunti da Amazon Pinpoint in 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);
        }
    }
}
```

Per l'esempio completo dell'SDK, consulta ListEndpointIs.java on. GitHub

Gestisci il numero massimo di endpoint in Amazon Pinpoint

Ogni membro del tuo pubblico può avere un massimo di 15 endpoint associati al proprio UserId, vedi. Quote per gli endpoint Se provi ad aggiungere un 16° endpoint allora, a seconda del ChannelType, otterrai BadRequestExceptiono avrai successo rimuovendo l'endpoint con il più vecchio. EffectiveDate

Aggiungi un sedicesimo endpoint

- Se il nuovo tipo di canale per l'endpoint è SMS, PUSH, VOICE, EMAIL, CUSTOM o IN_APP, BadRequestExceptionviene restituito perché il membro del pubblico ha raggiunto il numero massimo di endpoint. Devi rimuovere un endpoint associato al membro del pubblico e riprovare. Consulta Eliminazione degli endpoint da Amazon Pinpoint in modo programmatico.
- Se il nuovo tipo di canale per l'endpoint è ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX o BAIDU:
  - Verifica che almeno un endpoint attualmente associato al membro del pubblico abbia un endpoint tra ADM, GCM, APNS_VOICE, ChannelTypeAPNS_VOICE, APNS_VOIP_SANDBOX o BAIDU. In caso contrario, viene restituito e un endpoint deve essere rimosso prima di riprovare, vedi BadRequestException. Eliminazione degli endpoint da Amazon Pinpoint in modo programmatico
  - Altrimenti, l'endpoint con il più vecchio EffectiveDateviene impostato `INACTIVE` dove si ChannelTypetrova ADM, GCM, APNS_VOIP, APNS_VOIP_SANDBOX o BAIDU.
    - L'endpoint dal vecchio endpoint viene rimosso. UserId
    - Il nuovo endpoint è associato al membro del pubblico che dispone ancora del numero massimo di endpoint.

L'endpoint può essere riabilitato impostando lo Status su `ACTIVE` e UserIdriaggiungendolo all'endpoint.

Eliminazione degli endpoint da Amazon Pinpoint in modo programmatico

Un endpoint rappresenta un metodo per contattare uno dei clienti. Ogni endpoint può fare riferimento all'indirizzo e-mail di un cliente, all'identificatore del dispositivo mobile, al numero di telefono o ad un altro tipo di destinazione a cui è possibile inviare messaggi. In molte giurisdizioni, le informazioni di questo tipo sono considerate personali. Puoi eliminare gli endpoint quando non desideri più inviare messaggi a una determinata destinazione, ad esempio quando la destinazione diventa irraggiungibile o quando un cliente chiude un account.

Esempi

Gli esempi seguenti illustrano come eliminare un endpoint.

AWS CLI

Puoi utilizzare Amazon Pinpoint eseguendo i comandi con l' AWS CLI.

Example Comando di eliminazione endpoint

Per eliminare un endpoint, usa il comando `delete-endpoint`:

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

Dove:

- application-id è l'ID del progetto Amazon Pinpoint contenente l'endpoint.
- endpoint-id è l'ID dell'endpoint da eliminare.

La risposta a questo comando è la definizione JSON dell'endpoint eliminato.

AWS SDK per Java

Puoi utilizzare l'API Amazon Pinpoint nelle applicazioni Java utilizzando il client fornito da AWS SDK per Java.

Example Codice

Per eliminare un endpoint, usa il metodo `deleteEndpoint` del client `AmazonPinpoint`. Specifica un oggetto `DeleteEndpointRequest` come argomento del metodo:

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

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

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

                Usage:   <appName> <endpointId >

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

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

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

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

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

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

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

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

Per l'esempio completo dell'SDK, consulta DeleteEndpoint.java on. GitHub

HTTP

Puoi utilizzare Amazon Pinpoint effettuando richieste HTTP direttamente alla REST API.

Example Richiesta di eliminazione endpoint

Per eliminare un endpoint, invia una richiesta `DELETE` alla risorsa 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
```

Dove:

- application-id è l'ID del progetto Amazon Pinpoint contenente l'endpoint.
- endpoint-id è l'ID dell'endpoint da eliminare.

La risposta a questa richiesta è la definizione JSON dell'endpoint eliminato.

Crea o importa segmenti in Amazon Pinpoint

Un segmento di utenti rappresenta un sottoinsieme di utenti in base a caratteristiche condivise, ad esempio l'ultima volta che gli utenti hanno utilizzato la tua app o la piattaforma di dispositivi che utilizzano. Un segmento indica quali utenti riceveranno i messaggi distribuiti da una campagna. La definizione di segmenti consente di raggiungere i destinatari appropriati quando si vuole invitare gli utenti a utilizzare di nuovo l'app, proporre offerte speciali o aumentare il coinvolgimento e gli acquisti degli utenti.

Dopo aver creato un segmento, è possibile usarlo in una o più campagne. Una campagna invia messaggi mirati agli utenti inclusi nel segmento.

Per ulteriori informazioni, consulta Segmenti.

Argomenti

- Crea segmenti in Amazon Pinpoint
- Importazione di segmenti in Amazon Pinpoint
- Personalizza i segmenti di Amazon Pinpoint utilizzando una funzione AWS Lambda

Crea segmenti in Amazon Pinpoint

Per raggiungere i destinatari previsti per una campagna, puoi creare un segmento basato sui dati riportati dalla tua app. Ad esempio, per raggiungere gli utenti che di recente non hanno utilizzato la tua app, puoi definire un segmento per gli utenti che non hanno utilizzato la tua app negli ultimi 30 giorni.

Per altri esempi di codice, consulta Esempi di codice.

Crea segmenti con AWS SDK per Java

L'esempio seguente mostra come creare un segmento con l' AWS SDK per Java. L'esempio crea un segmento di utenti il cui team è il `Lakers` e che sono stati attivi negli ultimi 30 giorni. Una volta creato il segmento, puoi utilizzarlo come parte di una campagna o di un viaggio. Per un esempio di utilizzo di un segmento con una campagna, consultaCrea campagne Amazon Pinpoint in modo programmatico.

```
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;
        }
}
```

Quando esegui questo esempio, nella finestra della console dell'IDE viene stampato quanto segue:

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

Per l'esempio completo dell'SDK, consulta CreateSegment.java on. GitHub

Importazione di segmenti in Amazon Pinpoint

Con Amazon Pinpoint, puoi definire un segmento di utenti importando le informazioni sugli endpoint appartenenti al segmento. Un endpoint è una destinazione di messaggistica, ad esempio un token di dispositivo mobile push, un numero di telefono o un indirizzo e-mail.

L'importazione di segmenti è utile se hai già creato segmenti di utenti al di fuori di Amazon Pinpoint ma desideri coinvolgere i tuoi utenti nelle campagne di Amazon Pinpoint.

Quando importi un segmento, Amazon Pinpoint recupera gli endpoint del segmento da Amazon Simple Storage Service (Amazon S3). Prima di eseguire l'importazione, devi aggiungere gli endpoint ad Amazon S3 e creare un ruolo IAM che consente ad Amazon Pinpoint di accedere ad Amazon S3. Devi quindi fornire ad Amazon Pinpoint il percorso in Amazon S3 in cui sono archiviati gli endpoint; Amazon Pinpoint aggiungerà ogni endpoint al segmento.

Per creare il ruolo IAM, consulta Ruolo IAM per l'importazione di endpoint o segmenti. Per informazioni sull'importazione di un segmento utilizzando la console Amazon Pinpoint, consulta Importazione di segmenti nella Guida per l'utente di Amazon Pinpoint.

Per altri esempi di codice, consulta Esempi di codice.

Importa un segmento con AWS SDK per Java

L'esempio seguente illustra come importare un segmento utilizzando l' AWS SDK per 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;
    }
}
```

Per l'esempio completo dell'SDK, consulta ImportingSegments.java on. GitHub

Personalizza i segmenti di Amazon Pinpoint utilizzando una funzione AWS Lambda

Questa è la documentazione preliminare per una funzionalità nella versione beta pubblica. ed è soggetta a modifiche.

Puoi usarla AWS Lambda per personalizzare il modo in cui una campagna Amazon Pinpoint coinvolge il tuo pubblico di destinazione. Con AWS Lambda, puoi modificare il segmento della campagna nel momento in cui Amazon Pinpoint invia il messaggio della campagna.

AWS Lambda è un servizio di elaborazione che puoi utilizzare per eseguire codice senza fornire o gestire server. È possibile creare pacchetti di codice e quindi caricarli in Lambda come funzioni Lambda. Lambda esegue una funzione quando la funzione viene richiesta, manualmente o in modo automatico, in risposta a eventi. Per ulteriori informazioni, consulta la Guida per gli sviluppatori di AWS Lambda.

Per assegnare una funzione Lambda a una campagna, definisci le impostazioni `CampaignHook` della campagna utilizzando la risorsa Campagna nell'API Amazon Pinpoint. Tali impostazioni includono il nome della funzione Lambda e la modalità `CampaignHook`, che specifica se Amazon Pinpoint riceve un valore restituito dalla funzione.

Una funzione Lambda assegnata a una campagna viene definita estensione di Amazon Pinpoint.

Dopo aver definito le impostazioni `CampaignHook`, Amazon Pinpoint richiama automaticamente la funzione Lambda quando esegue la campagna, prima di inviare il messaggio della campagna. Quando Amazon Pinpoint richiama la funzione, fornisce dati di evento che si riferiscono al recapito del messaggio. Questi dati includono il segmento della campagna, ovvero l'elenco degli endpoint a cui Amazon Pinpoint invia il messaggio.

Se la modalità `CampaignHook` è impostata su `FILTER`, Amazon Pinpoint consente alla funzione di modificare e restituire il segmento prima di inviare il messaggio. Ad esempio, è possibile che la funzione aggiorni le definizioni di endpoint con attributi contenenti i dati provenienti da un'origine esterna ad Amazon Pinpoint. Oppure potrebbe filtrare il segmento rimuovendo determinati endpoint, in base alle condizioni indicate nel tuo codice di funzione. Dopo che ha ricevuto il segmento modificato dalla funzione, Amazon Pinpoint invia il messaggio a ciascun endpoint del segmento utilizzando il canale di distribuzione della campagna.

Elaborando i segmenti con AWS Lambda, hai un maggiore controllo sui destinatari dei messaggi e sul loro contenuto. È possibile personalizzare le campagne in tempo reale, nel momento in cui vengono inviati i messaggi. L'applicazione di filtri ai segmenti consente di coinvolgere sottoinsiemi di segmenti definiti in modo più specifico. L'aggiunta o l'aggiornamento di attributi di endpoint permette anche di rendere disponibili nuovi dati per le variabili dei messaggi.

> **Note**
>
> È inoltre possibile utilizzare le impostazioni `CampaignHook` per assegnare una funzione Lambda che gestisca il recapito dei messaggi. Questo tipo di funzione è utile per distribuire messaggi tramite canali personalizzati non supportati da Amazon Pinpoint, ad esempio piattaforme di social media. Per ulteriori informazioni, consulta Crea un canale personalizzato in Amazon Pinpoint utilizzando un webhook o una funzione Lambda.
> Quando si richiama un hook Lambda utilizzando Amazon Pinpoint, anche la funzione Lambda deve trovarsi nella stessa regione del progetto Amazon Pinpoint.

Per modificare i segmenti della campagna con AWS Lambda, crea innanzitutto una funzione che elabori i dati degli eventi inviati da Amazon Pinpoint e restituisca un segmento modificato. Quindi, autorizza Amazon Pinpoint a richiamare la funzione assegnando una policy di funzione Lambda. Infine, bisogna assegnare la funzione a una o più campagne definendo le impostazioni `CampaignHook`.

Per altri esempi di codice, consulta Esempi di codice.

Dati eventi

Quando richiama la funzione Lambda, Amazon Pinpoint fornisce il payload seguente come dati degli eventi:

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

```
    . . .
  }
}
```

AWS Lambda passa i dati dell'evento al codice della funzione. I dati sugli eventi forniscono gli attributi seguenti:

- `MessageConfiguration`: ha la stessa struttura dell'oggetto `DirectMessageConfiguration` della risorsa Messages nell'API Amazon Pinpoint.
- `ApplicationId`; ID del progetto Amazon Pinpoint a cui appartiene la campagna.
- `CampaignId`: ID della campagna Amazon Pinpoint per cui viene richiamata la funzione.
- `TreatmentId`: ID di una variante di campagna utilizzata per il testing A/B.
- `ActivityId`: ID dell'attività eseguita dalla campagna.
- `ScheduledTime`: data e ora, in formato ISO 8601, di recapito dei messaggi della campagna.
- `Endpoints`— Una mappa che associa l'endpoint alle definizioni degli endpoint IDs . Ogni payload di dati sugli eventi contiene fino a 50 endpoint. Se il segmento di campagna contiene più di 50 endpoint, Amazon Pinpoint richiama più volte la funzione, con un massimo di 50 endpoint alla volta, finché non vengono elaborati tutti gli endpoint.

Creazione di una funzione Lambda

Per informazioni su come creare una funzione Lambda, consulta Guida introduttiva nella Guida per gli sviluppatori di AWS Lambda . Quando crei una funzione, tieni presente che la consegna del messaggio non riesce quando si verificano le seguenti condizioni:

- La funzione Lambda impiega più di 15 secondi per restituire il segmento modificato.
- Amazon Pinpoint non è in grado di decodificare il valore restituito della funzione.
- Per richiamare la funzione, Amazon Pinpoint deve fare più di 3 tentativi.

Amazon Pinpoint accetta solo definizioni di endpoint nel valore restituito della funzione. La funzione non può modificare altri elementi nei dati di evento.

Funzione Lambda di esempio

La funzione Lambda elabora i dati degli eventi inviati da Amazon Pinpoint e restituisce l'endpoint modificato, come illustrato dal seguente gestore di esempio, scritto in 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 trasferisce i dati degli eventi al gestore come parametro `event`.

In questo esempio il gestore scorre tutti gli endpoint nell'oggetto `event.Endpoints` e aggiunge un nuovo attributo, `CreditScore`, all'endpoint. Il valore dell'attributo `CreditScore` è semplicemente un numero casuale.

L'`console.log()`istruzione registra l'evento in Logs. CloudWatch

L'istruzione `callback()` restituisce gli endpoint modificati ad Amazon Pinpoint. Di solito, il parametro `callback` è facoltativo nelle funzioni Lambda Node.js, ma in questo contesto è obbligatorio perché la funzione deve restituire ad Amazon Pinpoint gli endpoint aggiornati.

La funzione deve restituire gli endpoint nello stesso formato fornito dai dati dell'evento, ovvero una mappa che associa l'endpoint alle definizioni degli endpoint IDs , come nell'esempio seguente:

```
{
    "eqmj8wpxszeqy/b3vch04sn41yw": {
        "ChannelType": "GCM",
        "Address": "4d5e6f1a2b3c4d5e6f7g8h9i0j1a2b3c",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
            "Make": "android"
        },
```

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

La funzione di esempio modifica e restituisce l'oggetto `event.Endpoints` ricevuto nei dati di evento.

Facoltativamente, è possibile includere gli attributi `TitleOverride` e `BodyOverride` nelle definizioni degli endpoint che vengono restituite.

> **Note**
>
> Quando si utilizza questa soluzione per l'invio di messaggi, Amazon Pinpoint implementa gli attributi `TitleOverride` e `BodyOverride` solo per gli endpoint in cui l'attributo `ChannelType` ha uno dei seguenti valori: `ADM`, `APNS`, `APNS_SANDBOX`, `APNS_VOIP`, `APNS_VOIP_SANDBOX`, `BAIDU`, `GCM` o `SMS`.
> Amazon Pinpoint non rispetta questi attributi per gli endpoint in cui il valore dell'attributo `ChannelType` è `EMAIL`.

Assegna un criterio di funzione Lambda

Prima di utilizzare la funzione Lambda per elaborare gli endpoint, è necessario autorizzare Amazon Pinpoint a richiamare la funzione Lambda. Per concedere l'autorizzazione per l'invocazione, assegna una policy di funzione Lambda alla funzione. Una policy di funzione Lambda è una policy di autorizzazione basata su risorse che designa le entità che possono utilizzare la funzione e le azioni che tali entità possono eseguire.

Per ulteriori informazioni, consulta Uso delle policy basate sulle risorse per AWS Lambda nella Guida per gli sviluppatori di AWS Lambda .

Policy di funzione di esempio

La seguente politica concede l'autorizzazione al responsabile del servizio Amazon Pinpoint di utilizzare `lambda:InvokeFunction` l'azione per una campagna *`campaign-id`* specifica ():

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

La policy di funzione richiede un blocco `Condition` che includa una chiave `AWS:SourceArn`. Questo codice indica la campagna Amazon Pinpoint a cui è consentito richiamare la funzione. In questo esempio, la policy concede l'autorizzazione per una sola campagna. Il `Condition` blocco deve includere anche una `AWS:SourceAccount` chiave, che controlla quale AWS account può richiamare l'azione.

Per scrivere una policy più generica, utilizza un caratteri jolly di corrispondenza con più caratteri (*). Ad esempio, puoi utilizzare il seguente `Condition` blocco per consentire a qualsiasi campagna in uno specifico progetto Amazon Pinpoint (*`application-id`*) di richiamare la funzione:

```
...
"Condition": {
  "StringEquals": {
    "AWS:SourceAccount": "111122223333"
```

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

Se vuoi che la funzione Lambda sia la funzione predefinita utilizzata da tutte le campagne per un progetto, è consigliabile configurare il blocco `Condition` per la policy nel modo precedente. Per informazioni sull'impostazione di una funzione Lambda come predefinita per tutte le campagne in un progetto, consulta Assegnare una funzione Lambda a una campagna.

Concedi l'autorizzazione di chiamata di Amazon Pinpoint

Puoi usare il AWS Command Line Interface (AWS CLI) per aggiungere autorizzazioni alla politica della funzione Lambda assegnata alla tua funzione Lambda. Per consentire ad Amazon Pinpoint di richiamare una funzione per una determinata campagna, usa il comando Lambda `add-permission`, come mostrato nell'esempio seguente:

```
$ 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
```

Puoi cercare la tua campagna IDs utilizzando il comando get-campaigns in. AWS CLI Puoi anche cercare l'ID dell'applicazione utilizzando il comando get-apps.

Quando esegui il comando Lambda `add-permission`, viene restituito l'output seguente:

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

Il valore di `Statement` è una versione in formato stringa JSON dell'istruzione aggiunta alla policy della funzione Lambda.

Assegnare una funzione Lambda a una campagna

È possibile assegnare una funzione Lambda a una campagna Amazon Pinpoint specifica. In alternativa, puoi impostare la funzione Lambda come predefinita per tutte le campagne in un progetto, tranne le campagne a cui assegni una funzione individualmente.

Per assegnare una funzione Lambda a una campagna specifica, utilizza l'API Amazon Pinpoint per creare o aggiornare un oggetto `Campaign` e definire il relativo attributo `CampaignHook`. Per impostare una funzione Lambda come predefinita per tutte le campagne in un progetto, crea o aggiorna la risorsa `Settings` per il progetto in questione e definisci il relativo oggetto `CampaignHook`.

In entrambi i casi, imposta gli attributi `CampaignHook` seguenti:

- `LambdaFunctionName`: nome o nome della risorsa Amazon (ARN) della funzione Lambda richiamata da Amazon Pinpoint prima di inviare i messaggi per la campagna.
- `Mode`: impostato su `FILTER`. Con questa modalità, Amazon Pinpoint richiama la funzione e attende che restituisca gli endpoint modificati. Dopo averli ricevuti, Amazon Pinpoint invia il messaggio. Amazon Pinpoint attende un massimo di 15 secondi prima che il recapito dei messaggi venga impostato come non riuscito.

Con le impostazioni `CampaignHook` definite per una campagna, Amazon Pinpoint richiama la funzione Lambda specificata prima di inviare i messaggi della campagna. Amazon Pinpoint attende di ricevere gli endpoint modificati dalla funzione. Se riceve gli endpoint aggiornati, Amazon Pinpoint procede con il recapito dei messaggi utilizzando i dati aggiornati.

Crea campagne Amazon Pinpoint in modo programmatico

Per aumentare il coinvolgimento tra le app e i relativi utenti, con Amazon Pinpoint puoi creare e gestire campagne tramite notifiche push per raggiungere specifici segmenti di utenti.

Ad esempio, una campagna può promuovere l'uso dell'app presso gli utenti che non l'hanno eseguita di recente oppure offrire promozioni speciali agli utenti che non effettuano acquisti da tempo.

Una campagna invia un messaggio mirato a un segmento di utenti specificato. La campagna può inviare il messaggio a tutti gli utenti nel segmento. In alternativa, puoi allocare un'esclusione, ovvero una percentuale di utenti che non ricevono i messaggi.

È possibile pianificare la campagna in modo che il messaggio venga inviato una sola volta o con una determinata frequenza, ad esempio una volta a settimana. Per evitare che gli utenti ricevano il messaggio in orari inopportuni, la pianificazione può includere un orario di silenzio in cui non vengono inviati messaggi.

Per sperimentare strategie di campagna alternative, configura la tua campagna in modo che un A/B test. An A/B test includa due o più trattamenti del messaggio o della pianificazione. Le varianti sono versioni diverse del messaggio o della pianificazione. Man mano che gli utenti rispondono alla campagna, puoi osservare i dati analitici della campagna per confrontare l'efficacia di ogni variante.

Per ulteriori informazioni, consulta Campaigns nella Amazon Pinpoint REST API Guide o Campaigns in Amazon Pinpoint User Guide.

Crea una campagna Amazon Pinpoint standard

Una campagna standard invia una notifica push personalizzata a un determinato segmento in base a una pianificazione definita in precedenza. L'esempio seguente mostra come creare una campagna con l' AWS SDK per Java. Per un esempio di creazione di un segmento da trasmettere, consultaCrea segmenti in Amazon Pinpoint.

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

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

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

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

        final String usage = """

                Usage:   <appId> <segmentId>

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

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

        String appId = args[0];
```

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

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

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

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

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

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

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

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

            CreateCampaignResponse result =
client.createCampaign(CreateCampaignRequest.builder()
```

```
                    .applicationId(appID)
                    .writeCampaignRequest(request).build());

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

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

        return null;
    }
}
```

Quando esegui questo esempio, nella finestra della console dell'IDE viene stampato quanto segue:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Per l'esempio completo dell'SDK, consulta CreateCampaign.java on. GitHub

Crea una campagna Amazon Pinpoint di A/B prova con AWS SDK per Java

Una campagna di A/B test si comporta come una campagna standard, ma consente di definire trattamenti diversi per il messaggio o la pianificazione della campagna. Un test A/B include due o più varianti per il messaggio o la pianificazione. Le varianti sono versioni diverse del messaggio o della pianificazione. Man mano che gli utenti rispondono alla campagna, puoi osservare i dati analitici della campagna per confrontare l'efficacia di ogni variante.

L'esempio seguente mostra come creare una campagna di A/B prova con. AWS SDK per 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();
    }

}
```

Quando esegui questo esempio, nella finestra della console dell'IDE viene stampato quanto segue:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Gestione dei tag delle risorse di Amazon Pinpoint

Un tag è un'etichetta che puoi definire e associare facoltativamente a AWS risorse, inclusi determinati tipi di risorse Amazon Pinpoint. I tag consentono di categorizzare e gestire le risorse in diversi modi, ad esempio in base allo scopo, al proprietario, all'ambiente o ad altri criteri. Ad esempio, è possibile utilizzare i tag per applicare le policy o l'automazione, nonché per identificare le risorse che sono soggette a determinati requisiti di conformità. Puoi aggiungere tag ai seguenti tipi di risorse Amazon Pinpoint:

- Campagne
- Modelli dei messaggi
- Progetti (applicazioni)
- Segmenti

Una risorsa può avere fino a 50 tag. Ogni tag è composto da una chiave di tag obbligatoria e da un valore di tag opzionale, entrambi definibili dall'utente. Una chiave di tag è un'etichetta generale che funge da categoria per più valori di tag specifici. Un valore di tag funge da descrittore di una chiave di tag.

Una chiave di tag può contenere fino a un massimo di 128 caratteri. Un valore di tag può contenere fino a un massimo di 256 caratteri. I caratteri possono essere lettere Unicode, numeri, spazi o uno dei seguenti simboli: _ . : / = + -. Ai tag si applicano le seguenti ulteriori limitazioni:

- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Per ogni risorsa associata, ciascuna chiave di tag deve essere univoca e avere un solo valore.
- Il `aws:` prefisso è riservato all'uso di AWS; non puoi utilizzarlo in nessuna chiave o valore di tag che definisci. Inoltre, non è possibile modificare o rimuovere chiavi o valori di tag che utilizzano questo prefisso. I tag che utilizzano questo prefisso non vengono conteggiati per la quota di 50 tag per ogni risorsa.
- Non è possibile aggiornare o eliminare una risorsa solo in base ai suoi tag. Inoltre, è necessario specificare l'ARN (Amazon Resource Name) o l'ID della risorsa, a seconda dell'operazione utilizzata.
- È possibile associare i tag a risorse pubbliche o condivise. Tuttavia, i tag sono disponibili solo per il tuo AWS account, non per altri account che condividono la risorsa. Inoltre, i tag sono disponibili solo per le risorse che si trovano nella AWS regione specificata per il tuo AWS account.

Per aggiungere, visualizzare, aggiornare e rimuovere chiavi e valori di tag dalle risorse Amazon Pinpoint, puoi utilizzare AWS Command Line Interface (AWS CLI), l'API Amazon Pinpoint, l'API Tagging o un SDK. AWS Resource Groups AWS Per gestire le chiavi e i valori dei tag in tutte le AWS risorse che si trovano in una AWS regione specifica per il tuo AWS account (incluse le risorse Amazon Pinpoint), utilizza l'API AWS Resource Groups Tagging.

Per ulteriori informazioni sui comandi CLI che è possibile utilizzare per gestire le risorse Amazon Pinpoint, consulta la sezione relativa ad Amazon Pinpoint della documentazione di riferimento dei comandi della AWS CLI.

Per ulteriori informazioni sulle risorse nell'API Amazon Pinpoint, compresi i metodi HTTP(S), i parametri e gli schemi supportati, consulta la documentazione di riferimento dell'API Amazon Pinpoint.

Usa i tag Amazon Pinpoint nelle politiche IAM e nelle operazioni API

Dopo aver iniziato a implementare i tag, puoi applicare autorizzazioni basate su tag a livello di risorsa alle policy AWS Identity and Access Management (IAM) e alle operazioni API. Ciò include le operazioni che supportano l'aggiunta di tag alle risorse al momento della loro creazione. Utilizzando i tag in questo modo, puoi implementare un controllo granulare su quali gruppi e utenti del tuo AWS account sono autorizzati a creare e taggare risorse e quali gruppi e utenti sono autorizzati a creare, aggiornare e rimuovere tag in generale.

Ad esempio, è possibile creare una policy che consenta a un utente di avere accesso completo a tutte le risorse Amazon Pinpoint il cui nome è un valore nel tag `Owner` della risorsa:

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

```
      }
   ]
}
```

Se vengono definite autorizzazioni a livello di risorsa basate su tag, le autorizzazioni diventano subito effettive. Ciò significa che le risorse sono più sicure non appena vengono create e che è possibile avviare rapidamente l'applicazione di tag alle nuove risorse. È inoltre possibile utilizzare le autorizzazioni a livello di risorsa per controllare quali chiavi e valori di tag possono essere associati a risorse nuove ed esistenti. Per ulteriori informazioni, consulta Controllo dell'accesso alle risorse AWS mediante i tag nella Guida per l'utente di AWS .

Aggiungi tag alle risorse di Amazon Pinpoint in modo programmatico

Nei seguenti esempi viene illustrato come aggiungere un tag a una risorsa Amazon Pinpoint utilizzando la AWS CLI e la REST API Amazon Pinpoint. Puoi anche utilizzare qualsiasi AWS SDK supportato per aggiungere un tag a una risorsa.

Per aggiungere un tag a più risorse Amazon Pinpoint in un'unica operazione, utilizza le operazioni di etichettatura dei gruppi di risorse AWS CLI o l'AWS Resource Groups API di tagging.

Aggiungi tag utilizzando l'API

Per creare una nuova risorsa e aggiungere un tag utilizzando la REST API Amazon Pinpoint, invia una richiesta POST all'URI di risorsa appropriato. Includi il parametro `tags` e i valori nel corpo della richiesta. L'esempio seguente mostra come specificare un tag quando si crea un nuovo progetto.

```
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"
   }
}
```

Per aggiungere un tag a una risorsa esistente, invia una richiesta POST all'URI Tag. Includi l'Amazon Resource Name (ARN) della risorsa nell'URI. L'ARN deve essere codificato in formato URL. Nel corpo della richiesta, includi il parametro `tags` e i valori, come mostrato nel seguente esempio.

```
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"
   }
}
```

Aggiungi tag utilizzando il AWS CLI

Per creare una nuova risorsa e aggiungervi un tag utilizzando AWS CLI, utilizzate il `create` comando appropriato per la risorsa. Includi il parametro `tags` e i valori. L'esempio seguente mostra come specificare i tag quando si crea un nuovo progetto.

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}
```

Nell'esempio precedente, procedere nel modo seguente:

- Sostituiscila *MyProject* con il nome che vuoi dare al progetto.
- Sostituisci *key1* e *key2* con le chiavi dei tag che vuoi aggiungere alla risorsa.
- Sostituisci *value1* e *value2* con i valori dei tag che desideri aggiungere per le rispettive chiavi.

Per informazioni sui comandi utilizzabili per creare una risorsa Amazon Pinpoint, consulta la documentazione di riferimento dei comandi della AWS CLI.

Per aggiungere un tag a una risorsa esistente, utilizzare il comando `tag-resource` e specificare i valori appropriati per i parametri richiesti:

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}
```

Nell'esempio precedente, procedere nel modo seguente:

- Sostituisci *resource-arn* con l'Amazon Resource Name (ARN) della risorsa a cui desideri aggiungere un tag.
- Sostituisci *key1* e *key2* con le chiavi dei tag che desideri aggiungere alla risorsa.
- Sostituisci *value1* e *value2* con i valori dei tag che desideri aggiungere per le rispettive chiavi.

Visualizza i tag per le risorse Amazon Pinpoint in modo programmatico

I seguenti esempi mostrano come utilizzare la AWS CLI e la REST API Amazon Pinpoint per visualizzare l'elenco di tutti i tag (chiavi e valori) associati a una risorsa Amazon Pinpoint. Puoi anche utilizzare qualsiasi AWS SDK supportato per visualizzare i tag associati a una risorsa.

Visualizza i tag utilizzando l'API

Per utilizzare la REST API Amazon Pinpoint per visualizzare tutti i tag associati a una risorsa specifica, inviare una richiesta GET all'URI Tags e includere il nome della risorsa Amazon (ARN) della risorsa nell'URI. L'ARN deve essere codificato in formato URL. Ad esempio, la seguente richiesta recupera tutti i tag associati a una campagna specificata (*resource-arn*):

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

La risposta JSON alla richiesta include un oggetto `tags`. L'oggetto `tags` elenca tutte le chiavi e i valori di tag associati alla campagna.

Per visualizzare tutti i tag associati a più di una risorsa dello stesso tipo, inviare una richiesta GET all'URI appropriato per quel tipo di risorsa. Ad esempio, la richiesta seguente recupera informazioni su tutte le campagne del progetto specificato ()*application-id*:

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

La risposta JSON alla richiesta elenca tutte le campagne nel progetto. L'oggetto `tags` di ciascuna campagna elenca tutte le chiavi e i valori di tag associati alla campagna.

Visualizza i tag utilizzando il AWS CLI

Per utilizzare il AWS CLI per visualizzare un elenco dei tag associati a una risorsa specifica, esegui il `list-tags-for-resource` comando e specifica l'Amazon Resource Name (ARN) della risorsa per il `resource-arn` parametro, come mostrato nell'esempio seguente.

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
```

Per visualizzare un elenco di tutte le risorse Amazon Pinpoint con tag e tutti i tag associati a ciascuna di tali risorse, utilizza il comando get-resources dell'API Tagging. AWS Resource Groups Imposta il parametro `resource-type-filters` su `mobiletargeting`, come mostrato nell'esempio seguente.

Linux, macOS, or Unix

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

Windows Command Prompt

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

L'output del comando è un elenco di tutte ARNs le risorse Amazon Pinpoint con tag. L'elenco include tutte le chiavi e valori di tag che sono associati a ogni risorsa.

Aggiorna o sovrascrivi i tag per le risorse Amazon Pinpoint a livello di codice

Sono disponibili diversi modi per aggiornare (sovrascrivere) un tag di una risorsa Amazon Pinpoint. Il modo migliore per aggiornare un tag dipende da:

- Il tipo di risorsa di cui si desidera aggiornare i tag.
- Se si desidera aggiornare un tag per una o più risorse contemporaneamente.
- Se si desidera aggiornare una chiave di tag, un valore di tag o entrambi.

Per aggiornare un tag per un progetto Amazon Pinpoint o per più risorse contemporaneamente, utilizza le operazioni di tagging dei gruppi di risorse dell' AWS CLI API Tagging.AWS Resource Groups L'API Amazon Pinpoint attualmente non fornisce supporto diretto per nessuna di queste attività.

Per aggiornare un tag per una risorsa, è possibile rimuovere il tag attuale e aggiungere un nuovo tag tramite l'API Amazon Pinpoint.

Rimuovi i tag dalle risorse di Amazon Pinpoint in modo programmatico

Negli esempi seguenti viene illustrato come rimuovere un tag (sia la chiave che il valore) da una risorsa Amazon Pinpoint utilizzando la AWS CLI e la REST API Amazon Pinpoint. Puoi anche utilizzare qualsiasi AWS SDK supportato per rimuovere un tag da una risorsa.

Per rimuovere un tag da più risorse Amazon Pinpoint in un'unica operazione, utilizza le operazioni di etichettatura dei gruppi di risorse AWS CLI o l'AWS Resource Groups API di tagging. Per rimuovere da una risorsa solo un determinato valore di tag (non la chiave di tag), aggiornare i tag per la risorsa.

Rimuovi i tag utilizzando l'API

Per rimuovere un tag da una risorsa mediante la REST API Amazon Pinpoint, invia una richiesta DELETE all'URI Tags. Nell'URI, includere l'Amazon Resource Name (ARN) della risorsa da cui si desidera rimuovere un tag, seguita dal parametro `tagKeys` e dal tag da rimuovere. Per esempio:

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

Dove:

- *endpoint*è l'endpoint Amazon Pinpoint per la AWS regione che ospita la risorsa.
- *resource-arn*è l'ARN della risorsa da cui si desidera rimuovere un tag.
- *key*è il tag che si desidera rimuovere dalla risorsa.

Tutti i parametri devono essere codificati URL.

Per rimuovere da una risorsa più chiavi di tag e i relativi valori associati, aggiungere il parametro e l'argomento `tagKeys` per ogni ulteriore tag da rimuovere, separati da una e commerciale (&). Per esempio:

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

Tutti i parametri devono essere codificati URL.

Rimuovi i tag utilizzando il AWS CLI

Per rimuovere un tag da una risorsa utilizzando il AWS CLI, esegui il `untag-resource` comando. Includi il parametro `tag-keys`, come mostrato nell'esempio seguente.

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
```

In questo esempio, apporta le modifiche seguenti:

- Sostituisci *resource-arn* con l'ARN della risorsa da cui desideri rimuovere i tag.
- Sostituisci *key1* e *key2* con le chiavi dei tag che desideri rimuovere dalla risorsa.

Integra Amazon Pinpoint con la tua applicazione

L'integrazione di Amazon Pinpoint con il codice client è utile per comprendere e coinvolgere gli utenti.

Dopo l'integrazione e l'avvio dell'applicazione da parte degli utenti, quest'ultima si connette al servizio Amazon Pinpoint per aggiungere o aggiornare gli endpoint. Gli endpoint rappresentano le destinazioni a cui è possibile inviare messaggi, ad esempio i dispositivi degli utenti, gli indirizzi e-mail o i numeri di telefono.

L'applicazione può quindi fornire dati o eventi di utilizzo. Visualizza i dati sugli eventi nella console Amazon Pinpoint per scoprire quanti utenti hai, con quale frequenza usano la tua applicazione, quando la utilizzano e molto altro.

Quando l'applicazione fornisce endpoint ed eventi, puoi usare queste informazioni per creare campagne di messaggistica mirate per specifici destinatari o segmenti. (È anche possibile inviare messaggi direttamente a semplici elenchi di destinatari senza creare campagne.)

Usa gli argomenti in questa sezione per integrare Amazon Pinpoint con un'applicazione Web o per dispositivi mobili. Questi argomenti includono esempi di codice e procedure per l'integrazione con un' JavaScriptapplicazione Android, Swift o Flutter. Per iniziare a integrare le tue app, consulta the section called “Connect l'applicazione frontend utilizzando Amplify”.

Oltre al tuo cliente, puoi utilizzare support AWS SDKs o l'API Amazon Pinpoint per importare endpoint, esportare dati sugli eventi, definire segmenti di clienti, creare ed eseguire campagne e altro ancora.

Argomenti

- Utilizzo di Amazon Pinpoint con un SDK AWS
- Connect l'applicazione frontend ad Amazon AWS Pinpoint utilizzando Amplify
- Registra gli endpoint Amazon Pinpoint nella tua applicazione
- Segnala gli eventi di Amazon Pinpoint nella tua applicazione

Utilizzo di Amazon Pinpoint con un SDK AWS

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK per C++	AWS SDK per C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice
AWS SDK per Java	AWS SDK per Java esempi di codice
AWS SDK per JavaScript	AWS SDK per JavaScript esempi di codice
AWS SDK per Kotlin	AWS SDK per Kotlin esempi di codice
AWS SDK per .NET	AWS SDK per .NET esempi di codice
AWS SDK per PHP	AWS SDK per PHP esempi di codice
AWS Strumenti per PowerShell	AWS Strumenti per PowerShell esempi di codice
AWS SDK per Python (Boto3)	AWS SDK per Python (Boto3) esempi di codice
AWS SDK per Ruby	AWS SDK per Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

Per esempi specifici di Amazon Pinpoint, consulta Esempi di codice per l'utilizzo di Amazon Pinpoint AWS SDKs.

> **Esempio di disponibilità**
>
> Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

Connect l'applicazione frontend ad Amazon AWS Pinpoint utilizzando Amplify

Usa AWS Amplify per integrare la tua app con. AWS Per le app Swift, consulta la sezione relativa alle nozioni di base nella documentazione di Amplify per Swift. Per le app Android, consulta la sezione relativa alle nozioni di base nella documentazione di Amplify per SDK per Android. Per l'app React Native, vedi Guida introduttiva nella documentazione di Amplify. JavaScript Per le app Flutter, consulta la sezione relativa alle nozioni di base nella documentazione dell'SDK per Flutter. Questi argomenti ti aiutano a:

- Configurare le risorse di back-end.
- Connetti la'app alle risorse di back-end utilizzando le librerie Amplify.

Per ulteriori informazioni su come collegare la tua app frontend ad Amazon Pinpoint per analisi, messaggistica in-app e notifiche push, consulta Amplify.AWS

Approfondimenti

Dopo aver integrato AWS Amplify con la tua applicazione, aggiorna il codice per registrare i dispositivi degli utenti come endpoint. Per ulteriori informazioni, consulta Registra gli endpoint Amazon Pinpoint nella tua applicazione.

Registra gli endpoint Amazon Pinpoint nella tua applicazione

Quando un utente avvia una sessione, ad esempio lanciando l'app per dispositivi mobili, l'applicazione Web o per dispositivi mobili può registrare (o aggiornare) automaticamente un endpoint in Amazon Pinpoint. L'endpoint rappresenta il dispositivo con cui l'utente avvia la sessione. Include gli attributi che descrivono il dispositivo e può includere anche attributi personalizzati. Gli endpoint possono rappresentare anche altri metodi per comunicare con i clienti, ad esempio indirizzi e-mail o numero di cellulare.

Quando l'applicazione ha registrato endpoint, è possibile segmentare i destinatari in base agli attributi degli endpoint. Puoi quindi coinvolgere questi segmenti con campagne di messaggistica mirate. Puoi anche utilizzare la pagina Analisi nella console Amazon Pinpoint per visualizzare grafici relativi alla registrazione e all'attività degli endpoint, ad esempio Nuovi endpoint e Endpoint attivi al giorno.

Puoi assegnare un unico ID utente a più endpoint. Un ID utente rappresenta un singolo utente, mentre ciascun endpoint a cui viene assegnato l'ID utente rappresenta uno dei dispositivi dell'utente. Dopo aver assegnato un utente IDs agli endpoint, puoi visualizzare i grafici sull'attività degli utenti nella console, ad esempio Utenti attivi ogni giorno e Utenti attivi mensilmente.

Prima di iniziare

Se non l'hai già fatto, integra AWS Mobile SDK per Android o iOS o integra la libreria AWS Amplify con la tua JavaScript applicazione. Per ulteriori informazioni, consulta Connect l'applicazione frontend ad Amazon AWS Pinpoint utilizzando Amplify.

Registra gli endpoint con il AWS cellulare SDKs per Android o iOS

Puoi utilizzare AWS Mobile SDKs per Android o iOS per registrare e personalizzare gli endpoint. Per maggiori informazioni ed esempi di codice, consulta i seguenti documenti:

- Registrazione di endpoint nell'applicazione nella documentazione dell'SDK per Android.
- Registrazione di endpoint nell'applicazione nella documentazione dell'SDK per iOS.

Registra gli endpoint con la libreria AWS Amplify JavaScript

Puoi utilizzare la libreria AWS Amplify JavaScript per registrare e aggiornare gli endpoint nelle tue app. Per ulteriori informazioni e per visualizzare esempi di codice, consulta Update endpoint nella documentazione di AWS Amplify. JavaScript

Passaggi successivi

Dopo aver aggiornato l'app per registrare gli endpoint, le informazioni sul dispositivo e gli attributi personalizzati vengono forniti ad Amazon Pinpoint quando gli utenti avviano l'app. Queste informazioni possono essere utilizzate per definire i segmenti di destinatari. Puoi anche utilizzare la console per visualizzare le metriche degli endpoint e gli utenti a cui è assegnato l'utente. IDs Puoi anche completare i passaggi indicati Segnala gli eventi di Amazon Pinpoint nella tua applicazione per aggiornare l'app e segnalare i dati di utilizzo.

Segnala gli eventi di Amazon Pinpoint nella tua applicazione

Nella tua applicazione mobile o web, puoi utilizzare AWS Mobile SDKs o l'API Amazon Pinpoint Events per segnalare dati o eventi di utilizzo ad Amazon Pinpoint. È possibile segnalare gli eventi per

acquisire informazioni come i tempi delle sessioni, il comportamento di acquisto degli utenti, i tentativi di accesso o qualsiasi tipo di evento personalizzato sia necessario.

Quando l'applicazione segnala gli eventi, puoi visualizzare i dati analitici nella console Amazon Pinpoint. I grafici nella pagina Analytics (Analisi) offrono parametri per molti aspetti del comportamento degli utenti. Per ulteriori informazioni, consulta la sezione Grafico di riferimento per l'analisi di Amazon Pinpoint nella Guida per l'utente di Amazon Pinpoint.

Per analizzare e archiviare i dati sugli eventi all'esterno di Amazon Pinpoint, puoi configurare Amazon Pinpoint per lo streaming dei dati a Amazon Kinesis. Per ulteriori informazioni, consulta Trasmetti in streaming i dati degli eventi delle app tramite Kinesis e Firehose utilizzando Amazon Pinpoint.

Utilizzando le librerie AWS Mobile SDKs e AWS JavaScript Amplify, puoi chiamare l'API Amazon Pinpoint per segnalare i seguenti tipi di eventi:

Eventi di sessione

Indicano quando e con quale frequenza gli utenti aprono e chiudono l'app.

Quando l'applicazione segnala gli eventi relativi alle sessioni, nella pagina Analisi della console Amazon Pinpoint vengono visualizzati i grafici Sessioni, Endpoint attivi al giorno, Percentuale di conservazione su 7 giorni e altri ancora.

Eventi personalizzati

Si tratta di eventi non standard vengono definiti assegnando un tipo di evento personalizzato. È possibile aggiungere attributi e parametri personalizzati a un evento personalizzato.

Nella pagina Analytics (Analisi) della console, la scheda Events (Eventi) mostra i parametri per tutti gli eventi personalizzati segnalati dall'app.

Eventi di monetizzazione

Segnalano il fatturato generato dall'applicazione e il numero di articoli acquistati dagli utenti.

Nella pagina Analytics (Analisi), la scheda Revenue (Fatturato) mostra i grafici Revenue (Fatturato), Paying users (Utenti paganti), Units sold (Unità vendute) e altri ancora.

Eventi di autenticazione

Indicano la frequenza con cui gli utenti effettuano l'autenticazione nell'applicazione.

Nella pagina Analytics (Analisi), la scheda Users (Utenti) mostra i grafici Sign-ins (Accessi), Sign-ups (Registrazioni) e Authentication failures (Errori di autenticazione).

Prima di iniziare

Se non lo hai ancora fatto, esegui queste operazioni:

- Integra la tua app con AWS Amplify. Per informazioni, consulta Connect l'applicazione frontend ad Amazon AWS Pinpoint utilizzando Amplify.
- Aggiorna l'applicazione per registrare gli endpoint. Per informazioni, consulta Registra gli endpoint Amazon Pinpoint nella tua applicazione.

Segnala eventi con il AWS cellulare SDKs per Android o iOS

Puoi abilitare un'app mobile per segnalare eventi ad Amazon Pinpoint utilizzando AWS Mobile per SDKs iOS e Android.

Per ulteriori informazioni sull'aggiornamento dell'app per registrare e inviare eventi ad Amazon Pinpoint, consulta le seguenti pagine nella documentazione di Amplify AWS :

- Analytics nella documentazione dell'SDK per iOS
- Analytics nella documentazione dell'SDK per Android

Segnala eventi con la libreria AWS Amplify JavaScript

Puoi abilitare JavaScript e React Native app per segnalare gli eventi di utilizzo delle applicazioni ad Amazon Pinpoint utilizzando la libreria Amplify AWS . JavaScript Per ulteriori informazioni sull'aggiornamento dell'app per inviare eventi ad Amazon Pinpoint, consulta Analytics nella documentazione di Amplify AWS . JavaScript

Segnala eventi con l'API Amazon Pinpoint

Puoi utilizzare l'API Amazon Pinpoint o un AWS SDK per inviare eventi ad Amazon Pinpoint in blocco. Per ulteriori informazioni, consulta Eventi nella documentazione di riferimento dell'API Amazon Pinpoint.

Passaggi successivi

Dopo aver aggiornato l'app per segnalare gli eventi, invia i dati di utilizzo ad Amazon Pinpoint. Puoi visualizzare questi dati nella console e trasmetterli in streaming ad Amazon Kinesis. Puoi anche

aggiornare l'app per gestire le notifiche push inviate con Amazon Pinpoint. Per ulteriori informazioni, consulta i seguenti argomenti nella Guida per l'utente di AWS End User Messaging Push.

- Configurazione delle notifiche push
- Configurazione delle notifiche push Swift
- Configurazione delle notifiche push Android
- Configurazione delle notifiche push di Flutter
- Configurazione delle notifiche push React Native
- Creazione di un progetto
- Gestione delle notifiche push

Invia messaggi transazionali dalla tua app utilizzando Amazon Pinpoint

Puoi utilizzare l'API Amazon Pinpoint e AWS SDKs inviare messaggi transazionali direttamente dalla tua app. I messaggi transazionali sono messaggi inviati a destinatari specifici, a differenza dei messaggi inviati a segmenti. Sono diversi i motivi per cui potresti voler inviare messaggi transazionali anziché messaggi basati su campagne. Ad esempio, è possibile inviare una conferma dell'ordine via e-mail quando un cliente effettua un ordine. È inoltre possibile inviare una password monouso tramite SMS o voce utilizzabile dal cliente per completare il processo di creazione di un account per il tuo servizio.

Questa sezione include il codice di esempio in diversi linguaggi di programmazione che puoi utilizzare per iniziare a inviare messaggi vocali, messaggi SMS ed e-mail transazionali.

Per altri esempi di codice su endpoint, segmenti e canali, consulta Esempi di codice.

Argomenti in questa sezione:

- Inviare e-mail transazionali utilizzando Amazon Pinpoint
- Inviare messaggi SMS transazionali utilizzando Amazon Pinpoint
- Inviare messaggi vocali con Amazon Pinpoint

Inviare e-mail transazionali utilizzando Amazon Pinpoint

Questa sezione fornisce esempi di codice completi utilizzabili per inviare messaggi e-mail transazionali tramite Amazon Pinpoint.

- Utilizzando l' SendMessages operazione nell'API Amazon Pinpoint: puoi utilizzare l'`SendMessages`operazione nell'API Amazon Pinpoint per inviare messaggi in tutti i canali supportati da Amazon Pinpoint, inclusi i canali di notifica push, SMS, voce ed e-mail.

  Il vantaggio di utilizzare questa operazione è che la sintassi richiesta per l'invio di messaggi è molto simile in tutti i canali. In questo modo è più semplice riutilizzare il codice esistente. L'`SendMessages`operazione consente inoltre di sostituire il contenuto dei messaggi e-mail e di inviare e-mail all' IDs endpoint Amazon Pinpoint anziché a indirizzi e-mail specifici.

Questa sezione include il codice di esempio in diversi linguaggi di programmazione che puoi utilizzare per iniziare a inviare e-mail transazionali.

Per ulteriori esempi di codice su endpoint, segmenti e canali, consulta Esempi di codice.

Scegli un metodo per inviare email

Il miglior metodo da utilizzare per l'invio di e-mail transazionali varia a seconda del caso d'uso specifico. Ad esempio, se è necessario inviare e-mail utilizzando un'applicazione di terze parti o se non è disponibile un AWS SDK per il linguaggio di programmazione, potrebbe essere necessario utilizzare l'interfaccia SMTP. Se desideri inviare messaggi in altri canali supportati da Amazon Pinpoint e desideri utilizzare il codice consistente per effettuare tali richieste, devi utilizzare l'operazione `SendMessages` nell'API Amazon Pinpoint.

Scegli tra Amazon Pinpoint e Amazon SES

Se invii un gran numero di e-mail transazionali, come le conferme di acquisto o i messaggi di reimpostazione della password, prendi in considerazione l'utilizzo di Amazon SES. Amazon SES ha un'API e un'interfaccia SMTP, entrambe adatte per l'invio di e-mail da applicazioni o servizi. Offre inoltre funzionalità di posta elettronica aggiuntive, tra cui funzionalità di ricezione e-mail, set di configurazione e funzionalità di autorizzazione per l'invio.

Amazon SES include anche un'interfaccia SMTP che puoi integrare con le applicazioni di terze parti esistenti, inclusi i servizi di gestione delle relazioni con i clienti (CRM) come Salesforce. Per maggiori informazioni sull'invio di e-mail tramite Amazon SES, consulta Amazon Simple Email Service Developer Guide.

Invio di e-mail tramite l'API Amazon Pinpoint

Questa sezione contiene esempi di codice completi che puoi utilizzare per inviare e-mail tramite l'API Amazon Pinpoint utilizzando un AWS SDK. Devi aver verificato un indirizzo e-mail o un dominio prima di poter inviare un messaggio.

C#

Utilizza questo esempio per l'invio di e-mail tramite l'AWS SDK per .NET. Questo esempio si basa sul presupposto che l' SDK per .NET sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base su AWS SDK per .NET nella Guida per gli sviluppatori di AWS SDK per .NET .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta Configurazione delle AWS credenziali nella Guida per gli AWS SDK per .NET sviluppatori.

Questo esempio di codice è stato testato utilizzando la AWS SDK per .NET versione 3.3.29.13 e la versione di runtime 2.1.2 .NET Core.

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

namespace SendEmailMessage;

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

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

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

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

        // The Amazon Pinpoint project/application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or application
```

```
        // that you choose.
        string appId = configuration["AppId"]!;

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

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

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

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

        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK for .NET)</
h1>"
                          + "\n  <p>This email was sent using the "
                          + "\n    <a href='https://aws.amazon.com/pinpoint/'>Amazon
 Pinpoint</a> API "
                          + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                          + "\n  </p>"
                          + "\n</body>"
```

```
                    + "\n</html>";

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

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

```
                    }
                }
            }
        };
        Console.WriteLine("Sending message...");
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        Console.WriteLine("Message sent!");
        return response.MessageResponse;
    }
}
```

Java

Utilizza questo esempio per l'invio di e-mail tramite l'AWS SDK per Java. Questo esempio si basa sul presupposto che l' AWS SDK for Java 2.x sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base nella Guida per gli sviluppatori di AWS SDK for Java 2.x .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta l'argomento relativo alla configurazione della regione e delle credenziali predefinite nella Guida per gli sviluppatori di AWS SDK per Java .

Questo esempio di codice è stato testato utilizzando la AWS SDK per Java versione 2.3.1 e la versione OpenJDK 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);
        }
    }
}
```

Per l'esempio completo dell'SDK, consulta .java on. SendEmailMessage GitHub

JavaScript (Node.js)

Utilizzate questo esempio per inviare e-mail utilizzando l'AWS SDK for JavaScript in Node.js. L'esempio presuppone che tu abbia già installato e configurato l'SDK per JavaScript in Node.js.

Per ulteriori informazioni, consulta Guida introduttiva all'AWS SDK for JavaScript in Node.js Developer Guide.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta Impostazione delle credenziali nell'AWS SDK for JavaScript in Node.js Developer Guide.

Questo esempio di codice è stato testato utilizzando l'SDK per JavaScript Node.js versione 2.388.0 e Node.js versione 11.7.0.

```
"use strict";

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

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

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

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

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

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

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

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

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

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

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

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

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
```

```
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

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

Python

Utilizza questo esempio per l'invio di e-mail tramite l'AWS SDK per Python (Boto3). Questo esempio si basa sul presupposto che SDK per Python (Boto3) sia già stato installato e configurato. Per ulteriori informazioni, consulta Quickstart nella documentazione di riferimento delle API SDK AWS per 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()
```

È inoltre possibile utilizzare modelli di messaggi per inviare messaggi e-mail, come illustrato nell'esempio seguente:

```
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()
```

Questi esempi presuppongono che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per informazioni dettagliate, consulta Credentials nella documentazione di riferimento delle API AWS SDK per Python (Boto3).

Aggiungi intestazioni di annullamento dell'iscrizione alle e-mail utilizzando Amazon Pinpoint

> **Note**
>
> Prima di poter utilizzare le intestazioni delle e-mail, è necessario impostare un ruolo di invio dell'orchestrazione delle e-mail se si inviano e-mail da una campagna o da un percorso. Per l'invio diretto di e-mail, devi disporre delle autorizzazioni per e. `ses:SendEmail` `ses:SendRawEmail` Per ulteriori informazioni, consulta Creazione di un ruolo di invio per l'orchestrazione delle e-mail nella Guida per l'utente di Amazon Pinpoint.

L'inclusione di un link di annullamento dell'iscrizione nell'e-mail rappresenta una best practice e in alcuni paesi è un requisito legale. Per aggiungere un link di annullamento dell'iscrizione con un clic, aggiungi le seguenti intestazioni:

1. Imposta il nome dell'intestazione `List-Unsubscribe` e imposta il valore sul link di annullamento dell'iscrizione. Il link deve supportare le richieste HTTP POST per elaborare la richiesta di annullamento dell'iscrizione del destinatario.
2. Imposta il nome dell'intestazione su **`List-Unsubscribe-Post`** e imposta il valore su. `List-Unsubscribe=One-Click`

Puoi aggiungere fino a 15 intestazioni a un messaggio di posta elettronica. Per un elenco delle intestazioni supportate, consulta i campi di intestazione di Amazon SES nella Amazon Simple Email Service Developer Guide.

L'esempio seguente mostra come inviare un messaggio e-mail con intestazioni di annullamento dell'iscrizione utilizzando il. AWS Command Line InterfacePer ulteriori informazioni sulla configurazione di AWS CLI, vedere Configure the AWS CLI nella Guida per l'utente.AWS Command Line Interface

Nel comando seguente, effettuate le seguenti operazioni:

- Sostituisci *AppId* con l'ID dell'applicazione.
- Sostituisci *richard_roe@example.com* con l'indirizzo e-mail del destinatario.
- *https://example.com/unsub*Sostituiscilo con il tuo link di annullamento dell'iscrizione.
- *example123456*Sostituiscilo con un identificativo univoco per il destinatario.

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

```
          },
          {
            "Name": "List-Unsubscribe-Post",
            "Value": "List-Unsubscribe=One-Click"
          }
        ]
      }
    }
  }
}'
```

Inviare messaggi SMS transazionali utilizzando Amazon Pinpoint

Puoi utilizzare l'API Amazon Pinpoint per inviare messaggi SMS (messaggi di testo) a numeri di telefono o endpoint specifici. IDs Questa sezione contiene esempi di codice completi che puoi utilizzare per inviare messaggi SMS tramite l'API Amazon Pinpoint utilizzando un AWS SDK. Il tuo account deve essere in produzione e disporre di un'identità di origine attiva in grado di inviare messaggi SMS.

Per altri esempi di codice su endpoint, segmenti e canali, consulta Esempi di codice.

C#

Utilizza questo esempio per inviare un messaggio SMS tramite l'AWS SDK per .NET. Questo esempio si basa sul presupposto che l' SDK per .NET sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base nella Guida per gli sviluppatori di AWS SDK per .NET .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente IAM esistente. Per ulteriori informazioni, consulta l'argomento relativo alla configurazione delle credenziali AWS nella Guida per gli sviluppatori di AWS SDK per .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

Utilizza questo esempio per inviare un messaggio SMS tramite l'AWS SDK per Java. In questo esempio si presume che SDK per Java sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base nella Guida per gli sviluppatori di AWS SDK per Java .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente IAM esistente. Per ulteriori informazioni, consulta l'argomento relativo alla configurazione della regione e delle credenziali predefinite nella Guida per gli sviluppatori di AWS SDK per Java .

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
```

```
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

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

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

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

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

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

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

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

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

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

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

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

        public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
```

```
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

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

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

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

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

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

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

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

```
                }
        }
}
```

Per l'esempio completo dell'SDK, consulta SendMessage .java on. GitHub

JavaScript (Node.js)

Utilizzate questo esempio per inviare un messaggio SMS utilizzando l'AWS SDK for JavaScript in Node.js. L'esempio presuppone che tu abbia già installato e configurato l'SDK per JavaScript in Node.js. Per ulteriori informazioni, consulta Guida introduttiva all'AWS SDK for JavaScript in Node.js Developer Guide.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente IAM esistente. Per ulteriori informazioni, consulta Impostazione delle credenziali nell'AWS SDK for JavaScript in Node.js Developer Guide.

```
"use strict";

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

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

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

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

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

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

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

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

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

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

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

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

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
```

```
        SenderId: senderId,
      },
    },
  },
};

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

Python

Utilizza questo esempio per inviare un messaggio SMS tramite l'AWS SDK per Python (Boto3). Questo esempio si basa sul presupposto che SDK per Python sia già stato installato e configurato. Per ulteriori informazioni, consulta Quickstart in AWS SDK for Python (Boto3) Getting Started.

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

logger = logging.getLogger(__name__)


def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
```

```
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you send
                   this message. The SMS channel must be enabled for the project or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This phone
                               number must be associated with your Amazon Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you send
                         time-sensitive content, specify TRANSACTIONAL. If you send
                         marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
```

```
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK for
 "
        "Python (Boto 3)."
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")


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

È inoltre possibile utilizzare modelli di messaggi per inviare messaggi SMS, come illustrato nell'esempio seguente:

```
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()
```

Questi esempi presuppongono che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente IAM esistente. Per informazioni dettagliate, consulta Credentials nella documentazione di riferimento delle API AWS SDK per Python (Boto3).

Inviare messaggi vocali con Amazon Pinpoint

Puoi utilizzare l'API Amazon Pinpoint per inviare messaggi vocali a numeri di telefono specifici. Questa sezione contiene esempi di codice completi che puoi utilizzare per inviare messaggi vocali tramite l'API SMS e Voice di Amazon Pinpoint utilizzando un AWS SDK. Il tuo account deve essere in produzione e disporre di un'identità di origine attiva in grado di inviare messaggi vocali.

Per altri esempi di codice su endpoint, segmenti e canali, consulta Esempi di codice.

Java

Utilizza questo esempio per l'invio di un messaggio vocale tramite l'AWS SDK per Java. In questo esempio si presume che SDK per Java sia già stato installato e configurato. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base nella Guida per gli sviluppatori di AWS SDK per Java .

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori

informazioni, consulta Configurare AWS le credenziali e la regione per lo sviluppo nella Guida per gli AWS SDK per Java sviluppatori.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

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

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

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

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

```
public class SendVoiceMessage {

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

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

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

    public static void main(String[] args) {

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

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

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

        // Set the content type to application/json.
```

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

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

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

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

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

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

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

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

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

```
            System.exit(1);
        }
    }
}
```

Per l'esempio completo dell'SDK, consulta SendVoiceMessage.java on. GitHub

JavaScript (Node.js)

Utilizzate questo esempio per inviare un messaggio vocale utilizzando l' AWS SDK for JavaScript in Node.js. L'esempio presuppone che tu abbia già installato e configurato l'SDK per JavaScript in Node.js.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta Impostazione delle credenziali nell'AWS SDK for JavaScript in Node.js Developer Guide.

```
"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

Utilizza questo esempio per l'invio di un messaggio vocale tramite l' AWS SDK per Python (Boto3). Questo esempio si basa sul presupposto che SDK per Python (Boto3) sia già stato installato e configurato.

Questo esempio presuppone che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per informazioni dettagliate, consulta Credentials nella documentazione di riferimento delle API AWS SDK per Python (Boto3).

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

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
```

```
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
message.
    :param ssml_message: The content of the message. This example uses SSML to
control
                         certain aspects of the message, such as the volume and the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
        )
        raise
    else:
        return response["MessageId"]
```

```
def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


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

Usa l'API SMS and Voice Messaging per l'utente AWS finale, versione 2

Amazon Pinpoint include un'API, chiamata SMS and Voice API, versione 2, progettata per l'invio di SMS e messaggi vocali. Mentre l'API Amazon Pinpoint si concentra sull'invio di messaggi tramite campagne e percorsi pianificati e basati su eventi, l'API SMS e Voce offre nuove funzioni e funzionalità per l'invio diretto di SMS e messaggi vocali a singoli destinatari. Puoi utilizzare l'API SMS and Voice indipendentemente dalle funzionalità della campagna e del percorso di Amazon Pinpoint oppure puoi utilizzarle entrambe contemporaneamente per soddisfare diversi casi d'uso. Se utilizzi già Amazon Pinpoint per inviare SMS o messaggi vocali, il tuo account è già configurato per utilizzare questa API.

Questa API è una buona soluzione per gli utenti che dispongono di un'architettura multi-tenant, come Independent Software Vendors (). ISVs Questa API semplifica la separazione dei dati relativi a eventi, numeri di telefono di origine ed elenchi di opt-out per i diversi tenant.

Quando utilizzi l'API SMS e Voce, ti consigliamo di configurare set di configurazione e destinazioni di evento. L'API SMS e Voce non genera automaticamente i dati sugli eventi per i messaggi inviati. La configurazione delle destinazioni di evento garantisce l'acquisizione di dati importanti sugli eventi, come il recapito dei messaggi e gli eventi di errore.

La versione 2 di questa API è stata preceduta dalla versione 1. Se attualmente utilizzi la versione 1 di questa API, questa continuerà a essere disponibile e potrai continuare a usarla. Se esegui la migrazione alla versione 2, otterrai funzionalità aggiuntive, come la possibilità di creare pool di numeri di telefono, richiedere nuovi numeri di telefono a livello di codice e abilitare o disabilitare determinate funzionalità dei numeri di telefono.

> **Note**
>
> Alcune attività possono essere completate solo utilizzando la console Amazon Pinpoint. Ad esempio, verificare un numero di telefono da utilizzare mentre l'account è nella sandbox SMS e registrarsi per utilizzare 10DLC.

Per ulteriori informazioni sull'API Amazon Pinpoint SMS and Voice versione 2, consulta il riferimento all'API SMS and Voice, versione 2. Per informazioni su come creare, configurare e gestire le risorse

SMS e vocali di messaggistica con l'utente AWS finale, consulta la Guida per l'utente SMS di messaggistica con l'utente AWS finale

Genera password monouso (OTPs) con Amazon Pinpoint

Amazon Pinpoint include una funzionalità di gestione delle password monouso (OTP) che puoi utilizzare per generare nuove password monouso e inviarle ai destinatari come messaggi SMS.

> **Important**
>
> Per utilizzare questa funzionalità, il tuo account deve avere accesso alla produzione e un'identità di origine attiva. Per ulteriori informazioni, consulta Informazioni sulla sandbox SMS/MMS e Voice e Richiedi un numero di telefono nella Guida per l'utente di SMS di messaggistica per l'utente AWS finale.

In alcuni paesi e aree geografiche, è necessario ottenere un numero di telefono o un ID di origine dedicato prima di poter inviare messaggi SMS. Ad esempio, quando invii messaggi a destinatari negli Stati Uniti, devi disporre di un numero verde dedicato, un numero 10DLC o un codice breve. Quando invii messaggi a destinatari in India, devi disporre di un ID mittente registrato, che include un ID entità principale (PEID) e un ID modello. Questi requisiti continuano a essere valido quando si utilizza la funzionalità OTP.

Per utilizzare questa funzionalità, sono necessarie le autorizzazioni per inviare e verificare i messaggi OTP; consulta Codici OTP (One-Time Password). Se hai bisogno di aiuto per determinare le autorizzazioni, consulta. Risoluzione dei problemi relativi alla gestione di identità e accessi di Amazon Pinpoint

Puoi utilizzare l'operazione `SendOtpMessages` nell'API Amazon Pinpoint per inviare un codice OTP (One-Time Password) a un utente della tua applicazione. Quando usi questa API, Amazon Pinpoint genera un codice casuale e lo invia all'utente come messaggio SMS. La tua richiesta può includere i seguenti parametri:

- `Channel`: canale di comunicazione attraverso il quale viene inviato il codice OTP. Attualmente sono supportati solo i messaggi SMS, quindi l'unico valore accettabile è SMS.
- `BrandName`: nome del marchio, dell'azienda o del prodotto associato al codice OTP. Il nome può contenere un massimo di 20 caratteri.

> **Note**
>
> Quando Amazon Pinpoint invia il messaggio OTP, il nome del marchio viene inserito automaticamente nel seguente modello di messaggio:
>
> ```
> This is your One Time Password: {{otp}} from {{brand}}
> ```
>
> Quindi, se specifichi ExampleCorp come nome del marchio e Amazon Pinpoint genera una password monouso di 123456, invia il seguente messaggio all'utente:
>
> ```
> This is your One Time Password: 123456 from ExampleCorp
> ```

- `CodeLength`: numero di cifre contenute nel codice OTP inviato al destinatario. I codici OTP possono contenere da 5 a 8 cifre.
- `ValidityPeriod`: periodo di validità, espresso in minuti, del codice OTP. Il periodo di validità può essere compreso tra 5 e 60 minuti.
- `AllowedAttempts`: numero di volte in cui il destinatario può tentare senza successo di verificare il codice OTP. Se il numero di tentativi supera questo valore, il codice OTP diventa automaticamente non valido. Il numero massimo di tentativi consentiti è 5.
- `Language`: lingua, in formato IETF BCP-47, da utilizzare per l'invio del messaggio. I valori accettabili sono:
  - `de-DE`: tedesco
  - `en-GB`: inglese (Regno Unito)
  - `en-US`: inglese (Stati Uniti)
  - `es-419`: spagnolo (America Latina)
  - `es-ES`: spagnolo
  - `fr-CA`: francese (Canada)
  - `fr-FR`: francese
  - `it-IT`: italiano
  - `ja-JP`: giapponese
  - `ko-KR`: coreano
  - `pt-BR`: portoghese (Brasile)
  - `zh-CN`: cinese (semplificato)

  - `zh-TW`: cinese (tradizionale)
- `OriginationIdentity`: identità di origine (ad esempio codice lungo, codice breve o ID mittente) utilizzata per inviare il codice OTP. Se utilizzi un codice lungo o un numero verde per inviare il codice OTP (One-Time Password), il numero di telefono deve essere in formato E.164.
- `DestinationIdentity`: numero di telefono, nel formato E.164, al quale è stato inviato il codice OTP.
- `ReferenceId`: ID di riferimento univoco per la richiesta. L'ID di riferimento corrisponde esattamente all'ID di riferimento fornito al momento della verifica del codice OTP. L'ID di riferimento può contenere da 1 a 48 caratteri.
- `EntityId`: ID dell'entità registrato presso un ente regolatore. Questo parametro viene attualmente utilizzato solo per l'invio di messaggi ai destinatari in India. Se non effettui invii a destinatari in India, puoi omettere questo parametro.
- `TemplateId`: ID del modello registrato presso un ente regolatore. Questo parametro viene attualmente utilizzato solo per l'invio di messaggi ai destinatari in India. Se non effettui invii a destinatari in India, puoi omettere questo parametro.

> **Note**
>
> Per ulteriori informazioni sui requisiti per l'invio di messaggi a destinatari in India, consulta la procedura di registrazione dell'ID mittente indiano nella Guida per l'utente di Amazon Pinpoint.

Per assicurarti che il tuo account Amazon Pinpoint sia configurato correttamente per inviare messaggi OTP, puoi utilizzare l'opzione AWS CLI per inviare un messaggio di prova. Per ulteriori informazioni su AWS CLI, consulta la Guida per l'AWS Command Line Interface utente.

Per inviare un messaggio OTP di prova utilizzando il AWS CLI, esegui il send-otp-messagecomando nel terminale:

```
aws pinpoint send-otp-message --application-id 7353f53e6885409fa32d07cedexample --send-
otp-message-request-parameters
 Channel=SMS,BrandName=ExampleCorp,CodeLength=5,ValidityPeriod=20,AllowedAttempts=5,Origination
```

Nel comando precedente, procedi come segue.

- *7353f53e6885409fa32d07cedexample*Sostituiscilo con l'ID dell'applicazione.

- Sostituiscilo *ExampleCorp* con il nome della tua azienda.
- *5*Sostituiscilo `CodeLegth` con il numero di cifre che sarà contenuto nel codice OTP che verrà inviato al destinatario.
- Sostituisci *20* `ValidityPeriod` con il periodo di validità, in minuti, entro il quale il codice OTP sarà valido.
- Sostituisci `AllowedAttempts` con il numero di volte *5* in cui il destinatario può tentare senza successo di verificare l'OTP.
- *+18555550142*`OriginationIdentity`Sostituiscilo con l'identità di origine utilizzata per inviare il codice OTP.
- *+12065550007*`DestinationIdentity`Sostituiscilo con il numero di telefono a cui inviare il codice OTP.
- *SampleReferenceId*`ReferenceId`Sostituiscilo con un ID di riferimento univoco per la richiesta.

Risposta **`SendOtpMessage`**

Quando invii correttamente un messaggio OTP, viene visualizzata una risposta analoga all'esempio seguente:

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

Convalida dei messaggi OTP in Amazon Pinpoint

Dopo aver inviato un one-time-password, l'applicazione può chiamare l'API Amazon Pinpoint per verificarlo. Per verificare un codice OTP, chiama l'API `VerifyOtpMessages`. La richiesta deve includere i seguenti parametri:

- `DestinationIdentity`: numero di telefono, nel formato E.164, al quale è stato inviato il codice OTP.
- `ReferenceId`: ID di riferimento usato quando hai inviato il codice OTP al destinatario. L'ID di riferimento deve corrispondere esattamente.
- `Otp`: codice OTP che stai convalidando.

Puoi utilizzare il AWS CLI per testare il processo di convalida. Per ulteriori informazioni sull'installazione e la configurazione di AWS CLI, consulta la Guida per l'AWS Command Line Interface utente.

Per verificare una OTP utilizzando il AWS CLI, esegui il verify-otp-messagecomando nel terminale:

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

Nel comando precedente, procedi come segue.

- *7353f53e6885409fa32d07cedexample*Sostituiscilo con l'ID dell'applicazione.
- *+12065550007*`DestinationIdentity`Sostituiscilo con il numero di telefono a cui è stato inviato il codice OTP.
- *SampleReferenceId*`ReferenceId`Sostituiscilo con un ID di riferimento univoco per la richiesta. Questo valore deve corrispondere a `ReferenceID` quello utilizzato per inviare la richiesta.
- *01234*Sostituiscilo `Otp` con un Otp inviato a. `DestinationIdentity`

Risposta **`VerifyOtpMessage`**

Quando invii una richiesta all'API `VerifyOTPMessage`, viene restituito un oggetto `VerificationResponse` contenente un'unica proprietà (`Valid`). Se l'ID di riferimento, il numero di telefono e il codice OTP corrispondono tutti ai valori previsti da Amazon Pinpoint e se il codice OTP

non è scaduto, il valore `Valid` è `true`. In caso contrario, è `false`. Di seguito è riportato un esempio di risposta per una verifica riuscita del codice OTP.

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

Esempi di codice OTP per l'utilizzo di SDK for Python (Boto3) in Amazon Pinpoint

Questa sezione contiene esempi di codice che mostrano come utilizzare SDK per Python (Boto3) per inviare e verificare codici OTP (One-Time Password).

Genera un ID di riferimento

La seguente funzione genera un ID di riferimento univoco per ogni destinatario, in base al numero di telefono del destinatario, al prodotto o al marchio per cui il destinatario riceve un codice OTP (One-Time Password) e all'origine della richiesta (che, ad esempio, può essere il nome di una pagina di un sito o di un'app). Quando verifichi il codice OTP, devi fornire un ID di riferimento identico affinché la convalida abbia esito positivo. Sia l'esempio di codice di invio che quello di convalida utilizzano questa funzione di utilità.

Questa funzione non è obbligatoria, ma è un valido strumento per adattare il processo di invio e verifica del codice OTP a una transazione specifica in un modo da semplificarne il nuovo invio durante la fase di verifica. Puoi utilizzare qualsiasi ID di riferimento: questo infatti è solo un esempio di base. Tuttavia, gli altri esempi di codice in questa sezione si basano su questa funzione.

```
# 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()
```

Invia codici OTP

L'esempio di codice seguente mostra come utilizzare SDK per Python (Boto3) per inviare un codice OTP (One-Time Password).

```
# 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")
```

Convalida i codici OTP

L'esempio di codice seguente mostra come utilizzare SDK per Python (Boto3) per verificare un codice OTP (One-Time Password) già inviato. Affinché la fase di convalida abbia esito positivo, la richiesta deve includere un ID di riferimento che corrisponda esattamente all'ID di riferimento utilizzato per inviare il messaggio.

```
# 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")
```

Personalizza i messaggi in-app con Amazon Pinpoint e Amplify

Puoi utilizzare i messaggi in-app per inviare messaggi mirati agli utenti delle tue applicazioni. I messaggi in-app sono altamente personalizzabili. Possono includere pulsanti che aprono siti Web o indirizzano gli utenti a parti specifiche dell'app. Puoi configurare i colori dello sfondo e del testo, posizionare il testo e aggiungere pulsanti e immagini alla notifica. Puoi inviare un unico messaggio o creare un carosello contenente un massimo di cinque messaggi univoci. Per una panoramica dei messaggi in-app, comprese le istruzioni per la creazione di modelli di messaggi in-app, consulta Creazione di modelli in-app nella Guida per l'utente di Amazon Pinpoint.

Puoi utilizzarlo AWS Amplify per integrare senza problemi le funzionalità di messaggistica in-app di Amazon Pinpoint nella tua app. Amplify può gestire automaticamente i processi di recupero e rendering dei messaggi, nonché il processo di invio di dati di analisi ad Amazon Pinpoint. Questa integrazione è attualmente supportata per le applicazioni React Native. Per ulteriori informazioni, consulta l'argomento relativo alla messaggistica in-app nella documentazione del Framework Amplify.

Recupera messaggi in-app per un endpoint in modo programmatico utilizzando Amazon Pinpoint

Le tue applicazioni possono chiamare l'GetInAppMessagesAPI per recuperare tutti i messaggi in-app a cui ha diritto un determinato endpoint. Quando esegui la chiamata all'API `GetInAppMessages`, è necessario fornire i parametri riportati di seguito:

- `ApplicationId`: ID univoco del progetto Amazon Pinpoint a cui è associata la campagna di messaggi in-app.
- `EndpointId`: ID univoco dell'endpoint per cui stai recuperando i messaggi.

Quando viene chiamata con questi valori, l'API restituisce un elenco di messaggi. Per ulteriori informazioni sulle risposte generate da questa operazione, consulta `GetInAppMessages`Esempio JSON di risposta dell'API Amazon Pinpoint.

È possibile utilizzare il AWS SDKs per chiamare l'operazione. `GetInAppMessages` Gli esempi di codice seguenti includono funzioni che recuperano i messaggi in-app.

JavaScript

Crea il client in un modulo separato e lo esporta.

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

Recupera i messaggi in-app di un endpoint:

```
// 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()
```

`GetInAppMessages`Esempio JSON di risposta dell'API Amazon Pinpoint

Quando si chiama l'operazione GetInAppMessagesAPI, viene restituito un elenco di messaggi a cui l'endpoint specificato ha diritto. L'app può quindi eseguire il rendering del messaggio in base ai valori della risposta.

Di seguito è riportato un esempio dell'oggetto JSON restituito quando si esegue una chiamata all'API `GetInAppMessages`:

```
{
  "InAppMessagesResponse":{
    "InAppMessageCampaigns":[
      {
        "CampaignId":"inAppTestCampaign-4c545b28d21a490cb51b0b364example",
        "DailyCap":0,
        "InAppMessage":{
          "Content":[
            {
              "BackgroundColor":"#f8e71c",
              "BodyConfig":{
                "Alignment":"CENTER",
                "Body":"This is a sample in-app message sent using Amazon Pinpoint.",
                "TextColor":"#d0021b"
              },
              "HeaderConfig":{
                "Alignment":"CENTER",
                "Header":"Sample In-App Message",
                "TextColor":"#d0021b"
              },
              "ImageUrl":"https://example.com/images/thumbnail.png",
              "PrimaryBtn":{
                "DefaultConfig":{
                  "BackgroundColor":"#d0021b",
                  "BorderRadius":50,
                  "ButtonAction":"CLOSE",
                  "Text":"Dismiss",
                  "TextColor":"#f8e71c"
                }
              }
            }
          ],
```

```
          "Layout":"MIDDLE_BANNER"
        },
        "Priority":3,
        "Schedule":{
          "EndDate":"2021-11-06T00:08:05Z",
          "EventFilter":{
            "Dimensions":{
              "Attributes":{

              },
              "EventType":{
                "DimensionType":"INCLUSIVE",
                "Values":[
                  "_session.start"
                ]
              },
              "Metrics":{

              }
            }
          }
        },
        "SessionCap":0,
        "TotalCap":0,
        "TreatmentId":"0"
      }
    ]
  }
}
```

Le seguenti sezioni forniscono informazioni sui componenti di questa risposta e sui relativi attributi.

Oggetto **InAppMessageCampaigns**

L'oggetto `InAppMessageCampaigns` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`CampaignId`	Stringa contenente il nome e l'ID univoco della campagna Amazon Pinpoint da cui è stato inviato il messaggio	Creato automaticamente da Amazon Pinpoint al momento della creazione della campagna.

Attributo	Descrizione	Dove viene impostato
	. Il nome precede l'ID della campagna. I due valori sono separati da un trattino (-).	
`TreatmentId`	Numero intero che rappresenta l'ID del trattamento della campagna per questo messaggio. Se la campagna ha un solo trattamento, il valore è `0`.	
`Priority`	La priorità del messaggio in-app, espressa come numero intero compreso tra 1 e 5, incluso, dove 1 indica la priorità più alta e 5 indica la priorità più bassa.	Passaggio 1 del processo di creazione della campagna.
`InAppMessage`	Oggetto `InAppMessage` contenente informazioni su come viene visualizzato il messaggio.	In base al contenuto del modello di messaggio in-app specificato per la campagna.
`Schedule`	Oggetto Schedule contenente informazioni su quando il messaggio è stato inviato.	Passaggio 4 del processo di creazione della campagna (se la campagna è stata creata nella console) o dell'oggetto `Schedule` (se la campagna è stata creata utilizzando l'API o un SDK).

Attributo	Descrizione	Dove viene impostato
`DailyCap`	Numero di volte, espresso come numero intero, in cui un messaggio in-app può essere mostrato all'utente nell'arco di 24 ore.	Ereditato dalle impostazioni a livello di progetto. Se la campagna include impostazioni che sovrascrivono le impostazioni del progetto, vengono utilizzate le impostazioni della campagna.
`SessionCap`	Numero di volte, espresso come numero intero, in cui un messaggio in-app può essere mostrato all'utente durante una sessione dell'applicazione.	
`TotalCap`	Numero totale di volte, espresso come numero intero, in cui un messaggio in-app può essere mostrato a un endpoint per campagna.	

Oggetto **`InAppMessage`**

L'oggetto `InAppMessage` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`Content`	Un array contenente un InAppMessageContentoggetto, che descrive il contenuto del messaggio.	In base al contenuto del modello di messaggio in-app specificato per la campagna.
`Layout`	Stringa che descrive come verrà visualizzato il messaggio in-app sul dispositivo del destinatario. I valori possibili sono:	

Attributo	Descrizione	Dove viene impostato
	• `BOTTOM_BANNER` : un messaggio visualizzato come banner nella parte inferiore della pagina. • `TOP_BANNER` : un messaggio visualizzato come banner nella parte superiore della pagina. • `OVERLAYS`: un messaggio che copre l'intero schermo. • `MOBILE_FEED` : un messaggio visualizzato in una finestra davanti alla pagina. • `MIDDLE_BANNER` : un messaggio visualizzato come banner nella parte centrale della pagina. • `CAROUSEL`: un layout scorrevole contenente fino a cinque messaggi univoci.	

Oggetto **`HeaderConfig`**

L'oggetto `HeaderConfig` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`Alignment`	Stringa che specifica l'allinea mento del testo dell'inte stazione. I valori possibili sono `LEFT`, `CENTER` e `RIGHT`.	In base al contenuto del modello di messaggio in-app specificato per la campagna.

Attributo	Descrizione	Dove viene impostato
`Header`	Testo dell'intestazione del messaggio.	
`TextColor`	Colore del testo dell'inte stazione, espresso come codice colore esadecimale (come "#000000" per il nero).	

Oggetto **`BodyConfig`**

L'oggetto `BodyConfig` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`Alignment`	Stringa che specifica l'allineamento del corpo del messaggio. I valori possibili sono `LEFT`, `CENTER` e `RIGHT`.	In base al contenuto del modello di messaggio in-app specificato per la campagna.
`Body`	Il corpo del testo principale del messaggio.	
`TextColor`	Colore del corpo del testo, espresso sotto forma di stringa contenente un codice colore esadecimale (come "#000000" per il nero).	

Oggetto **`InAppMessageContent`**

L'oggetto `InAppMessageContent` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`BackgroundColor`	Colore di sfondo del messaggio in-app, espresso come stringa contenente un codice colore esadecimale (ad esempio "#000000" per il nero).	In base al contenuto del modello di messaggio in-app specificato per la campagna.
`BodyConfig`	Un BodyConfigoggetto, che contiene informazioni relative al contenuto principale del messaggio.	
`HeaderConfig`	Un HeaderConfigoggetto che contiene informazioni relative all'intestazione o al titolo del messaggio.	
`ImageUrl`	URL dell'immagine visualizzata nel messaggio.	
`PrimaryBtn`	Un InAppMessageButton oggetto che contiene informazioni sul pulsante principale del messaggio.	
`SecondaryBtn`	Un InAppMessageButton oggetto che contiene informazioni sul pulsante secondario del messaggio. Non presente se il modello di messaggio in-app non specifica un pulsante secondario.	

Oggetto `Schedule`

L'oggetto `Schedule` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`EndDate`	Ora pianificata, in formato ISO 8601, per il termine della campagna.	Passaggio 4 del processo di creazione della campagna (se la campagna è stata creata nella console) o dell'oggetto `Schedule` (se la campagna è stata creata utilizzando l'API o un SDK).
`EventFilter`	Informazioni sull'evento che causa la visualizzazione del messaggio in-app. Quando generi un evento che corrisponde a una campagna in-app Amazon Pinpoint, viene visualizzato il messaggio.	

Oggetto `InAppMessageButton`

Un oggetto `InAppMessageButton` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`DefaultConfig`	Un DefaultButtonConfig oggetto che contiene informazioni sulle impostazioni predefinite per un pulsante in un messaggio in-app.	In base al contenuto del modello di messaggio in-app specificato per la campagna.
`Android`	Un OverrideButtonConfig oggetto che specifica il comportamento del pulsante sui dispositivi Android. Ciò sovrascrive la configura zione predefinita dei pulsanti	

Attributo	Descrizione	Dove viene impostato
	specificata nell'oggetto `DefaultConfig` .	
`IOS`	Un OverrideButtonConfigoggetto che specifica il comportamento del pulsante sui dispositivi iOS. Ciò sovrascrive la configurazione predefinita dei pulsanti specificata nell'oggetto `DefaultConfig` .	
`Web`	Un OverrideButtonConfigoggetto che specifica il comportamento del pulsante nelle app Web. Ciò sovrascrive la configurazione predefinita dei pulsanti specificata nell'oggetto `DefaultConfig` .	

Oggetto **`DefaultButtonConfig`**

Un oggetto `DefaultButtonConfig` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`BackgroundColor`	Colore di sfondo del pulsante, espresso come stringa contenente un codice colore esadecimale (ad esempio "#000000" per il nero).	In base al contenuto del modello di messaggio in-app specificato per la campagna.
`BorderRadius`	Raggio del bordo del pulsante in pixel, espresso come	

Attributo	Descrizione	Dove viene impostato
	numero intero. Un numero più alto genera angoli più arrotondati.	
`ButtonAction`	Stringa che descrive l'azione che si verifica quando un destinatario sceglie un pulsante in un messaggio in-app. I valori possibili sono: • `LINK`: un collegamento a una destinazione Web. • `DEEP_LINK` : un collegamento a una pagina specifica di un'applicazione. • `CLOSE`: elimina il messaggio .	
`Link`	URL di destinazione di un pulsante. Non presente per i pulsanti in cui è presente. ButtonAction `CLOSE`	
`Text`	Il testo visualizzato sul pulsante.	
`TextColor`	Colore del testo del pulsante, espresso come stringa contenente un codice colore esadecimale (ad esempio "#000000" per il nero).	

Oggetto `OverrideButtonConfig`

L'oggetto `OverrideButtonConfig` è presente solo se il modello di messaggio in-app utilizza pulsanti di override. Un pulsante di override è un pulsante con una configurazione specifica per un particolare tipo di dispositivo, ad esempio un dispositivo iOS, un dispositivo Android o un browser Web.

Un oggetto `OverrideButtonConfig` contiene gli attributi seguenti:

Attributo	Descrizione	Dove viene impostato
`ButtonAction`	Azione che si verifica quando un destinatario sceglie un pulsante in un messaggio in-app. I valori possibili sono: • `LINK`: un collegamento a una destinazione Web. • `DEEP_LINK` : un collegamento a una pagina specifica di un'applicazione. • `CLOSE`: elimina il messaggio.	In base al contenuto del modello di messaggio in-app specificato per la campagna.
`Link`	URL di destinazione di un pulsante. Non presente per i pulsanti in cui l'attributo `ButtonAction` è impostato su `CLOSE`.	
`Text`	Il testo visualizzato sul pulsante.	
`TextColor`	Colore del testo del pulsante, espresso come stringa contenente un codice colore esadecimale (ad esempio "#000000" per il nero).	

Utilizza il servizio di convalida dei numeri di telefono di Amazon Pinpoint

Amazon Pinpoint include un servizio di convalida dei numeri di telefono che è possibile utilizzare per stabilire se un numero di telefono è valido e per ottenere ulteriori informazioni su tale numero. Ad esempio, quando si utilizza il servizio di convalida dei numeri di telefono, restituisce le informazioni riportate di seguito.

- Il numero di telefono nel formato E.164.
- Il tipo di numero di telefono (ad esempio mobile, rete fissa o VoIP).
- La città e il paese collegati al numero di telefono.
- Il fornitore di servizi associato al numero di telefono.

Non vi sono costi aggiuntivi per l'utilizzo del servizio di convalida dei numeri di telefono. Per ulteriori informazioni, consulta Prezzi di Amazon Pinpoint.

> **Important**
>
> Per i numeri di telefono la cui origine è negli Stati Uniti e in Canada, l'API di convalida dei numeri di telefono non restituirà più dati per `City`, `County`, `Timezone` e `ZipCode`.

Casi d'uso per la convalida dei numeri di telefono di Amazon Pinpoint

È possibile utilizzare il servizio di convalida dei numeri di telefono per diversi casi d'utilizzo, tra cui i seguenti:

- Verifica di numeri di telefono forniti su un modulo Web: se utilizzi moduli basati sul Web per raccogliere informazioni di contatto dai clienti, devo convalidare il numero di telefono fornito dai clienti prima di inviare il modulo. Utilizza il back-end del sito Web per convalidare il numero utilizzando l'API di Amazon Pinpoint. La risposta dell'API indica se il numero non è valido, ad esempio se il numero di telefono è formattato in modo errato. Se verifichi che il numero di telefono che il cliente ha fornito non è valido, il modulo di richiesta del cliente Web può chiedere al cliente di fornire un numero diverso.

- Pulizia del database di contatti esistente: se disponi di un database dei numeri di telefono dei clienti, puoi convalidare ogni numero di telefono e aggiornare il database in base ai risultati. Ad esempio, se trovi endpoint con i numeri di telefono che non sono in grado di ricevere messaggi SMS, puoi modificare la proprietà `ChannelType` per l'endpoint da `SMS` a `VOICE`. Puoi prima convalidare il numero di telefono e poi aggiornare la proprietà `ChannelType` per gli endpoint nuovi o esistenti seguendo le istruzioni in Aggiungi endpoint ad Amazon Pinpoint per un singolo endpoint o in Aggiungi un batch di endpoint ad Amazon Pinpoint per più endpoint.
- Scelta del canale corretto prima di inviare un messaggio: se intendi inviare un messaggio SMS, ma ti accorgi che il numero di destinazione non è valido, puoi inviare un messaggio al destinatario tramite un canale diverso. Ad esempio, se l'endpoint non è in grado di ricevere messaggi SMS, puoi inviare un messaggio vocale.

Convalida un numero di telefono utilizzando il AWS CLI

L'esempio seguente mostra come convalidare un numero di telefono utilizzando il. AWS CLI Per ulteriori informazioni, consulta la sezione phone-number-validate nella Documentazione di riferimento della AWS CLI . Ad esempio, risposte di convalida, vedere. Risposta di convalida del numero di telefono Per ulteriori informazioni sulla configurazione di AWS CLI, vedere Configure the AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il servizio di convalida del numero di telefono utilizzando il AWS CLI

- Nella riga di comando, inserisci il comando seguente:

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

Nel comando precedente, sostituiscilo *+442079460881* con il numero di telefono che desideri convalidare e *GB* con il codice ISO a due cifre del paese o della regione.

> **Note**
>
> Quando si fornisce un numero di telefono al servizio di convalida del numero di telefono, includere sempre il prefisso internazionale. Se non si include il prefisso internazionale, il servizio può restituire informazioni per un numero di telefono in un paese diverso. Ad esempio, puoi avere dei trattini nel numero di telefono. *+44-207-946-0881*

Risposta di convalida del numero di telefono

Le informazioni che il servizio di convalida dei numeri di telefono fornisce variano leggermente in base ai dati disponibili per il numero di telefono fornito dall'utente. Questa sezione contiene esempi di risposte che il servizio di convalida dei numeri di telefono restituisce.

> **Note**
>
> I dati forniti dal servizio di convalida dei numeri di telefono si basano su informazioni fornite dai provider di servizi di telecomunicazione e da altre entità in tutto il mondo. I provider in alcuni paesi potrebbero aggiornare queste informazioni meno frequentemente rispetto a provider di altri paesi. Ad esempio, se invii una richiesta di convalida di un numero di telefono cellulare e il numero fornito è stato trasferito da un operatore mobile a un altro, la risposta dal servizio di convalida dei numeri di telefono potrebbe includere il nome dell'operatore originale, rispetto a quello corrente.

Numeri di cellulare validi

Quando invii una richiesta al servizio di convalida dei numeri di telefono e il numero di telefono è un numero di telefono cellulare valido, restituisce informazioni simili al seguente esempio:

```
{
    "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"
    }
}
```

Numeri di telefono fisso validi

Se la richiesta contiene un numero di telefono fisso valido, il servizio di convalida dei numeri di telefono restituisce informazioni simili al seguente esempio:

```
{
    "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"
}
```

Numeri di telefono VoIP validi

Se la richiesta contiene un numero di telefono VoIP valido (Voice over Internet Protocol), il servizio di convalida dei numeri di telefono restituisce informazioni simili al seguente esempio:

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+441514960001",
        "CleansedPhoneNumberNational": "1514960001",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+441514960001",
        "PhoneType": "VOIP",
        "PhoneTypeCode": 2
    }
}
```

Numeri di telefono non validi

Se la richiesta contiene un numero di telefono non valido, il servizio di convalida dei numeri di telefono restituisce informazioni simili al seguente esempio:

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

Notare che la proprietà `PhoneType` in questo tipo di risposta indica che questo numero di telefono è `INVALID` e che non include informazioni sull'operatore o su un percorso associato al numero di telefono. Si consiglia di evitare l'invio di SMS o messaggi vocali ai numeri di telefono in cui `PhoneType` è `INVALID`, perché questi numeri non appartengono a destinatari effettivi.

Altri numeri di telefono

Occasionalmente, la risposta dal servizio di convalida dei numeri di telefono include un valore `PhoneType` di `OTHER`. Il servizio può restituire questo tipo di risposta nelle seguenti situazioni:

- Il numero di telefono è un numero gratuito.
- Il numero di telefono è riservato all'utilizzo in programmi televisivi e filmati, ad esempio numeri di telefono del Nord America che iniziano con 555.
- Il numero di telefono include un prefisso che non è attualmente in uso, ad esempio il prefisso 999 in America del Nord.
- Il numero di telefono è riservato a un altro scopo.

L'esempio seguente mostra la risposta che il servizio di convalida dei numeri di telefono fornisce quando la richiesta include un numero di telefono dell'America del Nord fittizio:

```
{
    "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"
    }
}
```

Numeri di telefono prepagati

Se la richiesta contiene un numero di telefono prepagato valido, il servizio di convalida dei numeri di telefono restituisce informazioni simili al seguente esempio:

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

Per ulteriori informazioni sulle informazioni contenute in queste risposte, consulta l'articolo relativo alla convalida dei numeri di telefono nella documentazione di riferimento dell'API Amazon Pinpoint.

Crea un canale personalizzato in Amazon Pinpoint utilizzando un webhook o una funzione Lambda

Amazon Pinpoint include il supporto integrato per l'invio di messaggi tramite notifiche push, e-mail, SMS e canali vocali. È inoltre possibile configurare Amazon Pinpoint per l'invio di messaggi tramite altri canali creando canali personalizzati. I canali personalizzati in Amazon Pinpoint consentono di inviare messaggi tramite qualsiasi servizio dotato di un'API, inclusi i servizi di terze parti. Puoi interagire con APIs usando un webhook o chiamando una AWS Lambda funzione.

I segmenti a cui si inviano campagne del canale personalizzato possono contenere endpoint di tutti i tipi, ovvero endpoint in cui il valore dell'attributo `ChannelType` è EMAIL, VOICE, SMS, CUSTOM o uno dei vari tipi di endpoint delle notifiche push.

Usa un webhook

Se utilizzi un webhook per inviare messaggi di canale personalizzati, l'URL del webhook deve iniziare con "https://". The webhook URL can only contain alphanumeric characters, plus the following symbols: hyphen (-), period (.), underscore (_), tilde (~), question mark (?), slash or solidus (/), pound or hash sign (#), and semicolon (:). The URL has to comply with. RFC3986

Quando si crea una campagna che specifica un URL webhook, Amazon Pinpoint invia un metodo HTTP `HEAD` a tale URL. La risposta alla richiesta `HEAD` deve contenere un'intestazione chiamata `X-Amz-Pinpoint-AccountId`. Il valore di questa intestazione deve essere uguale all'ID del tuo AWS account.

Utilizza una funzione Lambda

Se scegli invece di inviare messaggi di canale personalizzati creando una funzione Lambda, è meglio prima acquisire familiarità con i dati emessi da Amazon Pinpoint. Quando una campagna Amazon Pinpoint invia messaggi su un canale personalizzato, invia un payload alla funzione Lambda di destinazione come nell'esempio seguente:

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

I dati sugli eventi forniscono gli attributi seguenti:

- `ApplicationId`; ID del progetto Amazon Pinpoint a cui appartiene la campagna.
- `CampaignId`: ID della campagna Amazon Pinpoint che ha richiamato la funzione Lambda.
- `TreatmentId`: ID della variante della campagna. Se hai creato una campagna standard, questo valore è sempre 0. Se è stata creata una campagna di test A/B, questo valore è un numero intero compreso tra 0 e 4.
- `ActivityId`: ID dell'attività eseguita dalla campagna.
- `ScheduledTime`: ora in cui Amazon Pinpoint ha eseguito la campagna, espressa in formato ISO 8601.

- `Endpoints`: elenco degli endpoint interessati dalla campagna. Ogni payload può contenere fino a 50 endpoint. Se il segmento a cui è stata inviata la campagna contiene più di 50 endpoint, Amazon Pinpoint richiama più volte la funzione, con un massimo di 50 endpoint alla volta, finché non vengono elaborati tutti gli endpoint.

È possibile utilizzare questi dati di esempio durante la creazione e il test della funzione Lambda del canale personalizzato.

Assegna una funzione Lambda o un webhook a una singola campagna utilizzando l'API Amazon Pinpoint

Per assegnare una funzione Lambda o un webhook a una campagna specifica, usa l'API Amazon Pinpoint per creare o aggiornare un oggetto Campaign.

L'oggetto `MessageConfiguration` della campagna deve contenere anche un oggetto `CustomMessage`. Questo oggetto ha un membro:. `Data` Il valore di `Data` è una stringa JSON che contiene il payload del messaggio che si desidera inviare al canale personalizzato.

La campagna deve contenere un oggetto `CustomDeliveryConfiguration`. All'interno dell'oggetto `CustomDeliveryConfiguration` specificare quanto segue:

- `EndpointTypes`: array contenente tutti i tipi di endpoint a cui deve essere inviata la campagna personalizzata mediante il canale personalizzato. Può contenere uno o tutti i seguenti tipi di canale:
  - `ADM`
  - `APNS`
  - `APNS_SANDBOX`
  - `APNS_VOIP`
  - `APNS_VOIP_SANDBOX`
  - `BAIDU`
  - `CUSTOM`
  - `EMAIL`
  - `GCM`
  - `SMS`
  - `VOICE`
- `DeliveryUri`: destinazione a cui vengono inviati gli endpoint. Puoi specificare uno dei seguenti:

- L'URL del webhook a cui inviare i dati dell'endpoint durante l'esecuzione della campagna.
- Il nome della risorsa Amazon (ARN) di una funzione Lambda che desideri eseguire durante l'esecuzione della campagna.

> **Note**
>
> L'oggetto `Campaign` può anche contenere un oggetto `Hook`. Questo oggetto viene utilizzato solo per creare segmenti personalizzati da una funzione Lambda durante l'esecuzione di una campagna. Per ulteriori informazioni, consulta Personalizza i segmenti di Amazon Pinpoint utilizzando una funzione AWS Lambda.

Crea e configura una funzione Lambda per una campagna Amazon Pinpoint

Questa sezione fornisce una panoramica dei passaggi per creare una funzione Lambda che invia messaggi su un canale personalizzato. Innanzitutto, creare la funzione. Quindi, si aggiunge una politica di esecuzione alla funzione. Questa policy consente ad Amazon Pinpoint di eseguire la policy durante l'esecuzione di una campagna.

Per un'introduzione alla creazione di funzioni Lambda, consulta l'argomento relativo alla creazione di funzioni Lambda nella Guida per gli sviluppatori di AWS Lambda .

Funzione Lambda di esempio

Il seguente esempio di codice elabora il payload e registra il numero di endpoint di ciascun tipo di endpoint. 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)
```

Quando una campagna Amazon Pinpoint esegue questa funzione Lambda, Amazon Pinpoint invia alla funzione un elenco di membri del segmento. La funzione conta il numero di endpoint di ogni `ChannelType`. Quindi invia tali dati ad Amazon CloudWatch. Puoi visualizzare queste metriche nella sezione Metriche della CloudWatch console. Le metriche sono disponibili nel namespace. PinpointCustomChannelExecution

È possibile modificare questo esempio di codice in modo che si connetta anche all'API di un servizio esterno per inviare messaggi tramite tale servizio.

Formato di risposta della funzione Lambda per Amazon Pinpoint

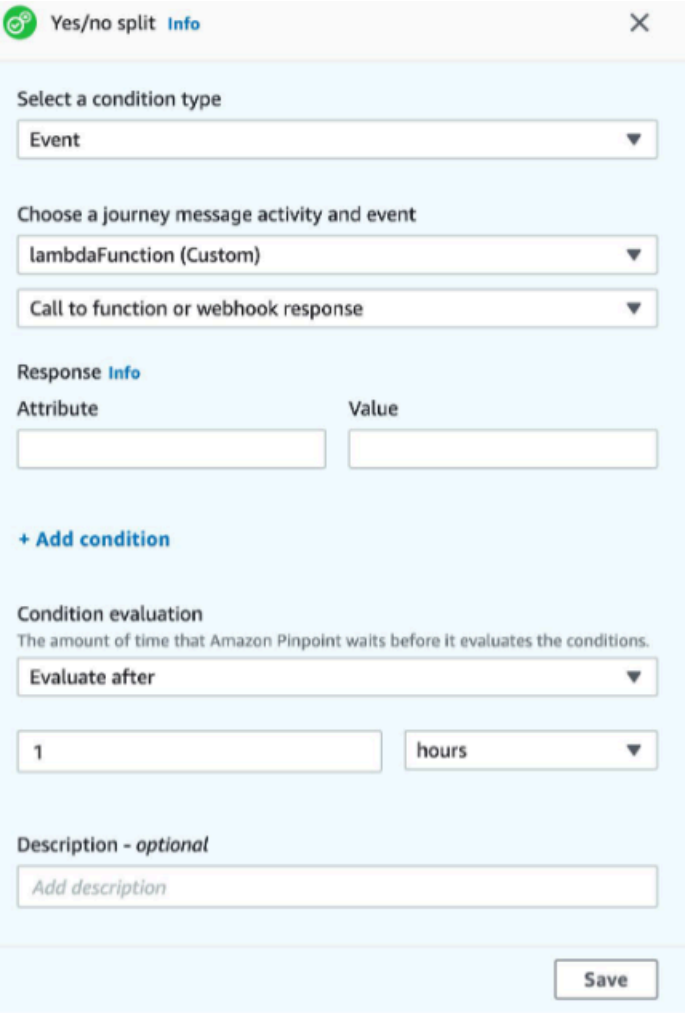
Se desideri utilizzare la suddivisione multivariata del percorso o sì/no per determinare il percorso dell'endpoint dopo un'attività di canale personalizzata, devi strutturare la risposta della funzione

Lambda in un formato comprensibile ad Amazon Pinpoint e quindi inviare gli endpoint lungo il percorso corretto.

La struttura della risposta deve avere il formato seguente:

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

Ciò ti consentirà quindi di selezionare una chiave e un valore per determinare il percorso degli endpoint.



Concedi ad Amazon Pinpoint l'autorizzazione a richiamare la funzione Lambda

Puoi usare AWS Command Line Interface (AWS CLI) per aggiungere autorizzazioni alla politica della funzione Lambda assegnata alla tua funzione Lambda. Per consentire ad Amazon Pinpoint di richiamare una funzione, usa il comando add-permission, come mostrato nell'esempio seguente:

```
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
```

Nel comando precedente, procedi come segue.

- Sostituisci *myFunction* con il nome della funzione Lambda.
- Sostituisci *us-east-1* con la AWS regione in cui utilizzi Amazon Pinpoint.
- Sostituisci *111122223333* con il tuo ID account AWS .

Quando esegui il comando `add-permission`, la funzione Lambda restituisce l'output seguente:

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

Il valore di `Statement` è una versione in formato stringa JSON dell'istruzione aggiunta alla policy di funzione Lambda.

Ulteriore limitazione della policy di esecuzione

È possibile modificare la policy di esecuzione limitandola a un progetto Amazon Pinpoint specifico. A tale scopo, sostituire * nell'esempio precedente con l'ID univoco del progetto. È possibile limitare ulteriormente la policy limitandola a una campagna specifica. Ad esempio, per limitare la policy in modo che il consenta solo una campagna con l'ID `95fee4cd1d7f5cd67987c1436example` in un progetto con l'ID `dbaf6ec2226f0a9a8615e3ea5example`, utilizzare il seguente valore per l'attributo `source-arn`:

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

> **Note**
>
> Se si limita l'esecuzione della funzione Lambda a una campagna specifica, è necessario innanzitutto creare la funzione con una policy meno restrittiva. Successivamente, è necessario creare la campagna in Amazon Pinpoint e scegliere la funzione. Infine, è necessario aggiornare la policy di esecuzione per fare riferimento alla campagna specificata.

Trasmetti in streaming i dati degli eventi delle app tramite Kinesis e Firehose utilizzando Amazon Pinpoint

In Amazon Pinpoint, un evento è un'azione che si verifica quando un utente interagisce con una delle tue applicazioni, quando invii un messaggio da una campagna o un percorso oppure quando invii un messaggio transazionale SMS o e-mail. Ad esempio, se invii un messaggio e-mail, si verificano diversi eventi:

- Quando invii il messaggio, si verifica un evento di invio .
- Quando il messaggio raggiunge la casella di posta in arrivo del destinatario, si verifica un evento recapitato .
- Quando il destinatario apre il messaggio, si verifica un evento di apertura .

È possibile configurare Amazon Pinpoint in modo che invii informazioni sugli eventi ad Amazon Kinesis. La piattaforma Kinesis offre servizi che puoi utilizzare per raccogliere, elaborare e analizzare i dati dei AWS servizi in tempo reale. Amazon Pinpoint può inviare dati sugli eventi a Firehose, che li trasmette ad archivi AWS dati come Amazon S3 o Amazon Redshift. Amazon Pinpoint può anche trasmettere dati al flusso di dati Kinesis, che acquisisce e archivia più flussi di dati per l'elaborazione da parte delle applicazioni di analisi.

Il flusso di eventi Amazon Pinpoint include informazioni sulle interazioni degli utenti con le applicazioni (app) connesse ad Amazon Pinpoint. Include anche informazioni su tutti i messaggi inviati dalle campagne, attraverso qualsiasi canale e dai viaggi. Ciò può includere anche tutti gli eventi personalizzati precedentemente definiti. Infine, include informazioni sulle tutte le e-mail transazionali e su tutti i messaggi SMS inviati.

> **Note**
>
> Amazon Pinpoint non esegue lo streaming di informazioni sulle notifiche push transazionali o sui messaggi vocali.

In questo capitolo vengono fornite informazioni sulla configurazione di Amazon Pinpoint per lo streaming dei dati degli eventi in Kinesis. Contiene anche esempi di dati di eventi trasmessi in streaming da Amazon Pinpoint.

Argomenti

- Configura Amazon Pinpoint per lo streaming dei dati degli eventi delle app tramite Amazon Kinesis o Amazon Data Firehose
- Flusso di dati sugli eventi delle app da Amazon Pinpoint
- Flusso di dati sugli eventi della campagna da Amazon Pinpoint
- Dati sugli eventi di viaggio da Amazon Pinpoint
- Flusso di dati sugli eventi di posta elettronica da Amazon Pinpoint
- Flusso di dati di eventi SMS da Amazon Pinpoint
- Eliminare un flusso di eventi da Amazon Pinpoint

Configura Amazon Pinpoint per lo streaming dei dati degli eventi delle app tramite Amazon Kinesis o Amazon Data Firehose

Puoi configurare Amazon Pinpoint per inviare i dati degli eventi a uno stream Amazon Kinesis o a un flusso di distribuzione Amazon Data Firehose. Amazon Pinpoint può inviare dati sugli eventi per campagne, percorsi e messaggi e-mail e SMS transazionali.

Questa sezione include informazioni sulla configurazione dello streaming di eventi a livello di programmazione. Puoi anche utilizzare la console Amazon Pinpoint per configurare lo streaming di eventi. Per informazioni sulla configurazione dello streaming di eventi utilizzando la console Amazon Pinpoint, consulta Impostazioni del flusso di eventi nella Guida per l'utente di Amazon Pinpoint.

Prerequisiti

Gli esempi in questa sezione richiedono il seguente input:

- L'ID di un'applicazione integrata con Amazon Pinpoint e che segnala gli eventi. Per informazioni sull'integrazione, consulta Integra Amazon Pinpoint con la tua applicazione.
- L'Amazon Resource Name (ARN) di uno stream Kinesis o di un flusso di distribuzione Firehose nel tuo account. AWS Per informazioni sulla creazione di queste risorse, consulta Creating and Managing Streams nella Amazon Kinesis Data Streams Developer Guide o Creazione di un flusso di distribuzione Amazon Data Firehose nella Amazon Data Firehose Developer Guide.
- L'ARN di un ruolo AWS Identity and Access Management (IAM) che autorizza Amazon Pinpoint a inviare dati allo stream. Per informazioni su come creare un ruolo, consulta Ruolo IAM per lo streaming di eventi su Kinesis.

AWS CLI

L' AWS CLI esempio seguente utilizza il comando. put-event-stream Questo comando configura Amazon Pinpoint per l'invio di eventi a un flusso Kinesis:

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

AWS SDK per Java

L'esempio Java seguente configura Amazon Pinpoint per l'invio di eventi a un flusso 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);
}
```

Questo esempio costruisce un `WriteEventStream` oggetto che memorizza il ARNs flusso Kinesis e il ruolo IAM. L'oggetto `WriteEventStream` viene passato a un oggetto `PutEventStreamRequest` per configurare Amazon Pinpoint per lo streaming degli eventi per un'applicazione specifica. L'oggetto `PutEventStreamRequest` viene passato al metodo `putEventStream` del client Amazon Pinpoint.

Puoi assegnare un flusso Kinesis a più applicazioni. In questo caso, Amazon Pinpoint invia i dati degli eventi con codifica Base64 da ogni applicazione al flusso e ciò consente di analizzare i dati come una raccolta. Il seguente metodo di esempio accetta un elenco di applicazioni (app) IDs e utilizza il metodo di esempio precedente per assegnare uno stream a ciascuna applicazione: `createEventStream`

```
public List<PutEventStreamResult> createEventStreamFromAppList(
        AmazonPinpoint pinClient, List<String> appIDs,
```

```
        String streamArn, String roleArn) {

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

Sebbene sia possibile assegnare un flusso a più applicazioni, non è possibile assegnare più flussi a una sola applicazione.

Flusso di dati sugli eventi delle app da Amazon Pinpoint

Dopo aver integrato la tua applicazione (app) con Amazon Pinpoint e configurato lo streaming degli eventi, Amazon Pinpoint recupera l'attività utente dell'app, gli eventi personalizzati e i dati di recapito dei messaggi dalla destinazione specificata durante la configurazione per consentirti di visualizzarli. Per informazioni su come configurare lo streaming degli eventi in modo da poter visualizzare i dati degli eventi, consulta. Configura Amazon Pinpoint per lo streaming dei dati degli eventi delle app tramite Amazon Kinesis o Amazon Data Firehose

Esempio di evento dell'app

L'oggetto JSON per un evento app contiene i dati mostrati nell'esempio seguente.

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

Attributi di eventi dell’app

Questa sezione definisce gli attributi inclusi nell'esempio precedente del flusso di eventi dell'app.

Attributo	Descrizione
`event_type`	Tipo di evento. I valori possibili sono: • _session.start: l'endpoint ha iniziato una nuova sessione. • _session.stop: l'endpoint ha terminato una sessione. • _userauth.sign_in: l'endpoint ha stabilito la connessione all'app.

Attributo	Descrizione
	• _userauth.sign_up: un nuovo endpoint ha completato la procedura di registrazione nell'app. • _userauth.auth_fail: l'endpoint ha tentato di accedere all'app, ma non è stato in grado di completare il processo. • _monetization.purchase: l'endpoint ha effettuato un acquisto nell'app. • _session.pause: la sessione dell'endpoint è stata sospesa. Le sessioni in pausa possono essere riavviate in modo da poter continuar e a raccogliere i parametri senza avviare una sessione completamente nuova. • _session.resume: la sessione dell'endpoint è stata ripresa.
`event_timestamp`	L'ora in cui l'evento è stato segnalato, espressa nel formato orario Unix in millisecondi.
`arrival_timestamp`	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.
`event_version`	La versione dello schema JSON dell'evento. **Tip** Seleziona questa versione nell'appl icazione di elaborazione eventi in modo da sapere quando aggiornare l'applica zione in risposta a un aggiornamento dello schema.

Attributo	Descrizione
`application`	Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella Applicazione.
`client`	Informazioni sull'endpoint che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella Client.
`device`	Informazioni sul dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella Dispositivo.
`session`	Informazioni sulla sessione che ha generato l'evento. Per ulteriori informazioni, consulta la tabella Sessione.
`attributes`	Attributi associati all'evento. Per gli eventi segnalati dalle app, questo oggetto include gli attributi personalizzati definiti dall'utente.
`metrics`	Parametri correlati all'evento. Facoltativamente, puoi configurare le app per l'invio di metriche personalizzate ad Amazon Pinpoint.

Applicazione

Include informazioni sul progetto Amazon Pinpoint a cui è associato l'evento.

Attributo	Descrizione
`app_id`	ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.
`cognito_identity_pool_id`	ID del pool di identità Amazon Cognito a cui è associato l'endpoint.

Attributo	Descrizione
`package_name`	Il nome del pacchetto dell'app, ad esempio `com.example.my_app` .
`sdk`	Informazioni sull'SDK utilizzato per segnalare l'evento. Per ulteriori informazioni, consulta la tabella SDK.
`title`	Il nome dell'app.
`version_name`	Il nome della versione dell'app, ad esempio `V2.5`.
`version_code`	Il numero di versione dell'app, ad esempio `3`.

SDK

Include informazioni sull'SDK utilizzato per segnalare l'evento.

Attributo	Descrizione
`name`	Il nome dell'SDK utilizzato per segnalare l'evento.
`version`	La versione dell'SDK.

Client

Include informazioni sull'endpoint che ha generato l'evento.

Attributo	Descrizione
`client_id`	ID dell'endpoint.
`cognito_id`	Token ID Amazon Cognito associato all'endpoint.

Dispositivo

Include informazioni sul dispositivo dell'endpoint che ha generato l'evento.

Attributo	Descrizione
`locale`	Informazioni sulle impostazioni di lingua e regione per il dispositivo dell'endpoint. Per ulteriori informazioni, consulta la tabella Locale.
`make`	Il produttore del dispositivo dell'endpoint.
`model`	L'identificatore del modello del dispositivo dell'endpoint.
`platform`	Informazioni sul sistema operativo del dispositivo dell'endpoint. Per ulteriori informazioni, consulta la tabella Piattaforma.

Locale

Include informazioni sulle impostazioni di lingua e regione per il dispositivo dell'endpoint.

Attributo	Descrizione
`code`	L'identificatore locale associato al dispositivo.
`country`	Il paese o la regione associata alle impostazioni locali del dispositivo.
`language`	La lingua associata alle impostazioni locali del dispositivo.

Piattaforma

Include informazioni sul sistema operativo del dispositivo dell'endpoint.

Attributo	Descrizione
`name`	Il nome del sistema operativo sul dispositivo.
`version`	La versione del sistema operativo sul dispositivo.

Sessione

Include informazioni sulla sessione che ha generato l'evento.

Attributo	Descrizione
`session_id`	Un ID univoco che identifica la sessione.
`start_timestamp`	La data e l'ora in cui è iniziata la sessione, espresse in formato orario Unix in millisecondi.
`stop_timestamp`	La data e l'ora in cui è terminata la sessione, espresse in formato orario Unix in millisecondi.

Flusso di dati sugli eventi della campagna da Amazon Pinpoint

Se utilizzi Amazon Pinpoint per inviare campagne tramite un canale, Amazon Pinpoint può trasmettere in streaming i dati sugli eventi relativi a tali campagne. Dopo aver configurato lo streaming degli eventi, Amazon Pinpoint recupera i dati degli eventi dell'app per e-mail o SMS che invii da una campagna dalla destinazione specificata durante la configurazione per consentirti di visualizzarli. Per informazioni dettagliate sui dati trasmessi da Amazon Pinpoint per e-mail e SMS, consulta e. the section called “Flusso di dati sugli eventi di posta elettronica da Amazon Pinpoint” the section called “Flusso di dati di eventi SMS da Amazon Pinpoint” Per informazioni su come configurare lo streaming di eventi, consulta. Configura Amazon Pinpoint per lo streaming dei dati degli eventi delle app tramite Amazon Kinesis o Amazon Data Firehose

Esempio di evento della campagna

L'oggetto JSON per un evento della campagna contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_campaign.send",
  "event_timestamp": 1562109497426,
  "arrival_timestamp": 1562109497494,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "d8dcf7c5-e81a-48ae-8313-f540cexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "treatment_id": "0",
    "campaign_activity_id": "5473285727f04865bc673e527example",
    "delivery_type": "GCM",
    "campaign_id": "4f8d6097c2e8400fa3081d875example",
    "campaign_send_status": "SUCCESS"
  },
  "client_context": {
    "custom": {
      "endpoint": "{\"ChannelType\":\"GCM\",\"EndpointStatus\":\"ACTIVE\",
          #\"OptOut\":\"NONE\",\"RequestId\":\"ec229696-9d1e-11e9-8bf1-85d0aexample\",
          #\"EffectiveDate\":\"2019-07-02T23:12:54.836Z\",\"User\":{}}"
    }
  },
  "awsAccountId": "123456789012"
}
```

Attributi degli eventi di campagna

Questa sezione definisce gli attributi inclusi nel flusso di eventi della campagna.

Attributo	Descrizione
`event_type`	Tipo di evento. I valori possibili sono:

Attributo	Descrizione
	• _campaign.send: Amazon Pinpoint ha eseguito la campagna. • _campaign.opened_notification: per le campagne basate sulle notifiche push, questo tipo di evento indica che il destinatario ha toccato la notifica per aprirla. • _campaign.received_foreground: per le campagne basate sulle notifiche push, questo tipo di evento indica che il destinatario ha ricevuto il messaggio come notifica in primo piano. • _campaign.received_background: per le campagne basate sulle notifiche push, questo tipo di evento indica che il destinatario ha ricevuto il messaggio come notifica in background. **Note** _campaign.opened_notification, _campaign.received_foreground e _campaign.received_background vengono restituiti solo se utilizzi AWS Amplify. Per ulteriori informazioni sull'integrazione della tua app con. AWS Amplify Per informazioni, consulta Connect l'applicazione frontend ad Amazon AWS Pinpoint utilizzando Amplify.
`event_timestamp`	L'ora in cui l'evento è stato segnalato, espressa nel formato orario Unix in millisecondi.

Attributo	Descrizione
`arrival_timestamp`	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.
`event_version`	La versione dello schema JSON dell'evento. **Tip** Seleziona questa versione nell'appl icazione di elaborazione eventi in modo da sapere quando aggiornare l'applica zione in risposta a un aggiornamento dello schema.
`application`	Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella Applicazione.
`client`	Informazioni sull'endpoint a cui è associato l'evento. Per ulteriori informazioni, consulta la tabella Client.
`device`	Informazioni sul dispositivo che ha segnalato l'evento. Per i messaggi di campagna e transazionali, questo oggetto è vuoto.
`session`	Informazioni sulla sessione che ha generato l'evento. Per le campagne, questo oggetto è vuoto.

Attributo	Descrizione
`attributes`	Attributi associati all'evento. Per gli eventi segnalati da una delle app, questo oggetto può includere attributi personalizzati definiti dall'app. Per gli eventi che vengono creati quando si invia una campagna, questo oggetto contiene gli attributi associati alla campagna. Per gli eventi che vengono generati quando si inviano messaggi transazionali, questo oggetto contiene informazioni correlate al messaggio stesso. Per ulteriori informazioni, consulta la tabella Attributi.
`client_context`	Contiene un oggetto `custom`, che contiene una proprietà `endpoint`. La proprietà `endpoint` contiene i contenuti del record di endpoint per l'endpoint a cui è stata inviata la campagna.
`awsAccountId`	L'ID dell' AWS account utilizzato per inviare il messaggio.

Applicazione

Include informazioni sul progetto Amazon Pinpoint a cui è associato l'evento.

Attributo	Descrizione
`app_id`	ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.
`sdk`	L'SDK utilizzato per segnalare l'evento.

Attributes

Include informazioni sulla campagna che ha prodotto l'evento.

Attributo	Descrizione
`treatment_id`	Se il messaggio è stato inviato utilizzando una campagna di test A/B, questo valore rappresenta il numero della descrizione (treatment) del messaggio. Per le campagne standard, questo valore è `0`.
`campaign_activity_id`	ID univoco generato da Amazon Pinpoint quando si verifica l'evento.
`delivery_type`	Metodo di recapito della campagna. Non confondere questo attributo con il campo `ChannelType` specificato sotto la proprietà `endpoint` di `client_context`. Il campo `ChannelType` è in genere basato sull'endpoint a cui viene inviato il messaggio. Per i canali che supportano un solo tipo di endpoint, i campi `delivery_type` e `ChannelType` hanno lo stesso valore. Ad esempio, per il canale e-mail, i campi `delivery_type` e `ChannelType` hanno lo stesso valore di EMAIL. Tuttavia, questa condizione non è sempre vera per i canali che supportano diversi tipi di endpoint, come i canali personalizzati. Puoi utilizzare un canale personalizzato per diversi endpoint, come EMAIL, SMS, CUSTOM e così via. In questo caso, `delivery_type` identifica un evento di recapito personalizzato (CUSTOM) e `ChannelType` specifica il tipo di endpoint a cui è stata inviata la campagna,

Attributo	Descrizione
	ad esempio EMAIL, SMS, CUSTOM e così via. Per ulteriori informazioni sulla creazione di canali personalizzati, consulta Crea un canale personalizzato. I valori possibili sono: • EMAIL • SMS • ADM • APNS • APNS_SANDBOX • APNS_VOIP • APNS_VOIP_SANDBOX • VOICE • GCM • BAIDU • PUSH • CUSTOM
`campaign_id`	L'ID univoco della campagna da cui è stato inviato il messaggio.

Attributo	Descrizione
`campaign_send_status`	Indica lo stato della campagna per l'endpoint di destinazione. I valori possibili includono: • SUCCESS: invio della campagna all'endpoint riuscito. • FAILURE: invio della campagna all'endpoint non riuscito. • DAILY_CAP: la campagna non è stata inviata all'endpoint perché all'endpoint è già stato inviato il numero massimo di messaggi giornalieri. • EXPIRED: la campagna non è stata inviata all'endpoint perché l'invio ha superato la durata massima o la velocità di invio impostata per la campagna. • QUIET_TIME: la campagna non è stata inviata all'endpoint a causa delle limitazioni relative al periodo di silenzio. • HOLDOUT: la campagna non è stata inviata all’endpoint perché l'endpoint è un membro del gruppo di esclusione. • DUPLICATE_ADDRESS: nel segmento sono presenti indirizzi di endpoint duplicati. La campagna è stata inviata una volta all'indir izzo dell'endpoint. • QUIET_TIME: la campagna non è stata inviata all'endpoint a causa delle limitazioni relative al periodo di silenzio. • CAMPAIGN_CAP: la campagna non è stata inviata all'endpoint perché all'endpoint è già stato inviato il numero massimo di messaggi dalla campagna in corso.

Attributo	Descrizione
	• FAILURE_PERMANENT: si è verificato un errore permanente durante l'invio all'endpo int. • TRANSIENT_FAILURE: si è verificato un errore temporaneo durante l'invio all'endpo int. • THROTTLED: l'invio è stato limitato a livello di larghezza di banda della rete. • UNKNOWN: errore sconosciuto. • HOOK_FAILURE: hook della campagna non riuscito. • CUSTOM_DELIVERY_FAILURE: recapito personalizzato non riuscito. • RECOMMENDATION_FAILURE: raccomand azione non riuscita. • UNSUPPORTED_CHANNEL: canale non supportato.

Client

Include informazioni sull'endpoint a cui era destinata la campagna.

Attributo	Descrizione
`client_id`	L'ID dell'endpoint a cui è stata inviata la campagna.

Dati sugli eventi di viaggio da Amazon Pinpoint

Quando pubblichi un percorso, Amazon Pinpoint può trasmettere in streaming i dati degli eventi per e-mail, SMS, push e messaggi personalizzati che invii dal percorso. Dopo aver configurato lo streaming degli eventi, Amazon Pinpoint recupera i dati dalla destinazione specificata durante

la configurazione per consentirti di visualizzarli. Per informazioni dettagliate sui dati trasmessi da Amazon Pinpoint per e-mail e SMS, consulta e. the section called “Flusso di dati sugli eventi di posta elettronica da Amazon Pinpoint” the section called “Flusso di dati di eventi SMS da Amazon Pinpoint” Per informazioni su come configurare lo streaming di eventi, consulta. Configura Amazon Pinpoint per lo streaming dei dati degli eventi delle app tramite Amazon Kinesis o Amazon Data Firehose

Esempio di evento di viaggio

L'oggetto JSON per un evento di viaggio contiene i dati mostrati nell'esempio seguente.

```
{
   "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"
}
```

Attributi di eventi di viaggio

Questa sezione definisce gli attributi inclusi nei dati del flusso di eventi generati da Amazon Pinpoint per un percorso.

Attributo	Descrizione
`event_type`	Tipo di evento. Per gli eventi dei percorsi, il valore per questo attributo è sempre `_journey.send` , che indica che Amazon Pinpoint ha eseguito il percorso.
`event_timestamp`	L'ora in cui l'evento è stato segnalato, espressa nel formato orario Unix in millisecondi.
`arrival_timestamp`	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.
`event_version`	La versione dello schema JSON dell'evento. **Tip** Seleziona questa versione nell'appl icazione di elaborazione eventi in modo da sapere quando aggiornare l'applica zione in risposta a un aggiornamento dello schema.

Attributo	Descrizione
`application`	Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella Applicazione.
`client`	Informazioni sull'endpoint associato all'event o. Per ulteriori informazioni, consulta la tabella Client.
`device`	Informazioni sul dispositivo che ha segnalato l'evento. Per i viaggi, questo oggetto è vuoto.
`session`	Informazioni sulla sessione che ha generato l'evento. Per i viaggi, questo oggetto è vuoto.
`attributes`	Attributi associati al viaggio e all'attività di viaggio che ha generato l'evento. Per ulteriori informazioni, consulta la tabella Attributi.
`client_context`	Contiene un oggetto `custom`, che contiene una proprietà `endpoint`. La proprietà `endpoint` contiene i contenuti del record di endpoint per l'endpoint associato all'evento.
`awsAccountId`	L'ID dell' AWS account utilizzato per eseguire il viaggio.

Applicazione

Include le informazioni sul progetto Amazon Pinpoint a cui è associato l'evento.

Attributo	Descrizione
`app_id`	ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.
`sdk`	L'SDK utilizzato per segnalare l'evento.

Client

Include informazioni sull'endpoint associato all'evento.

Attributo	Descrizione
`client_id`	ID dell'endpoint.

Attributes

Include informazioni sul viaggio che ha generato l'evento.

Attributo	Descrizione
`journey_run_id`	ID univoco del percorso che ha generato l'evento. Amazon Pinpoint genera e assegna automaticamente questo ID a ogni nuova esecuzione di un percorso.
`journey_send_status`	Indica lo stato di consegna del messaggio associato all'evento. I valori possibili includono: • SUCCESS: la campagna è stata inviata correttamente all'endpoint. • FAILURE: il messaggio non è stato inviato all'endpoint a causa di un errore. • CUSTOM_DELIVERY_FAILURE: recapito personalizzato non riuscito. • FAILURE_PERMANENT: si è verificato un errore permanente durante l'invio all'endpo int. **Tip** È possibile filtrare gli eventi con stato FAILURE_PERMANENT e `journey_send_status_code`

Attributo	Descrizione
	impostarli su 403 per determina re se vi è una violazione della politica di accesso e del ruolo. Per le campagne vocali in uscita, queste eccezioni sono tipiche dei casi in cui il ruolo di esecuzione della campagna Connect che collega i viaggi di Amazon Pinpoint alle campagne Amazon Connect viene inavverti tamente eliminato per le esecuzioni di viaggio in corso. • THROTTLED: l'invio è stato limitato a livello di larghezza di banda della rete. • UNSUPPORTED_CHANNEL: canale non supportato. • DAILY_CAP: il messaggio non è stato inviato all'endpoint in quanto l'invio del messaggio supera il numero massimo di messaggi che il percorso o il progetto può inviare a un singolo endpoint in un intervallo di 24 ore. • QUIET_TIME: il messaggio non è stato inviato a causa delle limitazioni per l'intervallo di silenzio per il percorso o progetto. • QUIET_TIME_MISSING_TIMEZONE: il messaggio non è stato inviato perché la stima del fuso orario non è in grado di stimare un fuso orario per l'endpoint e il periodo di silenzio è abilitato.
`journey_id`	L'ID univoco del viaggio che ha generato l'evento.

Attributo	Descrizione
`journey_activity_id`	L'ID univoco dell'attività di viaggio che ha generato l'evento.
`journey_activity_type`	Il tipo di attività di percorso dell'evento. Può essere EMAIL, SMS, PUSH, CONTACT_CENTER o CUSTOM. **Note** VOICE non è un tipo di attività di percorso supportato. Il **`journey_activity_type`** campo non è presente quando è impostato su QUIET_TIME_WAIT_FINISHED. **`journey_send_status`**
`journey_send_status_message`	La descrizione dello stato dell'evento di invio.
`journey_send_status_code`	Il codice di stato HTTP della richiesta.

Flusso di dati sugli eventi di posta elettronica da Amazon Pinpoint

Se utilizzi Amazon Pinpoint per inviare e-mail, Amazon Pinpoint può trasmettere in streaming i dati sugli eventi relativi a tali e-mail. Dopo aver configurato lo streaming degli eventi, Amazon Pinpoint recupera i dati degli eventi dalla destinazione specificata durante la configurazione per consentirti di visualizzarli. Per informazioni su come configurare lo streaming degli eventi, consulta. Configura Amazon Pinpoint per lo streaming dei dati degli eventi delle app tramite Amazon Kinesis o Amazon Data Firehose Amazon Pinpoint trasmette dati sui seguenti tipi di eventi per i messaggi e-mail:

- Invii
- Consegne
- Mancati recapiti
- Reclami
- Aperture

- Clic
- Rifiuti
- Annullamenti di sottoscrizioni
- Errori di rendering

Questi tipi di eventi sono spiegati in dettaglio in. Attributi di eventi e-mail

A seconda dell'API e delle impostazioni utilizzate per inviare messaggi e-mail, è possibile che vengano visualizzati altri tipi di eventi o dati diversi. Ad esempio, se invii messaggi utilizzando set di configurazione che pubblicano i dati degli eventi in Amazon Kinesis, ad esempio quelli forniti da Amazon Simple Email Service (Amazon SES), i dati possono includere anche eventi per errori di rendering dei modelli. Per informazioni su questi dati, consulta Monitoraggio tramite la pubblicazione degli eventi Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service. Prima di poter visualizzare i tuoi eventi devi configurare lo streaming degli eventi, vedi Configura Amazon Pinpoint per lo streaming dei dati degli eventi delle app tramite Amazon Kinesis o Amazon Data Firehose . Quando configuri lo streaming degli eventi, specifichi una destinazione in cui salvare i dati dell'evento, quindi puoi utilizzare la destinazione per recuperare i dati dell'evento per la visualizzazione.

Esempi di eventi via e-mail

Invio di e-mail

L'oggetto JSON per un evento di invio e-mail contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_email.send",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622025,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "9a311b17-6f8e-4093-be61-4d0bbexample"
  },
  "device": {
    "platform": {}
  },
```

```
  "session": {},
  "attributes": {
    "feedback": "received"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "send": {}
      }
    }
  }
}
```

Consegna di e-mail

L'oggetto JSON per un evento di consegna e-mail contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_email.delivered",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622690,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "e9a3000d-daa2-40dc-ac47-1cd34example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "delivered"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
```

```
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "delivery": {
          "smtp_response": "250 ok:  Message 82080542 accepted",
          "reporting_mta": "a8-53.smtp-out.amazonses.com",
          "recipients": ["recipient@example.com"],
          "processing_time_millis": 1310
        }
      }
    }
  }
}
```

Clic su e-mail

L'oggetto JSON per un evento di clic su e-mail contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_email.click",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618713751,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "49c1413e-a69c-46dc-b1c4-6470eexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "https://aws.amazon.com/pinpoint/"
  },
  "awsAccountId": "123456789012",
```

```
 "facets": {
   "email_channel": {
     "mail_event": {
       "mail": {
         "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
         "message_send_timestamp": 1564618621380,
         "from_address": "sender@example.com",
         "destination": ["recipient@example.com"],
         "headers_truncated": false,
         "headers": [{
           "name": "From",
           "value": "sender@example.com"
         }, {
           "name": "To",
           "value": "recipient@example.com"
         }, {
           "name": "Subject",
           "value": "Amazon Pinpoint Test"
         }, {
           "name": "MIME-Version",
           "value": "1.0"
         }, {
           "name": "Content-Type",
           "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
         }, {
           "name": "Message-ID",
           "value": "null"
         }],
         "common_headers": {
           "from": "sender@example.com",
           "to": ["recipient@example.com"],
           "subject": "Amazon Pinpoint Test"
         }
       },
       "click": {
         "ip_address": "72.21.198.67",
         "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15",
         "link": "https://aws.amazon.com/pinpoint/"
       }
     }
   }
 }
```

```
}
```

Apertura di e-mail

L'oggetto JSON per un evento di apertura e-mail contiene i dati mostrati nell'esempio seguente.

```
{
  "event_type": "_email.open",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618712316,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "8dc1f651-b3ec-46fc-9b67-2a050example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "opened"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
```

```
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "open": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko)"
        }
      }
    }
  }
}
```

Attributi di eventi e-mail

Questa sezione definisce gli attributi inclusi nell'esempio precedente dei dati del flusso di eventi che Amazon Pinpoint genera quando invii messaggi e-mail.

Attributo	Descrizione
`event_type`	Tipo di evento. I valori possibili sono: • _email.send: Amazon Pinpoint ha accettato il messaggio e ha tentato di recapitarlo al destinatario. • _email.delivered: il messaggio è stato recapitato al destinatario.

Attributo	Descrizione
	• _email.rejected: Amazon Pinpoint ha stabilito che il messaggio contiene malware e non ha tentato di inviarlo. • _email.hardbounce: un problema permanente ha impedito ad Amazon Pinpoint di recapitar e il messaggio. Amazon Pinpoint non tenterà più di recapitare il messaggio. • _email.softbounce: un problema temporaneo ha impedito ad Amazon Pinpoint di recapitar e il messaggio. Amazon Pinpoint tenterà di recapitare nuovamente il messaggio per un certo periodo di tempo. Se il messaggio continua a non essere recapitato, non verrà effettuato alcun altro tentativo. Lo stato finale dell'e-mail sarà quindi SOFTBOUNCE. • _email.complaint: il destinatario ha ricevuto il messaggio e lo ha segnalato al proprio provider e-mail come spam (ad esempio, utilizzando la funzionalità "Segnala come spam" del client e-mail). • _email.open: il destinatario ha ricevuto il messaggio e lo ha aperto. • _email.click: il destinatario ha ricevuto il messaggio e ha fatto clic su un collegamento al suo interno. • _email.unsubscribe: il destinatario ha ricevuto il messaggio e ha fatto clic su un collegame nto per l'annullamento della sottoscrizione al suo interno. • _email.rendering_failure: l'e-mail non è stata inviata a causa di un errore di rendering. Ciò può verificarsi se i dati del modello mancano

Attributo	Descrizione
	o se non vi è corrispondenza tra i parametri e i dati del modello.
`event_timestamp`	L'ora in cui il messaggio è stato inviato, espressa nel formato orario Unix in millisecondi. Questo valore è in genere lo stesso per tutti gli eventi generati per un messaggio.
`arrival_timestamp`	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.
`event_version`	La versione dello schema JSON dell'evento. **Tip** Seleziona questa versione nell'applicazione di elaborazione eventi in modo da sapere quando aggiornare l'applicazione in risposta a un aggiornamento dello schema.
`application`	Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella Applicazione.
`client`	Informazioni sul client dell'app installato nel dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella Client.
`device`	Informazioni sul dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella Dispositivo. Per gli eventi e-mail, questo oggetto è vuoto.
`session`	Per gli eventi e-mail, questo oggetto è vuoto.

Attributo	Descrizione
`attributes`	Attributi associati all'evento. Per ulteriori informazioni, consulta la tabella Attributi. Per gli eventi segnalati da una delle app, questo oggetto può includere attributi personalizzati definiti dall'app. Per gli eventi creati quando si invia un messaggio da una campagna o da un viaggio, questo oggetto contiene gli attributi associati alla campagna o al viaggio. Per gli eventi che vengono generati quando si inviano messaggi transazionali, questo oggetto contiene informazioni correlate al messaggio stesso.
`client_context`	Per gli eventi e-mail, questo oggetto contiene un oggetto denominato `custom`, che contiene un attributo `legacy_identifier` . Il valore dell'attributo `legacy_identifier` è l'ID del progetto da cui è stato inviato il messaggio.
`facets`	Ulteriori informazioni sul messaggio, ad esempio le intestazioni e-mail. Per ulteriori informazioni, consulta la tabella Facet.
`awsAccountId`	L'ID dell' AWS account utilizzato per inviare il messaggio.

Applicazione

Include informazioni sul progetto Amazon Pinpoint a cui è associato l'evento.

Attributo	Descrizione
`app_id`	ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.

Attributo	Descrizione
`sdk`	L'SDK utilizzato per segnalare l'evento. Se invii un messaggio e-mail transazionale chiamando direttamente l'API Amazon Pinpoint o utilizzando la console Amazon Pinpoint, questo oggetto è vuoto.

Attributes

Include informazioni sulla campagna o viaggio che ha prodotto l'evento.

Campagna

Include informazioni sulla campagna che ha prodotto l'evento.

Attributo	Descrizione
`feedback`	Per gli eventi `_email.click` , il valore di questo attributo è l'URL del collegamento su cui il destinatario ha fatto clic per generare l'evento. Per gli altri eventi, questo valore rappresenta il tipo di evento, ad esempio `received`, `opened` o `clicked`.
`treatment_id`	Se il messaggio è stato inviato utilizzando una campagna di test A/B, questo valore rappresenta il numero della descrizione (treatment) del messaggio. Per le campagne standard e i messaggi e-mail transazionali, questo valore è `0`.
`campaign_activity_id`	ID univoco generato da Amazon Pinpoint quando si verifica l'evento.
`campaign_id`	L'ID univoco della campagna che ha inviato il messaggio.

Viaggio

Include informazioni sul viaggio che ha prodotto l'evento.

Attributo	Descrizione
`journey_run_id`	ID univoco del percorso che ha inviato il messaggio. Amazon Pinpoint genera e assegna automaticamente questo ID a ogni nuova esecuzione di un percorso.
`feedback`	Per gli eventi `_email.click` , il valore di questo attributo è l'URL del collegamento su cui il destinatario ha fatto clic per generare l'evento. Per gli altri eventi, questo valore rappresen ta il tipo di evento, ad esempio `received`, `delivered` o `opened`.
`journey_id`	L'ID univoco del viaggio che ha inviato il messaggio.
`journey_activity_id`	L'ID univoco dell'attività di viaggio che ha inviato il messaggio.

Client

Identificatore univoco del client che è stato scelto come target dalla campagna o dal percorso.

Attributo	Descrizione
`client_id`	ID del client. Il valore è l'ID dell'endpoint per campagne e percorsi e, per l'invio transazio nale, è un UUID.

Facet

Include informazioni sul messaggio e sul tipo di evento.

Attributo	Descrizione
`email_channel`	Contiene un oggetto `mail_event` , che contiene due oggetti: `mail` e un oggetto corrispondente al tipo di evento.

Mail

Include informazioni sul contenuto del messaggio e-mail, nonché i metadati relativi al messaggio.

Attributo	Descrizione
`message_id`	ID univoco del messaggio. Amazon Pinpoint genera automaticamente questo ID quando accetta il messaggio.
`message_send_timestamp`	La data e l'ora in cui il messaggio è stato inviato, nel formato specificato in RFC 822.
`from_address`	L'indirizzo e-mail da cui è stato inviato il messaggio.
`destination`	Un array che contiene gli indirizzi e-mail a cui è stato inviato il messaggio.
`headers_truncated`	Un valore booleano che indica se le intestazioni e-mail sono state troncate.
`headers`	Un oggetto che contiene diverse coppie nome-valore che corrispondono alle intestazioni nel messaggio. Questo oggetto in genere contiene informazioni sulle seguenti intestazioni: • `From`: indirizzo e-mail del mittente. • `To`: indirizzo e-mail del destinatario. • `Subject`: riga dell'oggetto dell'e-mail.

Attributo	Descrizione
	Tip L'intestazione dell'oggetto non è inclusa per gli eventi _email.send della campagna. • `MIME-Version` : indica che il messaggio è in formato MIME. Se questa intestazione è presente, il valore è sempre `1.0`. • `Content-Type` : tipo di supporto MIME del contenuto del messaggio.
`common_headers`	Contiene informazioni su varie intestazioni comuni per i messaggi e-mail. Le informazioni possono includere la data di invio del messaggio e le righe del destinatario, del mittente e dell'oggetto del messaggio.

Flusso di dati di eventi SMS da Amazon Pinpoint

Se in un progetto è abilitato il canale SMS, Amazon Pinpoint può eseguire lo streaming dei dati di eventi relativi ai recapiti dei messaggi SMS per il progetto. Dopo aver configurato lo streaming degli eventi, Amazon Pinpoint recupera i dati degli eventi dalla destinazione specificata durante la configurazione per consentirti di visualizzarli. Per informazioni su come configurare lo streaming degli eventi, consulta. Configura Amazon Pinpoint per lo streaming dei dati degli eventi delle app tramite Amazon Kinesis o Amazon Data Firehose

Note

Gli eventi SMS generati dai corrieri possono richiedere fino a 72 ore per essere ricevuti e non devono essere utilizzati per determinare se c'è un ritardo nella consegna dei messaggi in uscita. Dopo 72 ore, se Amazon Pinpoint non ha ricevuto un evento finale da un corriere, il

servizio restituisce automaticamente un record_status UNKNOWN, poiché Amazon Pinpoint non sa cosa è successo a quel messaggio.

Esempio di evento SMS

L'oggetto JSON per un evento SMS contiene i dati mostrati nell'esempio seguente.


```
{
  "event_type": "_SMS.SUCCESS",
  "event_timestamp": 1553104954322,
  "arrival_timestamp": 1553104954064,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "123456789012"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "sender_request_id": "565d4425-4b3a-11e9-b0a5-example",
    "campaign_activity_id": "cbcfc3c5e3bd48a8ae2b9cb41example",
    "origination_phone_number": "+12065550142",
    "destination_phone_number": "+14255550199",
    "record_status": "DELIVERED",
    "iso_country_code": "US",
    "treatment_id": "0",
    "number_of_message_parts": "1",
    "message_id": "1111-2222-3333",
    "message_type": "Transactional",
    "campaign_id": "52dc44b35c4742c98c5935269example"
    "customer_context": "{\"userId\":\"user-id-4\"}"
  },
  "metrics": {
    "price_in_millicents_usd": 645.0
  },
  "awsAccountId": "123456789012"
```

```
}
```

Attributi di eventi SMS

Questa sezione definisce gli attributi inclusi nell'esempio precedente dei dati del flusso di eventi che Amazon Pinpoint genera quando invii messaggi SMS.

Evento

Attributo	Descrizione
`event_type`	Tipo di evento. I valori possibili sono: • _SMS.BUFFERED: il messaggio non è ancora stato recapitato al destinatario. • _SMS.SUCCESS: il messaggio è stato correttamente accettato dall'operatore/recapitato al destinatario. • _SMS.FAILURE: Amazon Pinpoint non è stato in grado di recapitare il messaggio al destinatario. Per ulteriori informazioni sull'errore che ha impedito il recapito del messaggio, consulta `attributes.record_status` . • _SMS.OPTOUT: il cliente ha ricevuto il messaggio e ha risposto inviando la parola chiave di opt-out (in genere "STOP").
`event_timestamp`	L'ora in cui l'evento è stato segnalato, espressa nel formato orario Unix in millisecondi.
`arrival_timestamp`	L'ora in cui l'evento è stato ricevuto da Amazon Pinpoint, espressa nel formato orario Unix in millisecondi.
`event_version`	La versione dello schema JSON dell'evento.

Attributo	Descrizione
	Tip Seleziona questa versione nell'appl icazione di elaborazione eventi in modo da sapere quando aggiornare l'applica zione in risposta a un aggiornamento dello schema.
`application`	Informazioni sul progetto Amazon Pinpoint associato all'evento. Per ulteriori informazioni, consulta la tabella Applicazione.
`client`	Informazioni sul client dell'app installato nel dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella Client.
`device`	Informazioni sul dispositivo che ha segnalato l'evento. Per ulteriori informazioni, consulta la tabella Dispositivo. Per gli eventi SMS, questo oggetto è vuoto.
`session`	Per gli eventi SMS, questo oggetto è vuoto.

Attributo	Descrizione
`attributes`	Attributi associati all'evento. Per gli eventi segnalati da una delle app, questo oggetto può includere attributi personalizzati definiti dall'app. Per gli eventi che vengono creati quando si invia una campagna, questo oggetto contiene gli attributi associati alla campagna. Per gli eventi che vengono generati quando si inviano messaggi transazionali, questo oggetto contiene informazioni correlate al messaggio stesso. Per ulteriori informazioni, consulta la tabella Attributi.
`metrics`	Parametri aggiuntivi associati all'evento. Per ulteriori informazioni, consulta la tabella Parametri.
`awsAccountId`	L'ID dell' AWS account utilizzato per inviare il messaggio.

Applicazione

Include informazioni sul progetto Amazon Pinpoint a cui è associato l'evento e, se applicabile, sull'SDK utilizzato per segnalare l'evento.

Attributo	Descrizione
`app_id`	ID univoco del progetto Amazon Pinpoint che ha segnalato l'evento.
`sdk`	L'SDK utilizzato per segnalare l'evento. Se invii un messaggio SMS transazionale chiamando direttamente l'API Amazon Pinpoint o utilizzan

Attributo	Descrizione
	do la console Amazon Pinpoint, questo oggetto è vuoto.

Attributes

Include informazioni sugli attributi associati all'evento.

Attributo	Descrizione
`sender_request_id`	Un ID univoco associato alla richiesta di invio del messaggio SMS.
`campaign_activity_id`	L'ID univoco dell'attività all'interno della campagna.
`origination_phone_number`	Il numero di telefono da cui è stato inviato il messaggio.
`destination_phone_number`	Il numero di telefono a cui hai tentato di inviare il messaggio.
`record_status`	Ulteriori informazioni sullo stato del messaggio. I valori possibili includono: • SUCCESSFUL/DELIVERED: il recapito del messaggio è riuscito. • PENDING: il messaggio non è stato ancora recapitato al dispositivo del destinatario. • INVALID: il numero di telefono di destinazi one non è valido. • UNREACHABLE: il dispositivo del destinata rio è attualmente irraggiungibile o non disponibile. Ad esempio, il dispositivo potrebbe essere spento o disconnesso dalla

Attributo	Descrizione
	rete. Puoi provare a inviare nuovamente il messaggio in un secondo momento. • UNKNOWN: si è verificato un errore che ha impedito il recapito del messaggio. Questo errore è in genere transitorio e puoi tentare di inviare nuovamente il messaggio in un secondo momento. • BLOCKED: il dispositivo del destinatario sta bloccando i messaggi SMS dal numero di origine. • CARRIER_UNREACHABLE: un problema a livello di rete mobile del destinatario ha impedito il recapito del messaggio. Questo errore è in genere transitorio e puoi tentare di inviare nuovamente il messaggio in un secondo momento. • SPAM: l'operatore di telefonia mobile del destinatario ha identificato il contenuto del messaggio come spam e ha bloccato il recapito del messaggio. • INVALID_MESSAGE: il corpo del messaggio SMS non è valido e non può essere recapitat o. • CARRIER_BLOCKED: l'operatore del destinatario ha bloccato il recapito di questo messaggio. Ciò si verifica spesso quando il carrier identifica il contenuto del messaggio come non richiesto o dannoso. • TTL_EXPIRED: non è stato possibile recapitare il messaggio SMS entro un determinato intervallo di tempo. Questo errore è in genere transitorio e puoi tentare

Attributo	Descrizione
	di inviare nuovamente il messaggio in un secondo momento. • MAX_PRICE_EXCEEDED: l'invio del messaggio comporta un addebito che supera la quota mensile di spesa per SMS per il tuo account. Puoi richiedere un aumento di questa quota completando la procedura descritta in Richiesta di aumento della quota di spesa mensile per l'invio di SMS per Amazon SNS nella Guida per l'utente di Amazon Pinpoint. • OPTED_OUT : il messaggio SMS non è stato inviato perché il destinatario ha scelto mediante opt-out di non ricevere messaggi da te. • NO_QUOTA_LEFT_ON_ACCOUNT: sul tuo account non è disponibile una quota di spesa sufficiente per inviare il messaggio. È possibile richiedere un aumento di questa quota completando la procedura riportata in Richiesta di aumenti della quota di spesa mensile per SMS nella Guida per l'utente degli SMS di messaggistica per l'utente AWS finale. • NO_ORIGINATION_IDENTITY_AVAILABLE_TO_SEND: il tuo account non contiene un numero di telefono utilizzabile per inviare il messaggio alla destinazione. • DESTINATION_COUNTRY_NOT_SUPPORTED: il paese di destinazione è bloccato. Per tutti i paesi supportati, consulta Paesi e aree geografiche supportati (canale SMS) nella Guida per l'utente dei servizi di messaggistica SMS per l'utente AWS finale.

Attributo	Descrizione
	• ACCOUNT_IN_SANDBOX: il tuo account si trova in un ambiente di sperimentazione (sandbox) e può inviare messaggi solo a numeri di destinazione verificati. Puoi verificare il numero di destinazione nella console Amazon Pinpoint o avviare il processo per spostare l'account dalla sandbox. Vedi Informazioni sulla sandbox SMS/MMS e Voice nella Guida per l'utente di messaggistica SMS per l'AWS utente finale. • RATE_EXCEEDED: hai tentato di inviare un messaggio troppo velocemente e si è verificata una limitazione (della larghezza di banda della rete). Devi rallentare la frequenza delle chiamate. Per informazioni dettagliate sui nostri limiti, consulta i limiti di Message Parts per Second (MPS) nella Guida per l'AWS utente di End User Messaging SMS. • INVALID_ORIGINATION_IDENTITY: l'identità di origine specificata non è valida. • ORIGINATION_IDENTITY_DOES_NOT_EXIST: l'identità di origine specificata non esiste. • INVALID_DLT_PARAMETERS: sono stati specificati parametri DLT non validi (obbligatori per le destinazioni in India). • INVALID_PARAMETERS: sono stati specificati parametri non validi. • ACCESS_DENIED: al tuo account è impedito l'invio di messaggi. Contatta il Supporto clienti per scoprire la causa e risolvere il problema.

Attributo	Descrizione
	• INVALID_KEYWORD: la parola chiave specificata non è valida. È possibile che la parola chiave abbia un formato errato o non sia stata impostata nel tuo account. • INVALID_SENDER_ID: l'ID mittente specifica to non è valido. È possibile che l'ID mittente abbia un formato o una lunghezza errati. • INVALID_POOL_ID: l'ID del pool specificato non è valido. È possibile che l'ID del pool sia essere in un formato errato o non appartenga al tuo account. • SENDER_ID_NOT_SUPPORTED_FOR _DESTINATION: il paese di destinazione non supporta l'ID mittente. È necessario utilizzar e un numero di telefono o un'altra identità di origine per l'invio. • INVALID_PHONE_NUMBER: il numero di telefono di origine specificato non è valido. È possibile che il numero di telefono abbia un formato o una lunghezza errati.
`iso_country_code`	Il paese associati al numero di telefono del destinatario, mostrati nel formato ISO 3166-1 alpha-2.
`treatment_id`	L'ID della descrizione (treatment) del messaggio, se il messaggio è stato inviato in una campagna A/B.
`treatment_id`	Se il messaggio è stato inviato utilizzando una campagna di test A/B, questo valore rappresen ta il numero della descrizione (treatment) del messaggio. Per i messaggi SMS transazionali, questo valore è 0.

Attributo	Descrizione
`number_of_message_parts`	Numero di parti di messaggio che Amazon Pinpoint ha creato per inviare il messaggio. In genere, i messaggi SMS possono contenere solo 160 caratteri GSM-7 o 67 caratteri non GSM, anche se questi limiti possono variare in base al paese . Se invii un messaggio che supera questi limiti, Amazon Pinpoint lo suddivide automaticamente in parti più piccole. I costi vengono fatturati in base al numero di parti del messaggio inviate.
`message_id`	ID univoco che Amazon Pinpoint genera quando accetta il messaggio.
`message_type`	Il tipo di messaggio. I valori possibili sono Promotional (Promozionale) e Transactional (Transazionale). Specifichi questo valore quando crei una campagna o quando invii messaggi transazionali utilizzando l'SendMessagesoperazione nell'API Amazon Pinpoint.
`campaign_id`	ID univoco della campagna Amazon Pinpoint che ha inviato il messaggio.
`customer_context`	Una stringa JSON del contenuto della `Context` mappa inviata in un'operazione Amazon SendMessagesPinpoint.

Client

Include informazioni sul client dell'app installato nel dispositivo che ha segnalato l'evento.

Attributo	Descrizione
`client_id`	Per gli eventi generati dalle app, questo valore è l'ID univoco del client dell'app installato sul dispositivo. Questo ID viene generato automaticamente da AWS Mobile SDK for iOS e da. AWS Mobile SDK for Android Per gli eventi generati quando invii campagne e messaggi transazionali, questo valore è uguale all'ID dell'endpoint a cui hai inviato il messaggio .
`cognito_id`	ID univoco assegnato al client dell'app nel pool di identità Amazon Cognito utilizzato dall'app.

Dispositivo

Include informazioni sul dispositivo che ha segnalato l'evento.

Attributo	Descrizione
`locale`	Le impostazioni locali del dispositivo.
`make`	La marca del dispositivo, ad esempio `Apple` o `Samsung`.
`model`	Il modello del dispositivo, ad esempio `iPhone`.
`platform`	La piattaforma del dispositivo, ad esempio `ios` o `android`.

Metriche

Include informazioni sui parametri associati all'evento.

Attributo	Descrizione
`price_in_millicents_usd`	L'importo addebitato per l'invio del messaggio . Questo prezzo è indicato in millesimi di un centesimo degli Stati Uniti. Ad esempio, se il valore di questo attributo è 645, vengono addebitati 0,645 ¢ per l'invio del messaggio (645/1000=0,645¢=$0,00645). **Note** Questa proprietà non viene visualizzata per i messaggi con un `event_type` di _SMS.BUFFERED.

Eliminare un flusso di eventi da Amazon Pinpoint

Se assegni un flusso Kinesis a un'applicazione, puoi disabilitare lo streaming di eventi per quell'applicazione. Amazon Pinpoint interrompe lo streaming degli eventi su Kinesis, ma puoi visualizzare l'analisi degli eventi utilizzando la console Amazon Pinpoint.

AWS CLI

Utilizza il comando `delete-event-stream`:

```
aws pinpoint delete-event-stream --application-id application-id
```

AWS SDK per Java

Utilizza il `deleteEventStream`metodo del client Amazon Pinpoint:

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

Interroga i dati di analisi di Amazon Pinpoint

Oltre a utilizzare le pagine di analisi sulla console Amazon Pinpoint, puoi utilizzare Amazon Pinpoint APIs Analytics per interrogare i dati di analisi per un sottoinsieme di metriche standard che forniscono informazioni sulle tendenze relative al coinvolgimento degli utenti, alla sensibilizzazione delle campagne e altro ancora. Queste metriche, note anche come indicatori chiave di performance (KPIs), sono valori misurabili che possono aiutarti a monitorare e valutare le prestazioni dei tuoi progetti, campagne e percorsi.

Se lo utilizzi APIs per interrogare i dati di analisi, puoi analizzarli utilizzando lo strumento di reporting di tua scelta, senza dover accedere alla console Amazon Pinpoint o analizzare i dati grezzi degli eventi da fonti come Amazon Kinesis Streams. Ad esempio, puoi creare un dashboard personalizzato che visualizzi i risultati settimanali delle campagne o che fornisca analisi approfondite sulle percentuali di consegna per le tue campagne.

Puoi interrogare i dati utilizzando l'API REST di Amazon Pinpoint, AWS Command Line Interface (AWS CLI) o un AWS SDK. Per eseguire una query sui dati, invia una richiesta all'API Amazon Pinpoint e utilizza i parametri supportati per specificare i dati desiderati e i filtri da applicare. Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. È quindi possibile trasferire i risultati a un altro servizio o a un'altra applicazione per un'analisi più approfondita, lo storage o la generazione di report.

Amazon Pinpoint raccoglie e aggrega automaticamente i dati per tutti i tipi di metriche supportati e per tutti i progetti, le campagne e i percorsi. Inoltre, i dati vengono aggiornati in modo continuo, generando un intervallo di tempo di latenza dei dati che è limitato a circa due ore. Si noti, tuttavia, che potrebbe esserci una latenza dei dati aggiuntiva per determinati parametri. Questo perché i dati per alcuni parametri si basano sulle informazioni che riceviamo dai provider di posta elettronica dei destinatari. Alcuni fornitori ci inviano immediatamente queste informazioni, mentre altri le inviano meno frequentemente.

Amazon Pinpoint archivia i dati per 90 giorni. Per archiviare i dati per più di 90 giorni o per accedere ai dati di analisi non elaborati in tempo reale, puoi configurare un progetto Amazon Pinpoint per lo streaming dei dati degli eventi su Amazon Kinesis Data Streams o Amazon Data Firehose. Per informazioni sulla configurazione dei flussi di eventi, consulta Trasmetti in streaming i dati degli eventi delle app tramite Kinesis e Firehose utilizzando Amazon Pinpoint.

Componenti e parametri di query per le metriche in Amazon Pinpoint

Per eseguire una query sui dati per un parametro, inviare una richiesta `get` alla risorsa dei parametri appropriata dell'API di Amazon Pinpoint. Nella richiesta, è possibile definire la query utilizzando parametri supportati per i seguenti componenti di query:

- Progetto: specifica un progetto fornendo il relativo ID come valore per il parametro `application-id`. Questo parametro è obbligatorio per tutti i parametri.
- Campagna: specifica una campagna fornendo il relativo ID come valore per il parametro `campaign-id`. Questo parametro è obbligatorio solo per i parametri della campagna.
- Percorso: specificare un percorso fornendo il relativo ID come valore per il parametro `journey-id`. Questo parametro è obbligatorio solo per i parametri di coinvolgimento e di esecuzione del viaggio e per i parametri di esecuzione dell'attività di viaggio.
- Attività del percorso: specifica un'attività del percorso fornendo il relativo ID come valore per il parametro `journey-activity-id`. Questo parametro è obbligatorio solo per i parametri di esecuzione dell'attività di viaggio.
- Intervallo di date: per filtrare facoltativamente i dati per intervallo di date, specifica la prima e l'ultima data e l'ora dell'intervallo utilizzando i parametri di ora di inizio e fine supportati. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio `2019-07-19T20:00:00Z` per le 20:00 UTC del 19 luglio 2019.

  Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. I parametri dell'intervallo di date sono supportati da tutti i parametri, ad eccezione dei parametri di esecuzione del viaggio e dei parametri di esecuzione dell'attività di viaggio.
- Parametro: specifica una metrica fornendo il relativo nome come valore per il parametro `kpi-name`. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Alcuni esempi sono `email-open-rate` e `successful-delivery-rate`. Questo parametro è obbligatorio per tutti i parametri, ad eccezione dei parametri di esecuzione del viaggio e dei parametri di esecuzione dell'attività di viaggio. Per un elenco completo dei parametri supportati e il valore `kpi-name` da utilizzare per ciascuno di essi, consulta Metriche standard per progetti, campagne e viaggi.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. Nella risposta, la struttura dei risultati varia a seconda del parametro richiesto.

Alcune metriche forniscono un solo valore, ad esempio il numero di messaggi recapitati da una campagna. Altre metriche forniscono più valori e in genere raggruppano tali valori in base a un campo pertinente, ad esempio il numero di messaggi recapitati da ciascuna campagna, raggruppati per esecuzione della campagna. Se un parametro fornisce e raggruppa più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati. Per ulteriori informazioni sulla struttura dei risultati delle query, consulta Usa i risultati delle query JSON.

Policy IAM per l'esecuzione di query sui dati di analisi di Amazon Pinpoint

Utilizzando l'API Amazon Pinpoint, puoi interrogare i dati di analisi per un sottoinsieme di metriche standard, note anche come indicatori chiave di prestazione (KPIs), che si applicano a progetti, campagne e percorsi di Amazon Pinpoint. Questi parametri consentono di monitorare e valutare le prestazioni dei progetti, delle campagne e dei viaggi.

Per gestire l'accesso a questi dati, puoi creare policy AWS Identity and Access Management (IAM) che definiscono le autorizzazioni per i ruoli o gli utenti IAM autorizzati ad accedere ai dati. Per supportare il controllo granulare dell'accesso a questi dati, Amazon Pinpoint fornisce diverse azioni distinte che è possibile specificare nelle policy IAM. Per visualizzare i dati di analisi sulla console Amazon Pinpoint (`mobiletargeting:GetReports`), è disponibile un'azione distinta, mentre per accedere ai dati di analisi a livello di programmazione tramite l'API Amazon Pinpoint sono disponibili altre azioni.

Per creare policy IAM che gestiscano l'accesso ai dati di analisi, puoi utilizzare l'API AWS Management Console AWS CLI, the o IAM. Tieni presente che la scheda Editor visivo sulla AWS Management Console al momento non include azioni per la visualizzazione o l'esecuzione di query sui dati di analisi di Amazon Pinpoint. Tuttavia, puoi aggiungere manualmente le operazioni necessarie alle policy IAM utilizzando la scheda JSON sulla console.

Ad esempio, la seguente policy consente l'accesso programmatico a tutti i dati di analisi per tutti i tuoi progetti, campagne e viaggi in tutte le AWS regioni:

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

*accountId*Dov'è l'ID del tuo AWS account.

Tuttavia, è consigliabile definire policy in grado di seguire il principio del privilegio minimo. In altre parole, è necessario creare policy che includano solo le autorizzazioni necessarie per eseguire un'attività specifica. Per supportare questa pratica e implementare un controllo più granulare, puoi limitare l'accesso programmatico ai dati di analisi solo per un particolare progetto in una AWS regione specifica, ad esempio:

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

Dove:

- *region*è il nome della AWS regione che ospita il progetto.
- *accountId*è l'ID AWS del tuo account.
- *projectId*è l'identificatore del progetto a cui desideri fornire l'accesso.

Analogamente, la seguente policy di esempio consente l'accesso programmatico ai dati di analisi solo per una particolare campagna:

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

Dove:

- *region*è il nome della AWS regione che ospita il progetto.
- *accountId*è il tuo Account AWS ID.

- *projectId*è l'identificatore del progetto associato alla campagna.
- *campaignId*è l'identificatore della campagna a cui desideri fornire l'accesso.

E la seguente policy di esempio consente l'accesso programmatico a tutti i dati di analisi, sia di coinvolgimento sia di esecuzione, per un particolare percorso e le attività che lo compongono:

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

Dove:

- *region*è il nome della AWS regione che ospita il progetto.
- *accountId*è l'ID AWS del tuo account.
- *projectId*è l'identificatore del progetto associato al viaggio.
- *journeyId*è l'identificatore del viaggio a cui desideri fornire l'accesso.

Per l'elenco completo delle azioni API Amazon Pinpoint che è possibile utilizzare nelle policy IAM, consulta Azioni di Amazon Pinpoint per le policy IAM. Per ulteriori informazioni sulla creazione e sulla gestione delle policy IAM, consulta la Guida per l'utente di IAM.

Metriche standard che si applicano a progetti, campagne e percorsi Amazon Pinpoint

Puoi utilizzare Amazon Pinpoint Analytics APIs per interrogare i dati di analisi per un sottoinsieme di metriche standard che si applicano a progetti, campagne e percorsi Amazon Pinpoint. Queste metriche, note anche come indicatori chiave di performance (KPIs), sono valori misurabili che possono aiutarti a monitorare e valutare le prestazioni di progetti, campagne e percorsi.

Amazon Pinpoint fornisce l'accesso programmatico ai dati di analisi per diversi tipi di metriche standard:

- Metriche delle applicazioni: queste metriche forniscono informazioni dettagliate sulle tendenze per tutte le campagne associate a un progetto, che viene indicato anche come applicazione. Ad esempio, è possibile utilizzare un parametro di applicazione per ottenere i dettagli del numero di messaggi aperti dai destinatari per ogni campagna associata a un progetto.
- Metriche della campagna: queste metriche forniscono informazioni dettagliate sulle prestazioni delle campagne individuali. Ad esempio, è possibile utilizzare un parametro di campagna per stabilire il numero di endpoint ai quali è stato inviato un messaggio della campagna o quanti di questi messaggi sono stati distribuiti agli endpoint.
- Metriche di coinvolgimento del percorso: queste metriche forniscono informazioni dettagliate sulle prestazioni dei singoli percorsi. Ad esempio, è possibile utilizzare un parametro di coinvolgimento del viaggio per ottenere una ripartizione del numero di messaggi aperti dai partecipanti a ciascuna attività di un viaggio.
- Metriche di esecuzione del percorso: queste metriche forniscono informazioni dettagliate sulle tendenze di partecipazione dei singoli percorsi. Ad esempio, è possibile utilizzare un parametro di esecuzione del viaggio per determinare quanti partecipanti hanno iniziato un viaggio.
- Metriche di esecuzione delle attività del percorso: queste metriche forniscono informazioni dettagliate sulle tendenze di partecipazione delle singole attività in un percorso. Ad esempio, è possibile utilizzare un parametro di esecuzione dell'attività di viaggio per determinare quanti partecipanti hanno iniziato un'attività e quanti partecipanti hanno completato ciascun percorso in un’attività.

Gli argomenti in questa sezione elencano e descrivono i singoli parametri su cui è possibile eseguire query per ogni tipo di parametro.

Argomenti

- Metriche delle applicazioni Amazon Pinpoint per le campagne
- Parametri delle applicazioni Amazon Pinpoint per i messaggi e-mail transazionali
- Parametri delle applicazioni Amazon Pinpoint per i messaggi SMS transazionali
- Metriche delle campagne Amazon Pinpoint
- Metriche di coinvolgimento del percorso Amazon Pinpoint
- Metriche di esecuzione del percorso Amazon Pinpoint
- Metriche di esecuzione delle attività del percorso Amazon Pinpoint
- Metriche del percorso e dell'esecuzione delle campagne di Amazon Pinpoint

Metriche delle applicazioni Amazon Pinpoint per le campagne

La tabella seguente elenca e descrive le metriche standard delle applicazioni su cui è possibile eseguire query per valutare le prestazioni di tutte le campagne associate a un progetto Amazon Pinpoint. Per eseguire query sui dati per queste metriche, utilizza la risorsa Metriche delle applicazioni dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro `kpi-name` in una query.

Parametro	Kpi-name	Descrizione
Percentuale di recapiti	`successful-delivery-rate`	Per tutte le campagne associate a un progetto, la percentuale di messaggi inviati ai destinatari. Questo parametro viene calcolato come il numero di messaggi inviati da tutte le campagne per un progetto e recapitati ai destinatari, diviso il numero di messaggi che sono stati inviati da tutte le campagne.
Percentuale di recapiti, raggruppati per data	`successful-delivery-rate-grouped-by-date`	Per tutte le campagne associate a un progetto, la percentuale di messaggi

Parametro	Kpi-name	Descrizione
		recapitati ai destinatari per ogni giorno nell'intervallo di date specificato. Questo parametro viene calcolato come il numero di messaggi inviati da tutte le campagne per un progetto e recapitati ai destinatari, diviso il numero di messaggi che sono stati inviati da tutte le campagne, per ogni giorno nell'intervallo di date specifica to. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Percentuale di e-mail aperte	`email-open-rate`	Per tutte le campagne associate a un progetto, la percentuale di messaggi e-mail aperti dai destinatari. Questo parametro viene calcolato come il numero di messaggi e-mail inviati da tutte le campagne per un progetto e aperti dai destinatari, diviso il numero di messaggi e-mail inviati da tutte le campagne e recapitati ai destinatari.

Parametro	Kpi-name	Descrizione
Percentuale di e-mail aperte, raggruppate per campagna	`email-open-rate-grouped-by-campaign`	Per ogni campagna associata a un progetto, la percentuale di messaggi e-mail aperti dai destinatari. Questo parametro viene calcolato come il numero di messaggi e-mail inviati da una campagna e aperti dai destinatari, diviso il numero di messaggi e-mail inviati dalla campagna e recapitati ai destinatari. I risultati della query per questo parametro sono raggruppati in base all'ID campagna (`CampaignId`), che è una stringa che identifica in modo univoco una campagna.
Consegne endpoint	`unique-deliveries`	Per tutte le campagne associate a un progetto, il numero di endpoint univoci ai quali sono stati consegnati i messaggi.

Parametro	Kpi-name	Descrizione
Consegne endpoint, raggruppate per campagna	`unique-deliveries-grouped-by-campaign`	Per ogni campagna associata a un progetto, il numero di endpoint univoci ai quali sono stati consegnati i messaggi. I risultati della query per questo parametro sono raggruppati in base all'ID campagna (`CampaignId`), che è una stringa che identifica in modo univoco una campagna.
Consegne endpoint, raggruppate per data	`unique-deliveries-grouped-by-date`	Per tutte le campagne associate a un progetto, il numero di endpoint univoci ai quali sono stati consegnati i messaggi, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.

Parametro	Kpi-name	Descrizione
Messaggi recapitati, raggruppati per campagna	`successful-deliveries-grouped-by-campaign`	Per ogni campagna associata a un progetto, il numero di messaggi che sono stati recapitati ai destinatari. Questo parametro viene calcolato come il numero di messaggi inviati da una campagna, meno il numero di messaggi che sono stati inviati dalla campagna e non sono stati recapitati ai destinatari a causa di un hard bounce. I risultati della query per questo parametro sono raggruppati in base all'ID campagna (`CampaignId`), che è una stringa che identifica in modo univoco una campagna.

Parametro	Kpi-name	Descrizione
Percentuale di notifiche push aperte	`push-open-rate`	Per tutte le campagne associate a un progetto, la percentuale di notifiche push aperte dai destinatari. Questo parametro viene calcolato in base al numero di notifiche push che sono state inviate da tutte le campagne per un progetto e aperte dai destinatari, diviso il numero di notifiche push che sono state inviate da tutte quelle campagne e consegnate ai destinatari.

Parametro	Kpi-name	Descrizione
Percentuale di notifiche push aperte, raggruppate per campagna	`push-open-rate-grouped-by-campaign`	Per ogni campagna associata a un progetto, la percentuale di notifiche push aperte dai destinatari. Questo parametro viene calcolato in base al numero di notifiche push che sono state inviate tramite una campagna e aperte dai destinatari, diviso il numero di notifiche push inviate dalla campagna e consegnate ai destinatari. I risultati della query per questo parametro sono raggruppati in base all'ID campagna (`CampaignId`), che è una stringa che identifica in modo univoco una campagna.

Parametri delle applicazioni Amazon Pinpoint per i messaggi e-mail transazionali

La tabella seguente elenca e descrive le metriche standard delle applicazioni su cui è possibile eseguire query per monitorare le tendenze relative a tutti i messaggi e-mail transazionali associati a un progetto Amazon Pinpoint. Per eseguire query sui dati per queste metriche, utilizza la risorsa Metriche delle applicazioni dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro `kpi-name` in una query.

Tieni presente che questi parametri non forniscono dati sui messaggi e-mail inviati dalle campagne. Forniscono dati solo sui messaggi e-mail transazionali. Per eseguire query sui dati per i messaggi inviati da una o più campagne, utilizza un parametro della campagna o un parametro di applicazione per le campagne.

Parametro	Kpi-name	Descrizione
Clic	`txn-emails-clicked`	Il numero di volte che i destinatari hanno fatto clic sui collegamenti nei messaggi. Se un singolo destinatario ha fatto clic su più collegamenti in un messaggio o ha fatto clic sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio.
Clic, raggruppati per data	`txn-emails-clicked-grouped-by-date`	Il numero di volte in cui i destinatari hanno fatto clic sui collegamenti nei messaggi, per ogni giorno nell'intervallo di date specificato. Se un singolo destinatario ha fatto clic su più collegamenti in un messaggio o ha fatto clic sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Complaint rate (Percentuale di reclami)	`txn-emails-complaint-rate`	La percentuale di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate. Questo parametro viene calcolato come il numero

Parametro	Kpi-name	Descrizione
		di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate, diviso per il numero di messaggi inviati.
Percentuale di reclami, raggruppati per data	`txn-emails-complaint-rate-grouped-by-date`	La percentuale di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate, per ogni giorno nell'intervallo di date specificato. Questo parametro viene calcolato come il numero di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate, diviso per il numero di messaggi inviati, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Reclami	`txn-emails-with-complaints`	Il numero di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate.

Parametro	Kpi-name	Descrizione
Reclami, raggruppati per data	`txn-emails-with-complaints-grouped-by-date`	Il numero di messaggi segnalati dai destinatari come e-mail non richieste o indesiderate, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Consegne	`txn-emails-delivered`	Il numero di messaggi consegnati ai destinatari. Questo parametro viene calcolato come il numero di messaggi inviati, meno il numero di messaggi che non è stato possibile consegnare a causa di un soft bounce o di un hard bounce o perché sono stati rifiutati. Un messaggio viene rifiutato se Amazon Pinpoint determina che il messaggio contiene malware. Amazon Pinpoint non tenta di inviare i messaggi rifiutati.

Parametro	Kpi-name	Descrizione
Consegne, raggruppate per data	`txn-emails-delivered-grouped-by-date`	Il numero di messaggi consegnati ai destinatari, per ogni giorno nell'intervallo di date specificato. Questo parametro viene calcolato come il numero di messaggi inviati, meno il numero di messaggi che non è stato possibile consegnare a causa di un soft bounce o di un hard bounce o perché sono stati rifiutati, per ogni giorno nell'intervallo di date specificato. Un messaggio viene rifiutato se Amazon Pinpoint determina che il messaggio contiene malware. Amazon Pinpoint non tenta di inviare i messaggi rifiutati. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Percentuale di recapiti	`txn-emails-delivery-rate`	La percentuale di messaggi consegnati ai destinatari. Questo parametro viene calcolato come il numero di messaggi inviati e consegnati ai destinatari, diviso il numero di messaggi inviati.

Parametro	Kpi-name	Descrizione
Percentuale di recapiti, raggruppati per data	`txn-emails-delivery-rate-grouped-by-date`	La percentuale di messaggi consegnati ai destinatari, per ogni giorno nell'intervallo di date specificato. Questo parametro viene calcolato come il numero di messaggi inviati e consegnati ai destinatari, diviso il numero di messaggi inviati per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Mancati recapiti permanenti	`txn-emails-hard-bounced`	Il numero di messaggi che non è stato possibile consegnare ai destinatari a causa di un hard bounce. Un mancato recapito permanente si verifica se un problema persistente impedisce il recapito di un messaggio, ad esempio se l'indirizzo e-mail del destinatario non esiste.

Parametro	Kpi-name	Descrizione
Hard bounce, raggruppati per data	`txn-emails-hard-bounced-grouped-by-date`	Il numero di messaggi che non è stato possibile consegnare ai destinatari a causa di un hard bounce, per ogni giorno nell'intervallo di date specificato. Un mancato recapito permanente si verifica se un problema persistente impedisce il recapito di un messaggio, ad esempio se l'indirizzo e-mail del destinatario non esiste. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Aperture	`txn-emails-opened`	Mostra il numero di messaggi che sono stati aperti dai destinatari:
Aperture, raggruppate per data	`txn-emails-opened-grouped-by-date`	Il numero di messaggi che sono stati aperti dai destinatari, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Invii	`txn-emails-sent`	Il numero di messaggi che sono stati inviati.

Parametro	Kpi-name	Descrizione
Invii, raggruppati per data	`txn-emails-sent-grouped-by-date`	Il numero di messaggi inviati, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Soft bounce	`txn-emails-soft-bounced`	Il numero di messaggi che non è stato possibile consegnare ai destinatari a causa di un soft bounce. Si verifica un caso di e-mail non recapitata se un problema temporaneo impedisce il recapito di un messaggio, ad esempio se la posta in arrivo del destinatario è piena o il server di ricezione è temporaneamente non disponibile.

Parametro	Kpi-name	Descrizione
Soft bounce, raggruppati per data	`txn-emails-soft-bounced-grouped-by-date`	Il numero di messaggi che non è stato possibile consegnare ai destinatari a causa di un soft bounce, per ogni giorno nell'intervallo di date specificato. Si verifica un caso di e-mail non recapitata se un problema temporaneo impedisce il recapito di un messaggio, ad esempio se la posta in arrivo del destinatario è piena o il server di ricezione è temporaneamente non disponibile. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.

Parametro	Kpi-name	Descrizione
Eventi clic utente univoci	`txn-emails-unique-clicks`	Il numero di destinatari univoci (endpoint) che hanno fatto clic sui collegamenti nei messaggi. A differenza del parametro Clicks (Clic), questo parametro indica il numero di destinatari univoci che hanno fatto clic sui collegamenti, non il numero di eventi di clic che si sono verificati. Ad esempio, se un singolo destinatario ha fatto clic su più collegamenti nello stesso messaggio o ha fatto clic sullo stesso link più di una volta, questo parametro segnala un solo evento clic per quel destinatario.

Parametro	Kpi-name	Descrizione
Eventi clic utente univoci, raggruppati per data	`txn-emails-unique-clicks-grouped-by-date`	Il numero di destinatari univoci (endpoint) che hanno fatto clic sui collegamenti nei messaggi, per ogni giorno nell'intervallo di date specificato. A differenza del parametro Clicks, grouped by date (Clic, raggruppati per data) questo parametro indica il numero di destinatari univoci che hanno fatto clic sui collegamenti, non il numero di eventi clic che si sono verificati. Ad esempio, se un singolo destinatario ha fatto clic su più collegamenti nello stesso messaggio o ha fatto clic sullo stesso link più di una volta, questo parametro segnala un solo evento clic per quel destinatario. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.

Parametro	Kpi-name	Descrizione
Eventi univoci di apertura degli utenti	`txn-emails-unique-opens`	Il numero di destinatari univoci (endpoint) che hanno aperto i messaggi. A differenza del parametro Opens (Aperture), questo parametro indica il numero di destinatari univoci che hanno aperto i messaggi, non il numero di eventi di apertura che si sono verificati. Ad esempio, se un singolo destinatario apre lo stesso messaggio più volte, questo parametro segnala un solo evento di apertura per quel destinatario.

Parametro	Kpi-name	Descrizione
Eventi univoci di apertura degli utenti, raggruppati per data	`txn-emails-unique-opens-grouped-by-date`	Il numero di destinatari univoci (endpoint) che hanno aperto messaggi, per ogni giorno nell'intervallo di date specificato. A differenza del parametro Opens, grouped by date (Aperture, raggruppate per data) questo parametro indica il numero di destinatari univoci che hanno aperto i messaggi, non il numero di eventi di apertura che si sono verificati. Ad esempio, se un singolo destinatario apre lo stesso messaggio più volte, questo parametro segnala un solo evento di apertura per quel destinatario. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.

Parametri delle applicazioni Amazon Pinpoint per i messaggi SMS transazionali

La tabella seguente elenca e descrive le metriche standard delle applicazioni su cui è possibile eseguire query per ottenere informazioni sulle tendenze relative a tutti i messaggi SMS transazionali associati a un progetto Amazon Pinpoint. Per eseguire query sui dati per queste metriche, utilizza la

risorsa Metriche delle applicazioni dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro `kpi-name` in una query.

Tieni presente che questi parametri non forniscono dati sui messaggi SMS inviati dalle campagne. Forniscono dati solo sui messaggi SMS transazionali. Per eseguire query sui dati per i messaggi inviati da una o più campagne, utilizza un parametro della campagna o un parametro di applicazione per le campagne.

Parametro	Kpi-name	Descrizione
Prezzo medio per messaggio, raggruppato per paese	`txn-sms-average-price-grouped-by-country`	Costo totale dell'invio dei messaggi, per ogni paese o regione a cui sono stati inviati i messaggi. Il prezzo è indicato in millesimi di un centesimo degli Stati Uniti. Ad esempio, se il valore di questo attributo è 645, vengono addebitati 0,645 ¢ per l'invio del messaggio (645/1000 =0,645¢=$0,00645). Questo parametro viene calcolato come costo totale di tutti i messaggi inviati ai destinatari in ciascun paese o regione, diviso per il numero di messaggi inviati ai destinatari in ciascuno di tali paesi e regioni. I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.

Parametro	Kpi-name	Descrizione
Prezzo medio per parte di messaggio, raggruppato per paese	`txn-sms-average-price-by-parts-grouped-by-country`	Costo totale dell'invio delle parti di messaggio, per ogni paese o regione a cui sono stati inviati i messaggi. Una parte di messaggio è una sezione di un messaggio SMS. Il prezzo è indicato in millesimi di un centesimo degli Stati Uniti. Ad esempio, se il valore di questo attributo è 645, vengono addebitati 0,645 ¢ per l'invio del messaggio (645/1000 =0,645¢=$0,00645). Questa metrica viene calcolata come costo totale di tutte le parti di messaggio inviate ai destinatari in ciascun paese o regione, diviso per il numero di parti di messaggio inviati ai destinatari in ciascuno di tali paesi e regioni. I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.
Consegne	`txn-sms-delivered`	Il numero di messaggi consegnati ai destinatari.

Parametro	Kpi-name	Descrizione
Consegne, raggruppate per paese	`txn-sms-delivered-grouped-by-country`	Il numero di messaggi consegnati ai destinatari per ogni paese o regione a cui sono stati inviati i messaggi. I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.
Consegne, raggruppate per data	`txn-sms-delivered-grouped-by-date`	Il numero di messaggi consegnati ai destinatari, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Errori di recapito	`txn-sms-error-distribution`	Il numero di volte in cui si è verificato un errore durante il tentativo di recapitare i messaggi, per ogni tipo di errore che si è verificato. I risultati della query per questo parametro sono raggruppati per codice di errore, per ogni tipo di errore che si è verificato.

Parametro	Kpi-name	Descrizione
Percentuale di recapiti	`txn-sms-delivery-rate`	La percentuale di messaggi consegnati ai destinatari. Questo parametro viene calcolato come il numero di messaggi inviati e consegnati ai destinatari, diviso il numero di messaggi inviati.
Percentuale di recapiti, raggruppati per data	`txn-sms-delivery-rate-grouped-by-date`	La percentuale di messaggi consegnati ai destinatari, per ogni giorno nell'intervallo di date specificato. Questo parametro viene calcolato come il numero di messaggi inviati e consegnati ai destinatari, diviso il numero di messaggi inviati per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.

Parametro	Kpi-name	Descrizione
Parti del messaggio consegnate	`txn-sms-delivered-by-parts`	Numero di parti di messaggio recapitate ai partecipanti. Una parte di messaggio è una sezione di un messaggio SMS. Se un messaggio SMS contiene più caratteri di quelli consentiti dal protocollo SMS, Amazon Pinpoint divide il messaggio nel numero di parti necessarie per l'invio del messaggio a un destinatario.
Parti di messaggio recapitate, raggruppate per paese	`txn-sms-delivered-by-parts-grouped-by-country`	Numero di parti di messaggio recapitate ai destinatari per ogni paese o regione a cui sono stati inviati i messaggi. Una parte di messaggio è una sezione di un messaggio SMS. I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.

Parametro	Kpi-name	Descrizione
Parti di messaggio inviate	`txn-sms-sent-by-parts`	Numero di parti di messaggio inviate. Una parte di messaggio è una sezione di un messaggio SMS. Se un messaggio SMS contiene più caratteri di quelli consentiti dal protocollo SMS, Amazon Pinpoint divide il messaggio nel numero di parti necessarie per l'invio del messaggio a un destinatario.
Parti di messaggio inviate, raggruppate per paese	`txn-sms-sent-by-parts-grouped-by-country`	Numero di parti messaggio inviate per ogni paese o regione a cui sono stati inviati i messaggi. Una parte di messaggio è una sezione di un messaggio SMS. I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.
Messages sent (Messaggi inviati)	`txn-sms-sent`	Il numero di messaggi che sono stati inviati.

Parametro	Kpi-name	Descrizione
Messaggi inviati, raggruppati per paese	`txn-sms-sent-grouped-by-country`	Il numero di messaggi inviati per ogni paese o regione a cui sono stati inviati i messaggi. I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.
Messaggi inviati, raggruppati per data	`txn-sms-sent-grouped-by-date`	Il numero di messaggi inviati, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.

Parametro	Kpi-name	Descrizione
Prezzo totale, raggruppato per paese	`txn-sms-total-price-grouped-by-country`	Costo totale dell'invio dei messaggi, per ogni paese o regione a cui sono stati inviati i messaggi. Il prezzo è indicato in millesimi di un centesimo degli Stati Uniti. Ad esempio, se il valore di questo attributo è 645, vengono addebitati 0,645 ¢ per l'invio del messaggio (645/1000 =0,645¢=$0,00645). I risultati delle query per questo parametro sono raggruppati per paese o regione, nel formato ISO 3166-1 alfa-2.

Metriche delle campagne Amazon Pinpoint

La tabella seguente elenca e descrive i parametri standard delle campagne su cui è possibile eseguire query per valutare le performance di una singola campagna. Per eseguire query sui dati per queste metriche, utilizza la risorsa Parametri della campagna dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro `kpi-name` nella query.

Parametro	Kpi-name	Descrizione
Bounce rate (Percentuale di mancati recapiti (bounce))	`hard-bounce-rate`	Per ogni esecuzione della campagna, la percentuale di messaggi e-mail che non è stato possibile recapitare ai destinatari. Questa metrica misura solo le occorrenze di mancato recapito permanent

Parametro	Kpi-name	Descrizione
		e, ovvero i messaggi in cui l'indirizzo e-mail del destinatario presenta un problema permanente che impedisce il recapito del messaggio. Questo parametro viene calcolato come il numero di messaggi e-mail non recapitati inviati da tutte le esecuzioni della campagna, diviso il numero di messaggi e-mail inviati da tutte quelle esecuzioni della campagna.

Parametro	Kpi-name	Descrizione
Percentuale di mancati recapiti, raggruppati per esecuzione della campagna	`hard-bounce-rate-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, la percentuale di messaggi e-mail che non è stato possibile recapitare ai destinatari. Questa metrica misura solo le occorrenze di mancato recapito permanente, ovvero i messaggi in cui l'indirizzo e-mail del destinatario presenta un problema permanente che impedisce il recapito del messaggio. Questo parametro viene calcolato come il numero di messaggi e-mail non recapitati inviati da un'esecuzione della campagna, diviso il numero di messaggi e-mail inviati dall'esecuzione della campagna. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.

Parametro	Kpi-name	Descrizione
Percentuale di recapiti	`successful-delivery-rate`	Per ogni esecuzione della campagna, la percentua le di messaggi recapitati ai destinatari. Questo parametro viene calcolato come il numero di messaggi che sono stati inviati da tutte le esecuzioni della campagna e recapitati ai destinatari, diviso il numero di messaggi inviati da tutte quelle esecuzioni della campagna.

Parametro	Kpi-name	Descrizione
La percentuale di recapiti, raggruppati per l'esecuzione della campagna	`successful-delivery-rate-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, la percentuale di messaggi recapitati ai destinatari. Questo parametro viene calcolato come il numero di messaggi inviati da un'esecuzione della campagna e recapitati ai destinatari, diviso il numero di messaggi inviati tramite l'esecuzione della campagna. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.

Parametro	Kpi-name	Descrizione
Percentuale di recapiti, raggruppati per data	`successful-delivery-rate-grouped-by-date`	Per tutte le esecuzioni della campagna, la percentua le di messaggi recapitati ai destinatari durante ogni giorno nell'intervallo di date specifica to. Questo parametro viene calcolato come il numero di messaggi che sono stati inviati da tutte le esecuzioni della campagna e recapitati ai destinatari, diviso il numero di messaggi inviati da tutte quelle esecuzioni della campagna, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.

Parametro	Kpi-name	Descrizione
Percentuale di e-mail aperte	`email-open-rate`	Per tutte le esecuzioni della campagna, la percentuale di messaggi e-mail aperti dai destinatari. Questo parametro viene calcolato come il numero di messaggi e-mail inviati da tutte le esecuzioni della campagna e aperti dai destinatari, diviso il numero di messaggi e-mail inviati da tutte quelle esecuzioni della campagna e recapitati ai destinatari.

Parametro	Kpi-name	Descrizione
Percentuale di e-mail aperte, raggruppate per l'esecuzione della campagna	`email-open-rate-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, la percentuale di messaggi e-mail aperti dai destinatari. Questo parametro viene calcolato come il numero di messaggi e-mail inviati da un'esecuzione della campagna e aperti dai destinatari, diviso il numero di messaggi e-mail inviati dall'esecuzione della campagna e recapitati ai destinatari. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.

Parametro	Kpi-name	Descrizione
E-mail aperte, raggruppate per l'esecuzione della campagna	`direct-email-opens-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, il numero di messaggi e-mail aperti dai destinatari. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.
Consegne endpoint	`unique-deliveries`	Per tutte le esecuzioni della campagna, il numero di endpoint univoci ai quali sono stati consegnati i messaggi.
Consegne endpoint, raggruppate per l'esecuzione della campagna	`unique-deliveries-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, il numero di endpoint univoci ai quali sono stati consegnati i messaggi. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.

Parametro	Kpi-name	Descrizione
Consegne endpoint, raggruppate per data	`unique-deliveries-grouped-by-date`	Per tutte le esecuzioni della campagna, il numero di endpoint univoci ai quali sono stati consegnati i messaggi, per ogni giorno nell'intervallo di date specificato. I risultati della query per questo parametro sono raggruppati per giorno di calendario, nel formato esteso ISO 8601.
Collegamenti selezionati, raggruppati per l'esecuzione della campagna	`clicks-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, il numero di volte che i destinatari hanno fatto clic su collegamenti nel messaggio e-mail. Se un singolo destinatario ha fatto clic su più collegamenti nel messaggio o ha fatto clic sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.

Parametro	Kpi-name	Descrizione
Messaggi recapitati, raggruppati per l'esecuzione della campagna	`successful-deliveries-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, il numero di messaggi recapitati ai destinatari. Questo parametro viene calcolato come il numero di messaggi inviati da una esecuzione della campagna, meno il numero di messaggi non recapitati ai destinatari dell'esecuzione a causa di un hard bounce. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.

Parametro	Kpi-name	Descrizione
I messaggi inviati, raggruppati per l'esecuzione della campagna	`attempted-deliveries-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, il numero di messaggi che sono stati inviati. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.
Percentuale di notifiche push aperte	`push-open-rate`	Per ogni esecuzione della campagna, la percentuale di notifiche push aperte dai destinatari. Questo parametro viene calcolato in base al numero di notifiche push che sono state inviate da tutte le esecuzioni della campagna e aperte dai destinatari, diviso il numero di notifiche push che sono state inviate da tutte quelle esecuzioni della campagna e consegnate ai destinatari.

Parametro	Kpi-name	Descrizione
Percentuale di notifiche push aperte, raggruppate per l'esecuzione della campagna	`push-open-rate-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, la percentuale di notifiche push aperte dai destinatari. Questo parametro viene calcolato in base al numero di notifiche push che sono state inviate da un'esecuzione della campagna e aperte dai destinatari, diviso il numero di notifiche push che sono state inviate dall'esecuzione della campagna e consegnate ai destinatari. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.

Parametro	Kpi-name	Descrizione
Totale notifiche push aperte, raggruppate per l'esecuzione della campagna	`direct-push-opens-grouped-by-campaign-activity`	Per ogni esecuzione della campagna, il numero di notifiche push aperte dai destinatari. I risultati della query per questo parametro sono raggruppati per ID attività della campagna (`CampaignActivityId`), che è una stringa che identifica in modo univoco una esecuzione della campagna.
Spesa totale SMS	`sms-spend`	Per tutte le campagne, l'importo totale, in millesimi di centesimo, speso per l'invio di SMS.

Metriche di coinvolgimento del percorso Amazon Pinpoint

La tabella seguente elenca e descrive le metriche di coinvolgimento del percorso standard per cui è possibile eseguire query per monitorare le tendenze di tutti i messaggi e-mail inviati da un percorso Amazon Pinpoint. Per eseguire query sui dati relativi a queste metriche, utilizza la risorsa Metriche di coinvolgimento del percorso dell'API Amazon Pinpoint. Nella tabella, la colonna kpi-name indica il valore da utilizzare per il parametro `kpi-name` in una query.

Parametro	Kpi-name	Descrizione
Clic	`journey-emails-clicked`	Il numero di volte che i partecipanti hanno fatto clic sui collegamenti nei messaggi. Se un singolo partecipante ha fatto clic su più collegamenti in un messaggio o ha fatto clic

Parametro	Kpi-name	Descrizione
		sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio.
Clic, raggruppati per attività	`emails-clicked-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di volte in cui i partecipanti hanno fatto clic sui link nei messaggi. Se un singolo partecipante ha fatto clic su più collegamenti in un messaggio o ha fatto clic sullo stesso collegamento più di una volta, ogni clic è incluso nel conteggio. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.
Reclami	`journey-emails-complained`	Il numero di messaggi segnalati dai partecipanti come e-mail non richieste o indesiderate.

Parametro	Kpi-name	Descrizione
Reclami, raggruppati per attività	`emails-complained-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di messaggi segnalati dai partecipanti come e-mail non richieste o indesiderate. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.
Consegne	`journey-emails-delivered`	Il numero di messaggi consegnati ai partecipanti. Questo parametro viene calcolato come il numero di messaggi inviati, meno il numero di messaggi che non è stato possibile consegnare a causa di un soft bounce o di un hard bounce o perché sono stati rifiutati.

Parametro	Kpi-name	Descrizione
Consegne raggruppate per attività	`emails-delivered-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di messaggi che sono stati recapitati ai partecipanti. Questo parametro viene calcolato come il numero di messaggi inviati, meno il numero di messaggi che non è stato possibile consegnare a causa di un soft bounce o di un hard bounce o perché sono stati rifiutati, per ogni attività nel viaggio. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.
Mancati recapiti permanenti	`journey-emails-hardbounced`	Il numero di messaggi che non è stato possibile consegnare ai partecipanti a causa di un hard bounce. Un mancato recapito permanente si verifica se un problema persistente impedisce il recapito di un messaggio, ad esempio se l'indirizzo e-mail del partecipante non esiste.

Parametro	Kpi-name	Descrizione
Hard bounce, raggruppati per attività	`emails-hardbounced-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di messaggi che non possono essere recapitati ai partecipanti a causa di un hard bounce. Un mancato recapito permanente si verifica se un problema persistente impedisce il recapito di un messaggio, ad esempio se l'indirizzo e-mail del partecipante non esiste. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.
Aperture	`journey-emails-opened`	Mostra il numero di messaggi che sono stati aperti dai partecipanti:
Aperture, raggruppate per attività	`emails-opened-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di messaggi che sono stati aperti dai partecipanti. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.

Parametro	Kpi-name	Descrizione
Rifiuti	`journey-emails-rejected`	Il numero di messaggi non inviati ai partecipanti perché rifiutati. Un messaggio viene rifiutato se Amazon Pinpoint determina che il messaggio contiene malware. Amazon Pinpoint non tenta di inviare i messaggi rifiutati.
Rifiuti, raggruppati per attività	`emails-rejected-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di messaggi che non sono stati inviati ai partecipanti perché sono stati rifiutati. Un messaggio viene rifiutato se Amazon Pinpoint determina che il messaggio contiene malware. Amazon Pinpoint non tenta di inviare i messaggi rifiutati. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.
Invii	`journey-emails-sent`	Il numero di messaggi che sono stati inviati.

Parametro	Kpi-name	Descrizione
Invia, raggruppati per attività	`emails-sent-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di messaggi inviati. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.
Soft bounce	`journey-emails-softbounced`	Il numero di messaggi che non è stato possibile consegnare ai partecipanti a causa di un soft bounce. Si verifica un caso di e-mail non recapitata se un problema temporaneo impedisce il recapito di un messaggio, ad esempio se la posta in arrivo del partecipante è piena o il server di ricezione è temporaneamente non disponibile.

Parametro	Kpi-name	Descrizione
Soft bounce, raggruppati per attività	`emails-softbounced-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di messaggi che non è stato possibile recapitare ai partecipanti a causa di un soft bounce. Si verifica un caso di e-mail non recapitata se un problema temporaneo impedisce il recapito di un messaggio, ad esempio se la posta in arrivo del partecipante è piena o il server di ricezione è temporaneamente non disponibile. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.
Annullamenti di sottoscrizioni	`journey-emails-unsubscribed`	Il numero di volte in cui i partecipanti hanno fatto clic sui link di annullamento della sottoscrizione nei messaggi. Se un singolo partecipante ha fatto clic sullo stesso link di annullamento della sottoscrizione più di una volta, ogni clic è incluso nel conteggio.

Parametro	Kpi-name	Descrizione
Annullamenti delle sottoscri zioni, raggruppati per attività	`emails-unsubscribed-grouped-by-journey-activity`	Per ogni attività del viaggio, il numero di volte in cui i partecipanti hanno fatto clic sui link di annullamento della sottoscrizione nei messaggi. Se un singolo partecipante ha fatto clic sullo stesso link di annullamento della sottoscri zione più di una volta, ogni clic è incluso nel conteggio. I risultati della query per questo parametro sono raggruppati per ID attività (`JourneyActivityId`), che è una stringa che identifica in modo univoco un'attività.

Metriche di esecuzione del percorso Amazon Pinpoint

La tabella seguente elenca e descrive le metriche di esecuzione standard su cui è possibile eseguire query per valutare lo stato dei partecipanti in un percorso Amazon Pinpoint. Per eseguire query sui dati per questi metriche, utilizza la risorsa Metriche di esecuzione percorso dell'API Amazon Pinpoint. Nella colonna Field (Campo) nella tabella viene identificato il nome del campo visualizzato nei risultati della query per ogni parametro.

Parametro	Campo	Descrizione
Partecipanti attivi	`ENDPOINT_ACTIVE`	Il numero di partecipanti che stanno procedendo attivamen te nelle attività del viaggio. Questo parametro è calcolato come il numero di partecipanti che hanno iniziato il viaggio,

Parametro	Campo	Descrizione
		meno il numero di partecipanti che hanno lasciato il viaggio e il numero di partecipanti che sono stati rimossi dal viaggio.
Annullamenti dei partecipanti	`CANCELLED`	Il numero di partecipanti che non hanno completato il viaggio perché il viaggio è stato annullato.
Partenze partecipanti	`ENDPOINT_LEFT`	Il numero di partecipanti che hanno abbandonato il viaggio.
Ingressi partecipanti	`ENDPOINT_ENTERED`	Il numero di partecipanti che hanno iniziato il viaggio.
Eccezioni dei partecipanti, limiti di rientro	`REENTRY_CAP_EXCEEDED`	Il numero di partecipanti che non hanno completato il viaggio perché avrebbero superato il numero massimo di volte in cui un singolo partecipante può rientrare nel viaggio.

Parametro	Campo	Descrizione
Eccezioni dei partecipanti, rifiuti	`ACTIVE_ENDPOINT_REJECTED`	Il numero di partecipanti che non possono iniziare il viaggio perché sono già partecipanti attivi al viaggio. Un partecipante viene rifiutato se inizia un viaggio e in seguito la sua definizione di endpoint viene aggiornata in modo tale da compromettere la sua inclusione in un segmento (in base ai criteri del segmento) o nel viaggio (in base alle condizioni di attività).

Metriche di esecuzione delle attività del percorso Amazon Pinpoint

La tabella seguente elenca e descrive le metriche di esecuzione standard su cui è possibile eseguire query per valutare lo stato dei partecipanti in ogni tipo di attività individuale in un percorso Amazon Pinpoint. Per eseguire query sui dati per queste metriche, utilizza la risorsa Metriche di esecuzione delle attività del percorso dell'API Amazon Pinpoint. Nella colonna Metrics (Parametri) della tabella sono elencati i campi visualizzati nei risultati della query per ogni tipo di attività. Fornisce anche una breve descrizione di ciascun campo.

Tipo di attività	Metriche
Divisione/Nessuna divisione (`CONDITIONAL_SPLIT`)	I parametri sono: • `Branch_FALSE` _ numero di partecipanti che non hanno rispettato le condizioni dell'attività e che hanno continuato con l'attività sul percorso "No". • `Branch_TRUE` : numero di partecipanti che hanno rispettato le condizioni dell'attività e

Tipo di attività	Metriche
	hanno continuato con l'attività sul percorso "Sì". I parametri aggiuntivi sono disponibili per l'attività in ogni percorso. Per informazioni su tali parametri, vedere la riga in questa tabella per quel tipo di attività.
Controllo (`HOLDOUT`)	I parametri sono: • `HOLDOUT`: numero di partecipanti rimossi dal percorso come parte della percentuale di controllo per l'attività. • `PASSED`: numero di partecipanti trasferiti all'attività successiva nel percorso.

Tipo di attività	Metriche
E-mail (`MESSAGE`)	I parametri sono: • `DAILY_CAP_EXCEEDED` : numero di messaggi non inviati perché hanno superato il numero massimo di messaggi che un singolo partecipante può ricevere durante un periodo di 24 ore. • `FAILURE_PERMANENT` : numero di messaggi non inviati a causa di un problema permanente. • `QUIET_TIME` : numero di messaggi non inviati perché sono stati recapitati durante il periodo di riposo per il fuso orario del partecipante. • `SERVICE_FAILURE` : numero di messaggi non inviati a causa di un problema in Amazon Pinpoint. • `SUCCESS`: numero di messaggi recapitati correttamente ai partecipanti. • `THROTTLED` : numero di messaggi non inviati perché il relativo invio ha superato la quota di invio dell'account Amazon Pinpoint.. • `TRANSIENT_FAILURE` : numero di messaggi non inviati a causa di un problema temporaneo. • `UNKNOWN`: numero di messaggi non inviati a causa di un problema sconosciuto.

Tipo di attività	Metriche
Divisione multivariata (`MULTI_CONDITIONAL_SPLIT`)	Per ogni percorso dell'attività, il numero dei partecipanti che hanno proceduto all'attività sul percorso. I risultati delle query per questa metrica sono raggruppati per percorso, `Branch_#` dove si `#` trova l'identificatore numerico di un percorso, `Branch_1` ad esempio per il primo percorso dell'attività. I parametri aggiuntivi sono disponibili per l'attività in ogni percorso. Per informazioni su tali parametri, vedere la riga in questa tabella per quel tipo di attività.
Divisione casuale (`RANDOM_SPLIT`)	Per ogni percorso dell'attività, il numero dei partecipanti che hanno proceduto all'attività sul percorso. I risultati della query per questa metrica sono raggruppati per percorso, `Branch_# #` dov'è l'identificatore numerico di un percorso, ad esempio per il primo percorso dell'attività. `Branch_1` I parametri aggiuntivi sono disponibili per l'attività in ogni percorso. Per informazioni su tali parametri, vedere la riga in questa tabella per quel tipo di attività.

Tipo di attività	Metriche
Attesa (`WAIT`)	I parametri sono: • `WAIT_FINISHED` : numero di partecipanti che hanno terminato l'attesa per il periodo di tempo specificato. • `WAIT_SKIPPED` : numero di partecipanti che non hanno aspettato il periodo di tempo specificato, in genere perché hanno iniziato l'attività o il percorso dopo l'ora di fine pianificata per l'attività. • `WAIT_STARTED` : numero di partecipanti che hanno iniziato ad aspettare e che non hanno saltato o terminato l'attesa per il periodo di tempo specificato.

Tipo di attività	Metriche
Contact Center (`CONTACT_CENTER`)	I parametri sono: • `CALL_QUEUED` : numero di partecipanti che sono stati chiamati e messi in coda su Amazon Connect. Include i tentativi di ricomposizione. • `CONTINUE_WAITING` : numero di partecipanti che continuano ad attendere che vengano effettuati i tentativi di chiamata. • `DIAL_FAILURE` : numero di partecipanti con tentativi giornalieri non riusciti. • `DROPPED`: numero di partecipanti che non soddisfano più le condizioni definite nelle precedenti attività del percorso al momento della partenza. • `TIMEOUT`: numero di partecipanti che non hanno ricevuto alcun codice di disposizione di Amazon Connect dopo diversi tentativi di chiamata. • `WAIT_FINISHED` : numero di partecipanti che hanno terminato l'attesa per il periodo di tempo specificato. • `WAIT_FOR_QUIET_HOURS` : numero di partecipanti che attendono il termine del periodo di silenzio per poter effettuare il recapito sul canale. • `WAIT_STARTED` : numero di partecipanti che hanno iniziato ad aspettare e che non hanno saltato o terminato l'attesa per il periodo di tempo specificato.

Metriche del percorso e dell'esecuzione delle campagne di Amazon Pinpoint

Puoi eseguire query sulle metriche di esecuzione standard per valutare lo stato dei partecipanti a ogni tipo di attività individuale per un percorso o una campagna Amazon Pinpoint. Per eseguire query sui dati relativi a queste metriche, utilizza la risorsa Parametri di esecuzione attività percorso o Parametri della campagna dell'API Amazon Pinpoint. Nella seguente tabella sono elencati i campi visualizzati nei risultati della query per ogni tipo di attività.

Nome parametro	Si applica a percorsi, campagne o entrambi	Descrizione
ENDPOINT_PRODUCED	Entrambi	Numero di endpoint inizialmente prodotti dal segmento o dall'evento prima dell'applicazione di qualsiasi filtro.
ENDPOINTS_FROM_USER	Entrambi	Se il cliente ha un segmento dedicato solo agli ID utente, verranno aggiunti tutti gli endpoint di tali utenti. Questa metrica misura il numero di endpoint aggiunti in questo modo.
ENDPOINT_OPT_OUT	Entrambi	L'endpoint è stato escluso mediante opt-out e non è stato inserito nella campagna o nel percorso.
ENDPOINT_INACTIVE	Entrambi	L'endpoint è inattivo e non rientra nella campagna o nel percorso.
FILTERED_OUT_BY_SEGMENT	Entrambi	L'endpoint non corrisponde ai filtri dei segmenti e non rientra nella campagna o nel percorso.

Nome parametro	Si applica a percorsi, campagne o entrambi	Descrizione
ENDPOINT_MISSING_ADDRESS	Entrambi	All'endpoint manca un indirizzo e non rientra nella campagna o nel percorso.
ENDPOINT_MISSING_CHANNEL	Entrambi	All'endpoint manca un canale e non rientra nella campagna o nel percorso.
ENDPOINT_MISSING_TIMEZONE	Entrambi	All'endpoint manca un valore per il fuso orario ed è stato escluso. Ciò accade solo quando è richiesto un valore del fuso orario.
ENDPOINT_TIMEZONE_MISMATCH	Entrambi	L'endpoint si trova in un fuso orario attualmente non incluso nell'esecuzione.
ENDPOINT_CHANNEL_MISMATCH	Campagne	La campagna non ha un messaggio configurato per il tipo di canale dell'endpoint specifico.
DUPLICATE_ENDPOINT	Entrambi	Sono stati rilevati e deduplicati gli endpoint duplicati.
DUPLICATE_USER	Entrambi	Sono stati rilevati e deduplicati gli utenti duplicati da un segmento di soli ID utente. Se hanno lo stesso ID utente, verrà restituita una metrica pari a 1.

Nome parametro	Si applica a percorsi, campagne o entrambi	Descrizione
PAUSED	Percorsi	Metrica rimossa dall'esec uzione perché il percorso è stato sospeso.
ENDED	Percorsi	Metrica rimossa dall'esec uzione perché il percorso è terminato.
TREATMENT_HOLDOUT	Campagne	Metrica restituita nelle campagne A/B, per gli endpoint le cui coorti non corrispondono al trattamento attuale. Ad esempio, in una suddivisione A/B 50/50, il 50% degli endpoint restituirà questa metrica per ogni trattamento
ENDPOINT_ESTIMATED _TIMEZONE	Percorsi	La stima del fuso orario è stata in grado di stimare un fuso orario per l'endpoint.

Interroga i dati di analisi di Amazon Pinpoint per le campagne

Oltre a utilizzare le pagine di analisi sulla console Amazon Pinpoint, puoi utilizzare Amazon Pinpoint APIs Analytics per interrogare i dati di analisi per un sottoinsieme di metriche standard che forniscono informazioni sulle tendenze di distribuzione e coinvolgimento per le campagne.

Ognuno di questi parametri è un valore misurabile, noto anche come indicatore di prestazioni chiave (KPI), che consente di monitorare e valutare le prestazioni di una o più campagne. Ad esempio, è possibile utilizzare un parametro di campagna per accedere ai dati relativi al numero di endpoint ai quali è stato inviato un messaggio della campagna o quanti di questi messaggi sono stati recapitati agli endpoint previsti.

Amazon Pinpoint raccoglie e aggrega automaticamente questi dati per tutte le campagne. Memorizza i dati per 90 giorni. Se hai integrato un'app mobile con Amazon Pinpoint utilizzando un SDK AWS per

dispositivi mobili, Amazon Pinpoint estende questo supporto per includere parametri aggiuntivi, come la percentuale di notifiche push aperte dai destinatari. Per informazioni sull'integrazione di un'app per dispositivi mobili, consulta Integra Amazon Pinpoint con la tua applicazione.

Se utilizzi Amazon Pinpoint Analytics APIs per interrogare i dati, puoi scegliere varie opzioni che definiscono l'ambito, i dati, il raggruppamento e i filtri per la tua query. A tale scopo, utilizzare i parametri che specificano il progetto, la campagna e il parametro su cui eseguire una query, oltre a eventuali filtri basati sulla data che si desidera applicare.

In questo argomento vengono illustrati e forniti esempi di come scegliere queste opzioni ed eseguire query di dati di una o più campagne.

Prerequisiti

Prima di eseguire query sui dati di analisi di una o più campagne, consente di raccogliere le informazioni seguenti, che sono utilizzate per definire la query:

- ID progetto: identificatore univoco per il progetto associato alla campagna o alle campagne. Nell'API Amazon Pinpoint, questo valore viene archiviato nella proprietà `application-id`. Nella console Amazon Pinpoint, questo valore viene visualizzato nel campo ID progetto nella pagina Tutti i progetti.
- ID campagna: l'identificatore univoco per la campagna, se desideri eseguire query e i dati solo per una campagna. Nell'API Amazon Pinpoint, questo valore viene archiviato nella proprietà `campaign-id`. Questo valore non viene visualizzato sulla console.
- Intervallo di date: facoltativamente, la prima e l'ultima data e l'ora dell'intervallo di date in cui eseguire query sui dati. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, devono iniziare da meno di 90 giorni rispetto al giorno corrente. Se non viene specificato un intervallo di date, Amazon Pinpoint esegue automaticamente le query sui dati dei 31 giorni di calendario precedenti.
- Tipo di metrica: tipo di metrica a cui inviare le query. Esistono due tipi di parametri, parametri delle applicazioni e parametri delle campagne. Un parametro dell'applicazione fornisce i dati per tutte le campagne associate a un progetto, denominate anche applicazione. Un parametro della campagna fornisce dati per una sola campagna.
- Metrica: nome della metrica a cui inviare le query, in particolare il valore `kpi-name` della metrica. Per un elenco completo dei parametri supportati e per il valore `kpi-name`di ognuno, consulta Metriche standard per progetti, campagne e viaggi.

Aiuta anche a stabilire se si desidera raggruppare i dati in base a un campo pertinente. In tal caso, è possibile semplificare l'analisi e la creazione di report scegliendo un parametro progettato per raggruppare automaticamente i dati. Ad esempio, in Amazon Pinpoint sono disponibili diverse metriche standard che riportano la percentuale di messaggi recapitati ai destinatari di una campagna. Uno di questi parametri raggruppa automaticamente i dati per data (`successful-delivery-rate-grouped-by-date`). Un altro parametro raggruppa automaticamente i dati per esecuzione di campagna (`successful-delivery-rate-grouped-by-campaign-activity`). Una terza metrica restituisce semplicemente un unico valore, ovvero la percentuale di messaggi recapitati ai destinatari da tutte le esecuzioni delle campagne (`successful-delivery-rate`).

Se non è possibile trovare un parametro standard che raggruppi i dati nel modo desiderato, è possibile sviluppare una serie di query in grado di restituire i dati desiderati e quindi suddividere manualmente o combinare i risultati della query in gruppi personalizzati progettati.

Infine, è importante verificare di essere autorizzati ad accedere ai dati su cui si desidera eseguire una query. Per ulteriori informazioni, consulta Policy IAM per l'esecuzione di query sui dati di analisi di Amazon Pinpoint.

Interroga i dati di Amazon Pinpoint per una campagna

Per eseguire una query sui dati di una campagna, è possibile utilizzare l'API Parametri della campagna e specificare i valori per i seguenti parametri richiesti:

- application-id: ID del progetto, ovvero l'identificatore univoco del progetto associato alla campagna. In Amazon Pinpoint, i termini progetto e applicazione hanno lo stesso significato.
- campaign-id: identificatore univoco della campagna.
- kpi-name: nome della metrica a cui inviare la query. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Per un elenco completo dei parametri supportati e per il valore `kpi-name`di ognuno, consulta Metriche standard per progetti, campagne e viaggi.

È inoltre possibile applicare un filtro che esegue le query sui dati di un intervallo di date specifico. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. Per filtrare i dati in base a date diverse, utilizzare i parametri dell'intervallo di date supportati per specificare la prima e l'ultima data e l'ora dell'intervallo di date. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio `2019-07-19T20:00:00Z` per le 20:00 UTC del 19 luglio 2019. Gli intervalli di date includono gli

estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni.

Gli esempi seguenti mostrano come interrogare i dati di analisi per una campagna utilizzando l'API REST di Amazon Pinpoint AWS CLI, il e il. AWS SDK per Java Puoi utilizzare qualsiasi AWS SDK supportato per interrogare i dati di analisi per una campagna. Gli AWS CLI esempi sono formattati per Microsoft Windows. Per Unix, Linux e macOS, sostituire il carattere di continuazione della riga caret (^) con una barra rovesciata ().

REST API

Per eseguire query sui dati di analisi per una campagna tramite la REST API Amazon Pinpoint, invia una richiesta HTTP(S) GET all'URI Parametri della campagna. Nell'URI, specificare i valori appropriati per i parametri del percorso richiesti:

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

Dove:

- *endpoint*è l'endpoint Amazon Pinpoint per la AWS regione che ospita il progetto associato alla campagna.
- *application-id* è l'identificatore univoco del progetto associato alla campagna.
- *campaign-id* è l'identificatore univoco per la campagna.
- *kpi-name*è il `kpi-name` valore della metrica da interrogare.

Tutti i parametri devono essere codificati URL.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, aggiungere i parametri di query e i valori `start-time` e `end-time` all'URI. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Utilizzare una e commerciale (&) per separare i parametri.

Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le esecuzioni di una campagna, dal 19 luglio 2019 al 26 luglio 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Dove:

- `pinpoint.us-east-1.amazonaws.com` è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto.
- `12345678901234567890123example` è l'identificatore univoco del progetto associato alla campagna.
- `80b8efd84042ff8d9c96ce2f8example` è l'identificatore univoco per la campagna.
- `unique-deliveries` è il valore `kpi-name` del parametro della campagna recapiti endpoint, ovvero il parametro che segnala il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutti i cicli di una campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-07-26T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

AWS CLI

Per interrogare i dati di analisi per una campagna utilizzando il AWS CLI, utilizza il get-campaign-date-range-kpi comando e specifica i valori appropriati per i parametri richiesti:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id application-id ^
    --campaign-id campaign-id ^
    --kpi-name kpi-name
```

Dove:

- *`application-id`* è l'identificatore univoco del progetto associato alla campagna.
- *`campaign-id`* è l'identificatore univoco per la campagna.
- *`kpi-name`*è il `kpi-name` valore della metrica da interrogare.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, aggiungere i parametri e i valori `end-time` e `start-time` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la richiesta seguente recupera il numero di endpoint

univoci a cui sono stati recapitati i messaggi, da tutte le esecuzioni di una campagna, dal 19 luglio 2019 al 26 luglio 2019:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --campaign-id 80b8efd84042ff8d9c96ce2f8example ^
    --kpi-name unique-deliveries ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Dove:

- `12345678901234567890123456example` è l'identificatore univoco del progetto associato alla campagna.
- `80b8efd84042ff8d9c96ce2f8example` è l'identificatore univoco per la campagna.
- `unique-deliveries` è il valore `kpi-name` del parametro della campagna recapiti endpoint, ovvero il parametro che segnala il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutti i cicli di una campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-07-26T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

SDK for Java

Per interrogare i dati di analisi per una campagna utilizzando AWS SDK per Java, utilizza il GetCampaignDateRangeKpiRequest metodo dell'API Campaign Metrics. Specificare i valori appropriati per i parametri richiesti:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withCampaignId("campaignId")
        .withKpiName("kpiName")
```

Dove:

- *`applicationId`* è l'identificatore univoco del progetto associato alla campagna.
- *`campaignId`* è l'identificatore univoco per la campagna.

- *kpiName*è il `kpi-name` valore della metrica da interrogare.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, includere i parametri e i valori `endTime` e `startTime` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le esecuzioni di una campagna, dal 19 luglio 2019 al 26 luglio 2019:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456789012345example")
        .withCampaignId("80b8efd84042ff8d9c96ce2f8example")
        .withKpiName("unique-deliveries")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Dove:

- `12345678901234567890123456789012345example` è l'identificatore univoco del progetto associato alla campagna.
- `80b8efd84042ff8d9c96ce2f8example` è l'identificatore univoco per la campagna.
- `unique-deliveries` è il valore `kpi-name` del parametro della campagna recapiti endpoint, ovvero il parametro che segnala il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutti i cicli di una campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-07-26T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. La struttura dei risultati varia a seconda del parametro interrogato. Alcuni parametri restituiscono un solo valore. Ad esempio, la metrica della campagna Recapiti agli endpoint (`unique-deliveries`), utilizzata negli esempi precedenti, restituisce un valore, ovvero il numero di endpoint univoci a cui sono stati recapitati i messaggi, per tutte le esecuzioni di una campagna. In questo caso, la risposta JSON è la seguente:

```
{
```

```
    "CampaignDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "CampaignId":"80b8efd84042ff8d9c96ce2f8example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```


Altri parametri restituiscono più valori e raggruppano tali valori in base a un campo pertinente. Se un parametro restituisce più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati.

Per ulteriori informazioni sulla struttura dei risultati delle query, consulta Usa i risultati delle query JSON.

Interroga i dati di Amazon Pinpoint per più campagne

È possibile eseguire query sui dati per più campagne in due modi. Il modo migliore dipende dal fatto che si desidera eseguire una query sui dati per le campagne che sono tutte associate allo stesso progetto. In tal caso, dipende anche dal fatto che si desidera eseguire una query sui dati per tutte o solo per un sottoinsieme di tali campagne.

Per eseguire una query sui dati delle campagne associate a progetti diversi o solo sui dati di un sottoinsieme delle campagne associate allo stesso progetto, l'approccio migliore consiste nel creare ed eseguire una serie di query individuali, una per ogni campagna sui cui dati si desidera eseguire una query. Nella sezione precedente viene illustrato come eseguire una query sui dati per una sola campagna.

Per eseguire una query sui dati di tutte le campagne associate allo stesso progetto, è possibile utilizzare l'API Parametri applicazioni. Specificare i valori per i parametri obbligatori seguenti.

- application-id: ID del progetto, che è l'identificatore univoco del progetto. In Amazon Pinpoint, i termini progetto e applicazione hanno lo stesso significato.
- kpi-name: nome della metrica a cui inviare la query. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Per un elenco completo dei parametri supportati e per il valore `kpi-name`di ognuno, consulta Metriche standard per progetti, campagne e viaggi.

È inoltre possibile filtrare i dati in base all'intervallo di date. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. Per filtrare i dati in base a date diverse, utilizzare i parametri dell'intervallo di date supportati per specificare la prima e l'ultima data e l'ora dell'intervallo di date. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio `2019-07-19T20:00:00Z` per le 20:00 UTC del 19 luglio 2019. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni.

Gli esempi seguenti mostrano come interrogare i dati di analisi per una campagna utilizzando l'API REST di Amazon Pinpoint AWS CLI, il e il. AWS SDK per Java Puoi utilizzare qualsiasi AWS SDK supportato per interrogare i dati di analisi per una campagna. Gli AWS CLI esempi sono formattati per Microsoft Windows. Per Unix, Linux e macOS, sostituire il carattere di continuazione della riga caret (^) con una barra rovesciata ().

REST API

Per eseguire query sui dati di analisi per più campagne utilizzando la REST API Amazon Pinpoint, invia una richiesta HTTP(S) GET all'URI Metriche delle applicazioni. Nell'URI, specificare i valori appropriati per i parametri del percorso richiesti:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dove:

- *`endpoint`*è l'endpoint Amazon Pinpoint per la AWS regione che ospita il progetto associato alle campagne.
- *`application-id`* è l'identificatore univoco per il progetto associato alle campagne.

- *kpi-name*è il kpi-name valore della metrica da interrogare.

Tutti i parametri devono essere codificati URL.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, aggiungere i parametri di query e i valori start-time e end-time all'URI. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Utilizzare una e commerciale (&) per separare i parametri.

Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le campagne di un progetto, dal 19 luglio 2019 al 26 luglio 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456789012345example/
kpis/daterange/unique-deliveries-grouped-by-campaign?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Dove:

- pinpoint.us-east-1.amazonaws.com è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto.
- 12345678901234567890123456789012345example è l'identificatore univoco per il progetto associato alle campagne.
- unique-deliveries-grouped-by-campaign è il valore kpi-name per il parametro dell'applicazione recapiti endpoint, raggruppati per campagna ovvero il parametro che restituisce il numero di endpoint univoci a cui sono stati recapitati i messaggi per ciascuna campagna.
- 2019-07-19T00:00:00Z rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- 2019-07-26T23:59:59Z rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

AWS CLI

Per interrogare i dati di analisi per più campagne utilizzando il AWS CLI, utilizza il get-application-date-range-kpi comando e specifica i valori appropriati per i parametri richiesti:

```
C:\> aws pinpoint get-application-date-range-kpi ^
```

```
    --application-id application-id ^
    --kpi-name kpi-name
```

Dove:

- *application-id* è l'identificatore univoco per il progetto associato alle campagne.
- *kpi-name*è il `kpi-name` valore della metrica da interrogare.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, includere i parametri e i valori `end-time` e `start-time` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le campagne di un progetto, dal 19 luglio 2019 al 26 luglio 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name unique-deliveries-grouped-by-campaign ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Dove:

- `12345678901234567890123456example` è l'identificatore univoco del progetto associato alla campagna.
- `unique-deliveries-grouped-by-campaign` è il valore `kpi-name` per il parametro dell'applicazione recapiti endpoint, raggruppati per campagna ovvero il parametro che restituisce il numero di endpoint univoci a cui sono stati recapitati i messaggi per ciascuna campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-07-26T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

SDK for Java

Per interrogare i dati di analisi per più campagne utilizzando il AWS SDK per Java, utilizza il GetApplicationDateRangeKpiRequest metodo dell'API Application Metrics. Specificare i valori appropriati per i parametri richiesti:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Dove:

- *applicationId* è l'identificatore univoco per il progetto associato alle campagne.
- *kpiName*è il `kpi-name` valore della metrica da interrogare.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, includere i parametri e i valori `endTime` e `startTime` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la richiesta seguente recupera il numero di endpoint univoci a cui sono stati recapitati i messaggi, da tutte le campagne di un progetto, dal 19 luglio 2019 al 26 luglio 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("unique-deliveries-grouped-by-campaign")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Dove:

- `12345678901234567890123456example` è l'identificatore univoco per il progetto associato alle campagne.
- `unique-deliveries-grouped-by-campaign` è il valore `kpi-name` per il parametro dell'applicazione recapiti endpoint, raggruppati per campagna ovvero il parametro che restituisce il numero di endpoint univoci a cui sono stati recapitati i messaggi per ciascuna campagna.
- `2019-07-19T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.

- `2019-07-26T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. La struttura dei risultati varia a seconda del parametro interrogato. Alcuni parametri restituiscono un solo valore. Altri parametri restituiscono più valori e quei valori sono raggruppati in base a un campo pertinente. Se un parametro restituisce più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati.

Ad esempio, la metrica Recapiti agli endpoint, raggruppati per campagna (`unique-deliveries-grouped-by-campaign`), utilizzata negli esempi precedenti, restituisce più valori, ovvero il numero di endpoint univoci a cui sono stati recapitati i messaggi, per ogni campagna associata a un progetto. In questo caso, la risposta JSON è la seguente:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
```

```
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```

In questo caso, il campo `GroupedBys` indica che i valori sono raggruppati per ID campagna (`CampaignId`).

Per ulteriori informazioni sulla struttura dei risultati delle query, consulta Usa i risultati delle query JSON.

Interroga i dati di analisi di Amazon Pinpoint per i messaggi transazionali

Oltre a utilizzare le pagine di analisi sulla console Amazon Pinpoint, puoi utilizzare Amazon Pinpoint APIs Analytics per interrogare i dati di analisi per un sottoinsieme di metriche standard che forniscono informazioni sulle tendenze di consegna e coinvolgimento per i messaggi transazionali inviati per un progetto.

Ognuno di questi parametri è un valore misurabile, noto anche come indicatore di prestazioni chiave (KPI), che consente di monitorare e valutare le prestazioni di questi messaggi transazionali. Ad esempio, è possibile utilizzare una metrica per scoprire il numero di messaggi e-mail o SMS transazionali inviati o quanti di questi messaggi sono stati recapitati ai destinatari. Amazon Pinpoint raccoglie e aggrega automaticamente questi dati per tutti i messaggi e-mail e SMS transazionali inviati per un progetto. Memorizza i dati per 90 giorni.

Se utilizzi Amazon Pinpoint Analytics APIs per interrogare i dati, puoi scegliere varie opzioni che definiscono l'ambito, i dati, il raggruppamento e i filtri per la tua query. A tale scopo, utilizza i parametri che specificano il progetto e il parametro su cui eseguire una query, oltre a eventuali filtri basati sulla data che desideri applicare.

Questo argomento illustra e fornisce esempi di come scegliere queste opzioni ed esegue query sui dati di messaggistica transazionale per un progetto.

Prerequisiti

Prima di eseguire query sui dati di analisi per i messaggi transazionali, consente di raccogliere le informazioni seguenti, che vengono utilizzate per definire la query:

- ID progetto: identificatore univoco per il progetto da cui sono stati inviati i messaggi. Nell'API Amazon Pinpoint, questo valore viene archiviato nella proprietà `application-id`. Nella console Amazon Pinpoint, questo valore viene visualizzato nel campo ID progetto nella pagina Tutti i progetti.
- Intervallo di date: facoltativamente, la prima e l'ultima data e l'ora dell'intervallo di date in cui eseguire query sui dati. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, devono iniziare da meno di 90 giorni rispetto al giorno corrente. Se non viene specificato un intervallo di date, Amazon Pinpoint esegue automaticamente le query sui dati dei 31 giorni di calendario precedenti.

- Metrica: nome della metrica a cui inviare le query, in particolare il valore `kpi-name` della metrica. Per un elenco completo dei parametri supportati e per il valore `kpi-name`di ognuno, consulta Metriche standard per progetti, campagne e viaggi.

Aiuta anche a stabilire se si desidera raggruppare i dati in base a un campo pertinente. In tal caso, è possibile semplificare l'analisi e la creazione di report scegliendo un parametro progettato per raggruppare automaticamente i dati. Ad esempio, in Amazon Pinpoint sono disponibili diverse metriche standard che restituiscono il numero di messaggi SMS transazionali recapitati ai destinatari. Uno di questi parametri raggruppa automaticamente i dati per data (`txn-sms-delivered-grouped-by-date`). Un altro parametro raggruppa automaticamente i dati per paese o regione (`txn-sms-delivered-grouped-by-country`). Una terza metrica restituisce semplicemente un unico valore, ovvero il numero di messaggi recapitati ai destinatari (`txn-sms-delivered`). Se non è possibile trovare un parametro standard che raggruppi i dati nel modo desiderato, è possibile sviluppare una serie di query in grado di restituire i dati desiderati e quindi suddividere manualmente o combinare i risultati della query in gruppi personalizzati progettati.

Infine, è importante verificare di essere autorizzati ad accedere ai dati su cui si desidera eseguire una query. Per ulteriori informazioni, consulta Policy IAM per l'esecuzione di query sui dati di analisi di Amazon Pinpoint.

Interroga i dati di Amazon Pinpoint per i messaggi e-mail transazionali

Per eseguire una query sui dati relativi ai messaggi e-mail transazionali inviati per un progetto, è possibile utilizzare l'API Parametri applicazioni e specificare i valori per i seguenti parametri richiesti:

- application-id: ID del progetto, che è l'identificatore univoco del progetto. In Amazon Pinpoint, i termini progetto e applicazione hanno lo stesso significato.
- kpi-name: nome della metrica a cui inviare la query. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Per un elenco completo dei parametri supportati e per il valore `kpi-name`di ognuno, consulta Metriche standard per progetti, campagne e viaggi.

È inoltre possibile applicare un filtro che esegue le query sui dati di un intervallo di date specifico. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. Per filtrare i dati in base a date diverse, utilizzare i parametri dell'intervallo di date supportati per specificare la prima e l'ultima data e l'ora dell'intervallo di date. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio

`2019-09-06T20:00:00Z` per le 20:00 UTC del 6 settembre 2019. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni.

Gli esempi seguenti mostrano come interrogare i dati di analisi per i messaggi e-mail transazionali utilizzando l'API REST di Amazon Pinpoint, il e AWS CLI il. AWS SDK per JavaÈ possibile utilizzare qualsiasi SDK AWS supportato per eseguire query sui dati di analisi dei messaggi transazionali. Gli AWS CLI esempi sono formattati per Microsoft Windows. Per Unix, Linux e macOS, sostituire il carattere di continuazione della riga caret (^) con una barra rovesciata ().

REST API

Per eseguire query sui dati di analisi per i messaggi e-mail transazionali tramite la REST API Amazon Pinpoint, invia una richiesta HTTP(S) GET all'URI Metriche delle applicazioni. Nell'URI, specificare i valori appropriati per i parametri del percorso richiesti:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dove:

- *endpoint*è l'endpoint Amazon Pinpoint per la AWS regione che ospita il progetto.
- *application-id* è l'identificatore univoco per il progetto.
- *kpi-name*è il `kpi-name` valore della metrica da interrogare.

Tutti i parametri devono essere codificati URL.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, aggiungere i parametri di query e i valori `start-time` e `end-time` all'URI. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Utilizzare una e commerciale (&) per separare i parametri.

Ad esempio, la seguente richiesta recupera il numero di messaggi e-mail transazionali inviati per un progetto dal 6 settembre 2019 al 13 settembre 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-emails-sent?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-13T23:59:59Z
```

Dove:

- `pinpoint.us-east-1.amazonaws.com` è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto.
- `12345678901234567890123456example` è l'identificatore univoco per il progetto.
- `txn-emails-sent` è il valore `kpi-name` del parametro dell'applicazione invii ovvero il parametro segnala il numero di messaggi e-mail transazionali inviati per un progetto.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-13T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

AWS CLI

Per interrogare i dati di analisi per i messaggi e-mail transazionali utilizzando il AWS CLI, utilizzate il get-application-date-range-kpi comando e specificate i valori appropriati per i parametri richiesti:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Dove:

- *`application-id`* è l'identificatore univoco per il progetto.
- *`kpi-name`*è il `kpi-name` valore della metrica da interrogare.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, aggiungere i parametri e i valori `end-time` e `start-time` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la seguente richiesta recupera il numero di messaggi e-mail transazionali inviati per un progetto dal 6 settembre 2019 al 13 settembre 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-emails-sent ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-13T23:59:59Z
```

Dove:

- `12345678901234567890123456example` è l'identificatore univoco per il progetto.
- `txn-emails-sent` è il valore `kpi-name` del parametro dell'applicazione invii ovvero il parametro segnala il numero di messaggi e-mail transazionali inviati per un progetto.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-13T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

SDK for Java

Per interrogare i dati di analisi per i messaggi e-mail transazionali utilizzando il AWS SDK per Java, utilizza il GetApplicationDateRangeKpiRequest metodo dell'API Application Metrics. Specificare i valori appropriati per i parametri richiesti:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Dove:

- *`applicationId`* è l'identificatore univoco per il progetto.
- *`kpiName`*è il `kpi-name` valore della metrica da interrogare.

Per applicare un filtro che interroga i dati per un intervallo di date specifico, includere i parametri e i valori `endTime` e `startTime` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la seguente richiesta recupera il numero di messaggi e-mail transazionali inviati per un progetto dal 6 settembre 2019 al 13 settembre 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-emails-sent")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

Dove:

- `12345678901234567890123456example` è l'identificatore univoco per il progetto.

- `txn-emails-sent` è il valore `kpi-name` del parametro dell'applicazione invii ovvero il parametro segnala il numero di messaggi e-mail transazionali inviati per un progetto.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-13T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. La struttura dei risultati varia a seconda del parametro interrogato. Alcuni parametri restituiscono un solo valore. Ad esempio, la metrica Invii (`txn-emails-sent`), utilizzata negli esempi precedenti, restituisce un valore, ovvero il numero di messaggi e-mail transazionali inviati da un progetto. In questo caso, la risposta JSON è la seguente:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-13T23:59:59Z",
        "KpiName":"txn-emails-sent",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"TxnEmailsSent",
                            "Type":"Double",
                            "Value":"62.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

Altri parametri restituiscono più valori e raggruppano tali valori in base a un campo pertinente. Se un parametro restituisce più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati.

Per ulteriori informazioni sulla struttura dei risultati delle query, consulta Usa i risultati delle query JSON.

Interroga i dati di Amazon Pinpoint per i messaggi SMS transazionali

Per eseguire una query sui dati relativi ai messaggi SMS transazionali inviati per un progetto, è possibile utilizzare l'API Parametri applicazioni e specificare i valori per i seguenti parametri richiesti:

- application-id: ID del progetto, che è l'identificatore univoco del progetto. In Amazon Pinpoint, i termini progetto e applicazione hanno lo stesso significato.
- kpi-name: nome della metrica a cui inviare la query. Questo valore descrive il parametro associato ed è formato da due o più termini, che sono composti da caratteri alfanumerici minuscoli, separati da un trattino. Per un elenco completo dei parametri supportati e per il valore `kpi-name`di ognuno, consulta Metriche standard per progetti, campagne e viaggi.

È inoltre possibile applicare un filtro che esegue le query sui dati di un intervallo di date specifico. Se non si specifica un intervallo di date, Amazon Pinpoint restituisce i dati dei 31 giorni di calendario precedenti. Per filtrare i dati in base a date diverse, utilizzare i parametri dell'intervallo di date supportati per specificare la prima e l'ultima data e l'ora dell'intervallo di date. I valori devono essere in formato ISO 8601 esteso e utilizzare il fuso orario UTC (Coordinated Universal Time), ad esempio `2019-09-06T20:00:00Z` per le 20:00 UTC del 6 settembre 2019. Gli intervalli di date includono gli estremi e devono essere limitati a 31 o a un numero minore di giorni di calendario. Inoltre, la prima data e la prima ora non devono precedere il giorno corrente di più di 90 giorni.

Gli esempi seguenti mostrano come interrogare i dati di analisi per i messaggi SMS transazionali utilizzando l'API REST di Amazon Pinpoint, il e AWS CLI il. AWS SDK per Java Puoi utilizzare qualsiasi AWS SDK supportato per interrogare i dati di analisi per i messaggi transazionali. Gli AWS CLI esempi sono formattati per Microsoft Windows. Per Unix, Linux e macOS, sostituire il carattere di continuazione della riga caret (^) con una barra rovesciata ().

REST API

Per eseguire query sui dati di analisi per i messaggi SMS transazionali tramite la REST API Amazon Pinpoint, invia una richiesta HTTP(S) GET all'URI Metriche delle applicazioni. Nell'URI, specificare i valori appropriati per i parametri del percorso richiesti:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Dove:

- *endpoint*è l'endpoint Amazon Pinpoint per la AWS regione che ospita il progetto.
- *application-id* è l'identificatore univoco per il progetto.
- *kpi-name*è il `kpi-name` valore della metrica da interrogare.

Tutti i parametri devono essere codificati URL.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, aggiungere i parametri di query e i valori `start-time` e `end-time` all'URI. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Utilizzare una e commerciale (&) per separare i parametri.

Ad esempio, la seguente richiesta recupera il numero di messaggi SMS transazionali inviati ogni giorno dal 6 settembre 2019 all'8 settembre 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-08T23:59:59Z
```

Dove:

- `pinpoint.us-east-1.amazonaws.com` è l'endpoint Amazon Pinpoint per la regione AWS che ospita il progetto.
- `12345678901234567890123456example` è l'identificatore univoco per il progetto.
- `txn-sms-sent-grouped-by-date` è il valore `kpi-name` del parametro dell'applicazione invii, raggruppati per data ovvero il parametro che restituisce il numero di messaggi SMS transazionali inviati ogni giorno nell'intervallo di date indicato.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-08T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

AWS CLI

Per interrogare i dati di analisi per i messaggi SMS transazionali utilizzando il AWS CLI, utilizzate il get-application-date-range-kpi comando e specificate i valori appropriati per i parametri richiesti:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Dove:

- *application-id* è l'identificatore univoco per il progetto.
- *kpi-name*è il `kpi-name` valore della metrica da interrogare.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, includere i parametri e i valori `end-time` e `start-time` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la seguente richiesta recupera il numero di messaggi SMS transazionali inviati ogni giorno dal 6 settembre 2019 all'8 settembre 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-sms-sent-grouped-by-date ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-08T23:59:59Z
```

Dove:

- `12345678901234567890123456example` è l'identificatore univoco per il progetto.
- `txn-sms-sent-grouped-by-date` è il valore `kpi-name` del parametro dell'applicazione invii, raggruppati per data ovvero il parametro che restituisce il numero di messaggi SMS transazionali inviati ogni giorno nell'intervallo di date indicato.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-08T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

SDK for Java

Per interrogare i dati di analisi per i messaggi SMS transazionali utilizzando il AWS SDK per Java, utilizza il GetApplicationDateRangeKpiRequest metodo dell'API Application Metrics e specifica i valori appropriati per i parametri richiesti:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Dove:

- *applicationId* è l'identificatore univoco per il progetto.
- *kpiName*è il `kpi-name` valore della metrica da interrogare.

Per applicare un filtro che richiama i dati per un intervallo di date specifico, includere i parametri e i valori `endTime` e `startTime` alla query. Utilizzando questi parametri, è possibile specificare la prima e l'ultima data e l'ora, in formato ISO 8601 esteso, di un intervallo di date inclusivo per il recupero dei dati. Ad esempio, la seguente richiesta recupera il numero di messaggi SMS transazionali inviati ogni giorno dal 6 settembre 2019 all'8 settembre 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-sms-sent-grouped-by-date")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-08T23:59:59Z")));
```

Dove:

- `12345678901234567890123456example` è l'identificatore univoco per il progetto.
- `txn-sms-sent-grouped-by-date` è il valore `kpi-name` del parametro dell'applicazione invii, raggruppati per data ovvero il parametro che restituisce il numero di messaggi SMS transazionali inviati ogni giorno nell'intervallo di date indicato.
- `2019-09-06T00:00:00Z` rappresenta la prima data e ora per il recupero dei dati, come parte di un intervallo di date inclusivo.
- `2019-09-08T23:59:59Z` rappresenta l'ultima data e l'ora in cui recuperare i dati, come parte di un intervallo di date inclusivo.

Dopo aver inviato la query, Amazon Pinpoint restituisce i risultati della query in una risposta JSON. La struttura dei risultati varia a seconda del parametro interrogato. Alcuni parametri restituiscono un solo valore. Altri parametri restituiscono più valori e raggruppano tali valori in base a un campo pertinente. Se un parametro restituisce più valori, la risposta JSON include un campo che indica quale campo è stato utilizzato per raggruppare i dati.

Ad esempio, la metrica Invii, raggruppati per data (`txn-sms-sent-grouped-by-date`), utilizzata negli esempi precedenti, restituisce più valori, ovvero il numero di messaggi SMS transazionali inviati durante ogni giorno dell'intervallo di date specificato. In questo caso, la risposta JSON è la seguente:


```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-08T23:59:59Z",
        "KpiName":"txn-sms-sent-grouped-by-date",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
                            "Type":"String",
                            "Value":"2019-09-06"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"TxnSmsSent",
                            "Type":"Double",
                            "Value":"29.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
                            "Type":"String",
                            "Value":"2019-09-07"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"TxnSmsSent",
                            "Type":"Double",
                            "Value":"35.0"
                        }
                    ]
                },
```

```
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
                            "Type":"String",
                            "Value":"2019-09-08"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"TxnSmsSent",
                            "Type":"Double",
                            "Value":"10.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

In questo caso, il campo `GroupedBys` indica che i valori sono raggruppati per giorno di calendario (`Date`). Ciò significa che:

- 29 messaggi sono stati inviati il 6 settembre 2019.
- 35 messaggi sono stati inviati il 7 settembre 2019.
- 10 messaggi sono stati inviati l'8 settembre 2019.

Per ulteriori informazioni sulla struttura dei risultati delle query, consulta Usa i risultati delle query JSON.

Usa i risultati delle query JSON di Amazon Pinpoint Analytics

Quando usi Amazon Pinpoint Analytics APIs per interrogare i dati di analisi, Amazon Pinpoint restituisce i risultati in una risposta JSON. Per le metriche delle applicazioni, le metriche delle campagne e le metriche di coinvolgimento del percorso, i dati nella risposta sono conformi a uno schema JSON standard per la creazione di report di dati di analisi di Amazon Pinpoint.

Ciò significa che è possibile utilizzare il linguaggio di programmazione o lo strumento di propria scelta per implementare una soluzione personalizzata che esegua query sui dati per uno o più di questi parametri, acquisisca i risultati di ogni query e quindi scriva i risultati in una tabella, un oggetto o un'altra posizione. È quindi possibile utilizzare i risultati delle query in tale posizione utilizzando un altro servizio o applicazione.

Ad esempio, puoi:

- Creare un dashboard personalizzato che esegua regolarmente le query su un set di parametri e visualizzi i risultati utilizzando il framework di visualizzazione dei dati preferito.
- Creare un report che tenga traccia dei tassi di coinvolgimento eseguendo una query sui parametri appropriati e visualizzando i risultati in un grafico o in un altro tipo di report che si progetta.
- Analizzare e scrivere i dati di analisi in un particolare formato di storage e quindi eseguire il port dei risultati in una soluzione di storage a lungo termine.

Tieni presente che Amazon Pinpoint Analytics APIs non è progettato per creare o archiviare oggetti persistenti da leggere o utilizzare successivamente in un progetto Amazon Pinpoint o nel tuo account Amazon Pinpoint. APIs Sono invece progettati per aiutarti a recuperare dati di analisi e trasferirli ad altri servizi e applicazioni per ulteriori analisi, archiviazione o report. Lo fanno in parte utilizzando la stessa struttura di risposta JSON e lo stesso schema per tutti i dati di analisi su cui è possibile eseguire query in modo programmatico per i parametri dell'applicazione, i parametri della campagna e i parametri di coinvolgimento del viaggio.

In questo argomento vengono illustrati la struttura, gli oggetti e i campi in una risposta JSON a una query per un parametro dell'applicazione, un parametro della campagna o un parametro di coinvolgimento del viaggio. Per informazioni sui campi in una risposta JSON a una query relativa a un parametro di esecuzione del viaggio o a un parametro di esecuzione dell'attività di viaggio, consulta Metriche standard che si applicano a progetti, campagne e percorsi Amazon Pinpoint.

Struttura JSON

Per aiutarti ad analizzare e utilizzare i risultati delle query, Amazon Pinpoint APIs Analytics utilizza la stessa struttura di risposta JSON per tutti i dati di analisi di Amazon Pinpoint che puoi interrogare a livello di codice per le metriche delle applicazioni, le metriche delle campagne e le metriche di coinvolgimento del viaggio. Ogni risposta JSON specifica i valori che hanno definito la query, ad esempio l'ID del progetto (`ApplicationId`). La risposta include anche un (e solo un) oggetto `KpiResult`. L'oggetto `KpiResult` contiene il set di risultati complessivo per una query.

Ogni oggetto `KpiResult` contiene un oggetto `Rows`. Si tratta di un array di oggetti che contengono risultati di query e metadati pertinenti sui valori in tali risultati. La struttura e il contenuto di un oggetto `Rows` presentano le seguenti caratteristiche generali:

- Ogni riga di risultati di query è un oggetto JSON separato, denominato `Values`, nell'oggetto `Rows`. Ad esempio, se una query restituisce tre valori, l'oggetto `Rows` contiene tre oggetti `Values`. Ogni oggetto `Values` contiene un singolo risultato per la query.
- Ogni colonna dei risultati della query è una proprietà dell'oggetto `Values` a cui si applica. Il nome della colonna viene archiviato nel campo `Key` dell'oggetto `Values`.
- Per i risultati delle query raggruppate, ogni oggetto `Values` ha un oggetto `GroupedBys` associato. L'oggetto `GroupedBys` indica quale campo è stato utilizzato per raggruppare i risultati. Fornisce inoltre il valore di raggruppamento per l'oggetto `Values` associato.
- Se i risultati della query per un parametro sono null, l'oggetto `Rows` è vuoto.

Al di là di queste caratteristiche generali, la struttura e il contenuto dell'oggetto `Rows` variano a seconda del parametro. Questo perché Amazon Pinpoint supporta due tipi di metriche, ovvero metriche a valore singolo e metriche a valore multiplo.

Un parametro a valore singolo fornisce un solo valore cumulativo. Un esempio è la percentuale di messaggi recapitati ai destinatari da tutte le esecuzioni di una campagna. Un parametro a valore multiplo fornisce più di un valore e raggruppa tali valori in base a un campo pertinente. Un esempio è la percentuale di messaggi recapitati ai destinatari per ogni esecuzione di una campagna, raggruppati per esecuzione di campagna.

È possibile stabilire rapidamente se un parametro è un parametro a valore singolo o un parametro a valore multiplo facendo riferimento al nome del parametro. Se il nome non contiene `grouped-by`, si tratta di un parametro a valore singolo. Se lo fa, è un parametro a valore multiplo. Per un elenco completo dei parametri su cui è possibile eseguire query a livello di programmazione, consulta Metriche standard che si applicano a progetti, campagne e percorsi Amazon Pinpoint.

Parametri a valore singolo

Per un parametro a valore singolo, l'oggetto `Rows` contiene un oggetto `Values` che:

- Specifica il nome descrittivo del parametro su cui è stata eseguita la query.
- Fornisce il valore per il parametro su cui è stata eseguita la query.
- Identifica il tipo di dati del valore restituito.

Ad esempio, la seguente risposta JSON contiene i risultati della query per un parametro a valore singolo. Questo parametro riporta il numero di endpoint univoci a cui sono stati recapitati i messaggi da tutte le campagne associate a un progetto, dal 1° agosto 2019 al 31 agosto 2019:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"1368.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

In questo esempio, la risposta indica che tutte le campagne del progetto hanno recapitato messaggi a 1.368 endpoint univoci dal 1 agosto 2019 al 31 agosto 2019, dove:

- `Key` è il nome descrittivo del parametro il cui valore è specificato nel campo `Value` (`UniqueDeliveries`).
- `Type` è il tipo di dati del valore specificato nel campo `Value` (`Double`).
- `Value` è il valore effettivo per il parametro su cui è stata eseguita la query, inclusi eventuali filtri applicati (`1368.0`).

Se il risultato della query per una metrica a valore singolo è nullo (non maggiore o uguale a zero), l'oggetto `Rows` è vuoto. Amazon Pinpoint restituisce un valore nullo per una metrica se non ci sono dati da restituire per la metrica. Per esempio:

```
{
```

```
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"2345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[

            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Parametri a più valori

La struttura e il contenuto dell'oggetto `Rows` per un parametro a più valori sono per lo più uguali a un parametro a valore singolo. L'oggetto `Rows` per un parametro a valore multiplo contiene anche un oggetto `Values`. L'oggetto `Values` specifica il nome descrittivo del parametro con query, fornisce il valore per tale parametro e identifica il tipo di dati di tale valore.

Tuttavia, l'oggetto `Rows` per un parametro a più valori contiene anche uno o più oggetti `GroupedBy`. Esiste un oggetto `GroupedBy` per ogni oggetto `Values` nei risultati della query. L'oggetto `GroupedBy` indica quale campo è stato utilizzato per raggruppare i dati nei risultati e il tipo di dati di quel campo. Indica anche il valore di raggruppamento per quel campo (per l'oggetto `Values` associato).

Ad esempio, la seguente risposta JSON contiene i risultati della query per un parametro a più valori che segnala il numero di endpoint univoci a cui sono stati recapitati i messaggi per ogni campagna associata a un progetto, dal 1 agosto 2019 al 31 agosto 2019:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
```

```
                        "Value":"80b8efd84042ff8d9c96ce2f8example"
                    }
                ],
                "Values":[
                    {
                        "Key":"UniqueDeliveries",
                        "Type":"Double",
                        "Value":"123.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"CampaignId",
                        "Type":"String",
                        "Value":"810c7aab86d42fb2b56c8c966example"
                    }
                ],
                "Values":[
                    {
                        "Key":"UniqueDeliveries",
                        "Type":"Double",
                        "Value":"456.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"CampaignId",
                        "Type":"String",
                        "Value":"42d8c7eb0990a57ba1d5476a3example"
                    }
                ],
                "Values":[
                    {
                        "Key":"UniqueDeliveries",
                        "Type":"Double",
                        "Value":"789.0"
                    }
                ]
            }
        ]
```

```
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

In questo esempio, la risposta indica che tre delle campagne del progetto hanno inviato messaggi a endpoint univoci dal 1 agosto 2019 al 31 agosto 2019. Per ciascuna di queste campagne, la ripartizione dei conteggi dei recapiti è:

- La campagna `80b8efd84042ff8d9c96ce2f8example` ha recapitato messaggi a 123 endpoint univoci.
- La campagna `810c7aab86d42fb2b56c8c966example` ha recapitato messaggi a 456 endpoint univoci.
- La campagna `42d8c7eb0990a57ba1d5476a3example` ha recapitato messaggi a 789 endpoint univoci.

Dove la struttura generale degli oggetti e dei campi è:

- `GroupedBys.Key`: nome della proprietà o del campo che archivia il valore di raggruppamento specificato nel campo `GroupedBys.Value` (`CampaignId`).
- `GroupedBys.Type`: tipo di dati del valore specificato nel campo `GroupedBys.Value` (`String`).
- `GroupedBys.Value`: valore effettivo del campo utilizzato per raggruppare i dati, come specificato nel campo `GroupedBys.Key` (ID campagna).
- `Values.Key`: nome descrittivo della metrica il cui valore è specificato nel campo `Values.Value` (`UniqueDeliveries`).
- `Values.Type`: tipo di dati del valore specificato nel campo `Values.Value` (`Double`).
- `Values.Value`: valore effettivo della metrica su cui è stata eseguita la query, inclusi tutti i filtri applicati.

Se i risultati della query per una metrica a più valori sono null (non maggiori di o uguali a zero) per un progetto, una campagna o un'altra risorsa specifica, Amazon Pinpoint non restituisce alcun oggetto o campo per la risorsa. Se i risultati della query per una metrica a più valori sono null per tutte le risorse, Amazon Pinpoint restituisce un oggetto `Rows` vuoto.

Oggetti e campi JSON

Oltre a specificare i valori che hanno definito una query, ad esempio l'ID del progetto (`ApplicationId`), ogni risposta JSON a una query per un parametro dell'applicazione, il parametro della campagna o il parametro del viaggio include un oggetto `KpiResult`. Questo oggetto contiene il set di risultati complessivo per una query, che è possibile analizzare per inviare dati di analisi analitici a un altro servizio o applicazione. Ogni oggetto `KpiResult` contiene alcuni o tutti i seguenti oggetti e campi standard, a seconda del parametro.

Oggetto o campo	Descrizione
`Rows`	Un array di oggetti che contiene il set di risultati per una query.
`Rows.GroupedBys`	Per un parametro a più valori, un parametro di campi che definisce il campo e i valori utilizzati per raggruppare i dati nei risultati delle query.
`Rows.GroupedBys.Key`	Per un parametro a più valori, il nome della proprietà o del campo che memorizza il valore specificato nel campo `GroupedBys.Value` .
`Rows.GroupedBys.Type`	Per un parametro a più valori, il tipo di dati del valore specificato nel campo `GroupedBys.Value` .
`Rows.GroupedBys.Value`	Per un parametro a più valori, il valore effettivo per il campo utilizzato per raggruppare i dati nei risultati delle query. Questo valore è correlato a un oggetto `Values` associato.
`Rows.Values`	Un array di campi che contiene i risultati delle query.
`Rows.Values.Key`	Il nome descrittivo del parametro su cui è stata eseguita la query. Il valore del parametro è specificato nel campo `Values.Value` .

Oggetto o campo	Descrizione
`Rows.Values.Type`	Il tipo di dati del valore specificato nel campo `Values.Value` .
`Rows.Values.Value`	Il valore effettivo per il parametro su cui è stata eseguita la query, inclusi tutti i filtri applicati.

Per informazioni sui campi in una risposta JSON a una query relativa a un parametro di esecuzione del viaggio o a un parametro di esecuzione dell'attività di viaggio, consulta Metriche standard che si applicano a progetti, campagne e percorsi Amazon Pinpoint.

Registra le chiamate API Amazon Pinpoint con AWS CloudTrail

Amazon Pinpoint è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in Amazon Pinpoint. CloudTrail acquisisce le chiamate API per Amazon Pinpoint come eventi. Le chiamate acquisite includono le chiamate dalla console Amazon Pinpoint e le chiamate di codice alle operazioni delle API di Amazon Pinpoint.

Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon Simple Storage Service (Amazon S3), inclusi gli eventi per Amazon Pinpoint. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti utilizzando la cronologia degli eventi sulla console. CloudTrail Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata ad Amazon Pinpoint, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la Guida per l'AWS CloudTrail utente.

Informazioni su Amazon Pinpoint in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività di evento supportata in Amazon Pinpoint, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon Pinpoint, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail

- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di stabilire:

- Se la richiesta è stata effettuata con credenziali root o AWS Identity and Access Management utente.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento userIdentity di CloudTrail .

Puoi creare un trail e archiviare i file di log nel bucket Amazon S3 per il periodo di tempo desiderato. Puoi anche definire regole del ciclo di vita di Amazon S3 per archiviare o eliminare file di log automaticamente. Per impostazione predefinita, i file di log sono crittografati mediante la crittografia lato server (SSE) di Amazon S3.

Per ricevere una notifica della consegna dei file di log, configura CloudTrail la pubblicazione di notifiche Amazon SNS quando vengono consegnati nuovi file di log. Per ulteriori informazioni, consulta l'argomento relativo alla configurazione delle notifiche Amazon SNS per CloudTrail.

Puoi anche aggregare i file di log di Amazon Pinpoint da AWS più regioni e AWS più account in un unico bucket Amazon S3. Per ulteriori informazioni, consulta le pagine relative alla ricezione di file di log di CloudTrail da più regioni e alla ricezione di file di log di CloudTrail da più account.

Puoi utilizzarlo CloudTrail per registrare le azioni per i seguenti Amazon Pinpoint APIs:

- API Amazon Pinpoint
- API SMS e Voce di Amazon Pinpoint

Azioni dell'API Amazon Pinpoint supportate nei CloudTrail file di registro

L'API Amazon Pinpoint supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- CreateApp
- CreateCampaign
- CreateEmailTemplate
- CreateExportJob
- CreateImportJob
- CreateJourney
- CreatePushTemplate
- CreateRecommenderConfiguration
- CreateSegment
- CreateSmsTemplate
- CreateVoiceTemplate
- DeleteAdmChannel
- DeleteApnsChannel
- DeleteApnsSandboxChannel
- DeleteApnsVoipChannel
- DeleteApnsVoipSandboxChannel
- DeleteApp
- DeleteBaiduChannel
- DeleteCampaign
- DeleteEmailChannel
- DeleteEmailTemplate
- DeleteEndpoint
- DeleteEventStream
- DeleteGcmChannel
- DeleteJourney
- DeletePushTemplate
- DeleteRecommenderConfiguration
- DeleteSegment

- DeleteSmsChannel
- DeleteSmsTemplate
- DeleteUserEndpoints
- DeleteVoiceChannel
- DeleteVoiceTemplate
- GetAdmChannel
- GetApnsChannel
- GetApnsSandboxChannel
- GetApnsVoipChannel
- GetApnsVoipSandboxChannel
- GetApp
- GetApplicationDateRangeKpi
- GetApplicationSettings
- GetApps
- GetBaiduChannel
- GetCampaign
- GetCampaignActivities
- GetCampaignDateRangeKpi
- GetCampaignVersion
- GetCampaignVersions
- GetCampaigns
- GetChannels
- GetEmailChannel
- GetEmailTemplate
- GetEndpoint
- GetEventStream
- GetExportJob
- GetExportJobs

- GetGcmChannel
- GetImportJob
- GetImportJobs
- GetJourney
- GetJourneyDateRangeKpi
- GetJourneyExecutionActivityMetrics
- GetJourneyExecutionMetrics
- GetPushTemplate
- GetRecommenderConfiguration
- GetRecommenderConfigurations
- GetSegment
- GetSegmentExportJobs
- GetSegmentImportJobs
- GetSegmentVersion
- GetSegmentVersions
- GetSegments
- GetSmsChannel
- GetSmsTemplate
- GetUserEndpoints
- GetVoiceChannel
- GetVoiceTemplate
- ListJourneys
- ListTagsForResource
- ListTemplates
- ListTemplateVersions
- PhoneNumberValidate
- PutEvents
- PutEventStream

- RemoveAttributes
- SendMessages
- Invia OTPMessage
- SendUsersMessages
- TagResource
- UntagResource
- UpdateAdmChannel
- UpdateApnsChannel
- UpdateApnsSandboxChannel
- UpdateApnsVoipChannel
- UpdateApnsVoipSandboxChannel
- UpdateApplicationSettings
- UpdateBaiduChannel
- UpdateCampaign
- UpdateEmailChannel
- UpdateEmailTemplate
- UpdateEndpoint
- UpdateEndpointsBatch
- UpdateGcmChannel
- UpdateJourney
- UpdateJourneyState
- UpdatePushTemplate
- UpdateRecommenderConfiguration
- UpdateSegment
- UpdateSmsChannel
- UpdateSmsTemplate
- UpdateTemplateActiveVersion
- UpdateVoiceChannel
- UpdateVoiceTemplate

Azioni dell'API e-mail Amazon Pinpoint supportate nei CloudTrail file di registro

L'API Amazon Pinpoint Email supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- CreateDedicatedIpPool
- CreateEmailIdentity
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- DeleteDedicatedIpPool
- DeleteEmailIdentity
- GetAccount
- GetConfigurationSet
- GetConfigurationSetEventDestinations
- GetDedicatedIp
- GetDedicatedIps
- GetEmailIdentity
- ListConfigurationSets
- ListDedicatedIpPools
- ListEmailIdentities
- PutAccountDedicatedIpWarmupAttributes
- PutAccountSendingAttributes
- PutConfigurationSetDeliveryOptions
- PutConfigurationSetReputationOptions
- PutConfigurationSetSendingOptions
- PutConfigurationSetTrackingOptions
- PutDedicatedIpInPool

- PutDedicatedIpWarmupAttributes
- PutEmailIdentityDkimAttributes
- PutEmailIdentityFeedbackAttributes
- PutEmailIdentityMailFromAttributes
- UpdateConfigurationSetEventDestination

La seguente azione dell'API Amazon Pinpoint Email non è stata registrata: CloudTrail

- SendEmail

Azioni supportate per SMS e API vocali di Amazon Pinpoint versione 1 nei CloudTrail file di registro

L'API Amazon Pinpoint SMS and Voice versione 1 supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- GetConfigurationSetEventDestinations
- UpdateConfigurationSetEventDestination

La seguente azione API Amazon Pinpoint SMS and Voice versione 1 non è stata registrata: CloudTrail

- SendVoiceMessage

CloudTrail esempi di inserimento di log che mostrano le azioni dell'API Amazon Pinpoint

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro.

Un evento rappresenta una singola richiesta da un'origine. Include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra `CreateCampaign` le azioni `GetCampaigns` e le azioni dell'API Amazon Pinpoint.


```
{
  "Records": [
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "GetCampaigns",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T00:56:48Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": true,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "application-id": "example71dfa4c1aab66332a5839798f",
        "page-size": "1000"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Jersey/${project.version} (HttpUrlConnection 1.8.0_144)",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "principalId": "123456789012",
        "sessionContext": {
          "attributes": {
            "creationDate": "2018-02-02T16:55:29Z",
            "mfaAuthenticated": "false"
          }
        },
        "type": "Root"
      }
    },
    {
```

```
    "awsRegion": "us-east-1",
    "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
    "eventName": "CreateCampaign",
    "eventSource": "pinpoint.amazonaws.com",
    "eventTime": "2018-02-03T01:05:16Z",
    "eventType": "AwsApiCall",
    "eventVersion": "1.05",
    "readOnly": false,
    "recipientAccountId": "123456789012",
    "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
    "requestParameters": {
      "Description": "***",
      "HoldoutPercent": 0,
      "IsPaused": false,
      "MessageConfiguration": "***",
      "Name": "***",
      "Schedule": {
        "Frequency": "ONCE",
        "IsLocalTime": true,
        "StartTime": "2018-02-03T00:00:00-08:00",
        "Timezone": "utc-08"
      },
      "SegmentId": "exampleda204adf991a80281aa0e591",
      "SegmentVersion": 1,
      "application-id": "example71dfa4c1aab66332a5839798f"
    },
    "responseElements": {
      "ApplicationId": "example71dfa4c1aab66332a5839798f",
      "CreationDate": "2018-02-03T01:05:16.425Z",
      "Description": "***",
      "HoldoutPercent": 0,
      "Id": "example54a654f80948680cbba240ede",
      "IsPaused": false,
      "LastModifiedDate": "2018-02-03T01:05:16.425Z",
      "MessageConfiguration": "***",
      "Name": "***",
      "Schedule": {
        "Frequency": "ONCE",
        "IsLocalTime": true,
        "StartTime": "2018-02-03T00:00:00-08:00",
        "Timezone": "utc-08"
      },
      "SegmentId": "example4da204adf991a80281example",
      "SegmentVersion": 1,
```

```
        "State": {
          "CampaignStatus": "SCHEDULED"
        },
        "Version": 1
      },
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.9 Python/3.4.3 Linux/3.4.0+ botocore/1.8.34",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/userName",
        "principalId": "AIDAIHTHRCDA62EXAMPLE",
        "type": "IAMUser",
        "userName": "userName"
      }
    }
  ]
}
```

L'esempio seguente mostra una voce di CloudTrail registro che illustra le `CreateConfigurationSetEventDestination` azioni `CreateConfigurationSet` e nell'API SMS and Voice di Amazon Pinpoint.

```
{
  "Records": [
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:45:55Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSet",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "ConfigurationSetName":"MyConfigurationSet"
```

```
      },
      "responseElements":null,
      "requestID":"56dcc091-e20d-11e8-87d2-9994aexample",
      "eventID":"725843fc-8846-41f4-871a-7c52dexample",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "recipientAccountId":"123456789012"
    },
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:47:08Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSetEventDestination",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "EventDestinationName":"CloudWatchEventDestination",
        "ConfigurationSetName":"MyConfigurationSet",
        "EventDestination":{
          "Enabled":true,
          "MatchingEventTypes":[
            "INITIATED_CALL",
            "INITIATED_CALL"
          ],
          "CloudWatchLogsDestination":{
            "IamRoleArn":"arn:aws:iam::111122223333:role/iamrole-01",
            "LogGroupArn":"arn:aws:logs:us-east-1:111122223333:log-
group:clientloggroup-01"
          }
        }
      },
      "responseElements":null,
      "requestID":"81de1e73-e20d-11e8-b158-d5536example",
      "eventID":"fcafc21f-7c93-4a3f-9e72-fca2dexample",
      "readOnly":false,
```

```
      "eventType":"AwsApiCall",
      "recipientAccountId":"111122223333"
    }
  ]
}
```

Usa un modello di raccomandazione in Amazon Pinpoint con AWS Lambda

In Amazon Pinpoint, puoi recuperare le raccomandazioni personalizzate da un modello di raccomandazione e aggiungerle ai messaggi inviati da campagne e percorsi. Un modello della funzione di suggerimento è un tipo di modello di machine learning (ML) che individua pattern nei dati e genera previsioni e suggerimenti in base ai pattern individuati. Predice ciò che un particolare utente preferirà da un determinato set di prodotti o articoli, e fornisce tali informazioni come una serie di suggerimenti per l'utente.

Utilizzando i modelli di raccomandazione con Amazon Pinpoint, puoi inviare raccomandazioni personalizzate ai destinatari dei messaggi in base agli attributi e al comportamento di ciascun destinatario. Con AWS Lambda, puoi anche personalizzare e migliorare questi consigli. Ad esempio, puoi trasformare dinamicamente un suggerimento da un singolo valore di testo (ad esempio un nome prodotto o un ID) a contenuti più sofisticati (ad esempio un nome prodotto, una descrizione e un'immagine). Puoi inoltre farlo in tempo reale, quando Amazon Pinpoint invia il messaggio.

Questa funzionalità è disponibile nelle seguenti AWS regioni: Stati Uniti orientali (Virginia settentrionale); Stati Uniti occidentali (Oregon); Asia Pacifico (Mumbai); Asia Pacifico (Sydney) ed Europa (Irlanda).

Aggiungi consigli sui modelli di raccomandazione ai messaggi in Amazon Pinpoint

Per utilizzare un modello di raccomandazione con Amazon Pinpoint, crea una soluzione Amazon Personalize e quindi implementala come campagna Amazon Personalize. Quindi, puoi creare una configurazione per il modello di raccomandazione in Amazon Pinpoint. Nella configurazione è necessario specificare le impostazioni che determinano come recuperare ed elaborare i dati relativi alle raccomandazioni dalla campagna Amazon Personalize. Ciò include se richiamare una AWS Lambda funzione per eseguire un'ulteriore elaborazione dei dati recuperati.

Amazon Personalize è un AWS servizio progettato per aiutarti a creare modelli di machine learning che forniscono consigli personalizzati in tempo reale per i clienti che utilizzano le tue applicazioni. Grazie a procedure semplificate, Amazon Personalize supporta l'utente nel processo di creazione e addestramento di un modello di machine learning, quindi nella preparazione e implementazione del modello come campagna Amazon Personalize. Puoi quindi recuperare i suggerimenti personalizzati

in tempo reale dalla campagna. Per ulteriori informazioni su Amazon Personalize, consulta la Guida per gli sviluppatori di Amazon Personalize.

AWS Lambda è un servizio di elaborazione che puoi utilizzare per eseguire codice senza fornire o gestire server. Impacchettate il codice e lo caricate AWS Lambda come funzione Lambda. AWS Lambda quindi esegue la funzione quando la funzione viene richiamata. Puoi richiamare una funzione manualmente, come risposta automatica agli eventi o in risposta a richieste provenienti da applicazioni o servizi, compreso Amazon Pinpoint. Per ulteriori informazioni su come creare e richiamare le funzioni Lambda, consulta Guida per gli sviluppatori di AWS Lambda.

Dopo aver creato una configurazione Amazon Pinpoint per un modello di raccomandazione, puoi aggiungere raccomandazioni dal modello ai messaggi inviati da campagne e percorsi. Puoi eseguire questa operazione utilizzando i modelli di messaggio che contengono variabili di messaggio per gli attributi suggeriti. Un attributo suggerito è un endpoint dinamico o un attributo utente progettato per memorizzare i dati dei suggerimenti. Puoi definire questi attributi crei la configurazione per un modello della funzione di suggerimento.

Puoi utilizzare le variabili per gli attributi suggeriti nei seguenti tipi di modelli di messaggio:

- Modelli di posta elettronica, per i messaggi di posta elettronica inviati da campagne o viaggi.
- Modelli di notifica push, per le notifiche push inviate dalle campagne.
- Modelli SMS, per messaggi di testo SMS inviati da campagne.

Per ulteriori informazioni sull'utilizzo dei modelli di raccomandazione con Amazon Pinpoint, consulta Modelli di machine learning nella Guida per l'utente di Amazon Pinpoint.

Se configuri Amazon Pinpoint per richiamare una funzione Lambda che elabora i dati delle raccomandazioni, Amazon Pinpoint esegue le seguenti attività generali ogni volta che invia raccomandazioni personalizzate in un messaggio per una campagna o un percorso:

1. Valuta ed elabora le impostazioni di configurazione e il contenuto del messaggio e del modello di messaggio.
2. Determina che il modello di messaggio è connesso a un modello della funzione di suggerimento.
3. Valuta le impostazioni di configurazione per la connessione e l'utilizzo del modello. Queste impostazioni sono definite dalla risorsa del Modello della funzione di suggerimento per il modello.
4. Rileva una o più variabili di messaggio per gli attributi suggeriti definiti dalle impostazioni di configurazione per il modello.

5. Recupera i dati delle raccomandazioni dalla campagna Amazon Personalize specificata nelle impostazioni di configurazione per il modello. Utilizza il GetRecommendationsfunzionamento dell'API Amazon Personalize Runtime per eseguire questa attività.
6. Aggiunge i dati suggeriti appropriati a un attributo dinamico suggerito (`RecommendationItems`) per ogni destinatario del messaggio.
7. Richiama la funzione Lambda e invia i dati delle raccomandazioni per ciascun destinatario a tale funzione per l'elaborazione.

   I dati vengono inviati come oggetto JSON che contiene la definizione dell'endpoint per ogni destinatario. Ogni definizione di endpoint include un campo `RecommendationItems` contenente un array ordinato contenente da 1 a 5 valori. Il numero di valori nella matrice dipende dalle impostazioni di configurazione del modello.
8. Attende che la funzione Lambda elabori i dati e restituisca i risultati.

   I risultati consistono in un oggetto JSON che contiene una definizione dell'endpoint aggiornata per ogni destinatario. Ogni definizione dell'endpoint aggiornata contiene un nuovo oggetto `Recommendations`. Questo oggetto contiene da 1 a 10 campi, uno per ogni attributo consigliato personalizzato definito nelle impostazioni di configurazione per il modello. Ciascuno di questi campi memorizza dati dei suggerimenti avanzati per l'endpoint.
9. Utilizza la definizione dell'endpoint aggiornata per ciascun destinatario per sostituire ogni variabile di messaggio con il valore appropriato per tale destinatario.
10. Invia una versione del messaggio contenente i suggerimenti personalizzati per ciascun destinatario del messaggio.

Per personalizzare e migliorare le raccomandazioni in questo modo, crea una funzione Lambda che elabori le definizioni di endpoint inviate da Amazon Pinpoint e restituisca le definizioni aggiornate. Quindi, assegna una policy di funzione Lambda alla funzione e autorizza Amazon Pinpoint a richiamare la funzione. Quindi, configura il modello di raccomandazione in Amazon Pinpoint. Quando configuri il modello, specifica la funzione da richiamare e definisci gli attributi suggeriti da utilizzare.

Crea una funzione Lambda per Amazon Pinpoint da richiamare per un modello di raccomandazione

Per informazioni su come creare una funzione Lambda, consulta Guida introduttiva nella Guida per gli sviluppatori di AWS Lambda . Quando progetti e sviluppi la tua funzione, tiene a mente i requisiti e le linee guida seguenti.

Dati dell'evento di input

Quando Amazon Pinpoint richiama una funzione Lambda per un modello di raccomandazione, invia un payload contenente la configurazione e altre impostazioni per la campagna o il percorso di invio del messaggio. Il payload include un `Endpoints` oggetto, che è una mappa che associa l'endpoint alle definizioni degli endpoint IDs per i destinatari dei messaggi.

Le definizioni di endpoint utilizzano la struttura definita dalla risorsa Endpoint dell'API Amazon Pinpoint. Tuttavia, includono anche un campo per un attributo dinamico suggerito denominato `RecommendationItems`. Il campo `RecommendationItems` contiene uno o più elementi suggeriti per l'endpoint, come restituiti dalla campagna Amazon Personalize. Il valore di questo campo è un array ordinato contenente da 1 a 5 elementi suggeriti (come stringhe). Il numero di elementi nell'array dipende dal numero di elementi suggeriti configurati per il recupero da parte di Amazon Pinpoint per ciascun endpoint o utente.

Per esempio:

```
"Endpoints": {
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
```

```
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z"
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1210",
```

```
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z"
    }
}
```

Nell'esempio precedente, le impostazioni Amazon Pinpoint pertinenti sono:

- Il modello della funzione di suggerimento è configurato per recuperare tre elementi suggeriti per ogni endpoint o utente. (Il valore della proprietà `RecommendationsPerMessage` è impostato su `3`.) Con questa impostazione, Amazon Pinpoint recupera e aggiunge solo i primi, secondi e terzi elementi suggeriti per ogni endpoint o utente.
- Il progetto è configurato per utilizzare attributi utente personalizzati che memorizzano il nome, il cognome e il quartiere in cui vive ogni utente. (L'oggetto `UserAttributes` contiene i valori per questi attributi.)
- Il progetto è configurato per utilizzare un attributo endpoint personalizzato (`AddressType`) che indica se l'endpoint è l'indirizzo (canale) preferito dall'utente per ricevere messaggi dal progetto. (L'oggetto `Attributes` contiene il valore per questo attributo.)

Quando Amazon Pinpoint richiama la funzione Lambda e invia il payload come dati dell'evento, AWS Lambda passa i dati alla funzione Lambda per l'elaborazione.

Ogni payload può contenere dati per un massimo di 50 endpoint. Se un segmento contiene più di 50 endpoint, Amazon Pinpoint richiama la funzione ripetutamente, per un massimo di 50 endpoint alla volta, finché non vengono elaborati tutti i dati.

Dati e requisiti di risposta

Quando progetti e sviluppi la funzione Lambda, considera le quote per i modelli di machine learning. Se la funzione non soddisfa le condizioni definite da queste quote, Amazon Pinpoint non sarà in grado di elaborare e inviare il messaggio.

Tieni a mente anche i seguenti requisiti:

- La funzione deve restituire le definizioni degli endpoint aggiornate nello stesso formato fornito dai dati degli eventi di input.

- Ogni definizione di endpoint aggiornata può contenere da 1 a 10 attributi consigliati personalizzati per l'endpoint o l'utente. I nomi di questi attributi devono corrispondere ai nomi degli attributi specificati quando configuri il modello di raccomandazione in Amazon Pinpoint.
- Tutti gli attributi suggeriti personalizzati devono essere restituiti in un singolo oggetto `Recommendations` per ogni endpoint o utente. Questo requisito consente di garantire che i conflitti di denominazione non si verifichino. Puoi aggiungere l'oggetto `Recommendations` in qualsiasi posizione in una definizione di endpoint.
- Il valore per ogni attributo suggerito personalizzato deve essere una stringa (valore singolo) o una matrice di stringhe (valori multipli). Se il valore è un array di stringhe, è consigliabile mantenere l'ordine degli elementi suggeriti restituiti da Amazon Personalize, come indicato nel campo `RecommendationItems`. In caso contrario, il contenuto potrebbe non riflettere le previsioni del modello per un endpoint o un utente.
- La funzione non deve modificare altri elementi nei dati dell'evento, inclusi altri valori di attributo per un endpoint o un utente. Deve aggiungere e restituire solo valori per gli attributi consigliati personalizzati. Amazon Pinpoint non accetterà aggiornamenti a nessun altro valore nella risposta della funzione.
- La funzione deve essere ospitata nella stessa AWS regione del progetto Amazon Pinpoint che richiama la funzione. Se la funzione e il progetto non si trovano nella stessa regione, Amazon Pinpoint non è in grado di inviare i dati degli eventi alla funzione.

Se uno dei requisiti precedenti non viene soddisfatto, Amazon Pinpoint non sarà in grado di elaborare e inviare il messaggio a uno o più endpoint. Ciò potrebbe causare un fallimento di una campagna o di un'attività di viaggio.

Infine, si consiglia di riservare 256 esecuzioni simultanee per la funzione.

Nel complesso, la funzione Lambda deve elaborare i dati degli eventi inviati da Amazon Pinpoint e restituire le definizioni di endpoint modificate. È possibile eseguire questa operazione mediante l'iterazione di ciascun endpoint nell'oggetto `Endpoints` e, per ogni endpoint, creando e impostando valori per gli attributi suggeriti personalizzati che si desidera utilizzare. Il seguente handler di esempio, scritto in Python e che prosegue l'esempio precedente di dati degli eventi di input, mostra quanto segue:

```
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
```

Nell'esempio precedente, AWS Lambda passa i dati dell'evento al gestore come parametro. `event` L'handler esegue l'iterazione di ciascun endpoint nell'oggetto `Endpoints` e imposta i valori per gli attributi suggeriti personalizzati denominati `Recommendations.Title` e `Recommendations.Genre`. L'istruzione `return` restituisce ogni definizione di endpoint aggiornata ad Amazon Pinpoint.

Continuando con l'esempio precedente di dati degli eventi di input, le definizioni degli endpoint aggiornati sono:

```
"Endpoints":{
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z",
```

```
        "Recommendations":{
            "Title":[
                "Fleabag",
                "Late Night",
                "Catastrophe"
            ],
            "Genre":[
                "Comedy",
                "Comedy",
                "Comedy"
            ]
        }
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z",
```

```
        "Recommendations":{
            "Title":[
                "Hanna",
                "Hunters",
                "Agatha Christie\'s The ABC Murders"
            ],
            "Genre":[
                "Action",
                "Drama",
                "Crime"
            ]
        }
    }
}
```

Nell'esempio precedente, la funzione ha modificato l'oggetto `Endpoints` ricevuto e restituito i risultati. L'oggetto `Endpoint` per ogni endpoint ora contiene un nuovo oggetto `Recommendations`, che contiene i campi `Title` e `Genre`. Ciascuno di questi campi memorizza una matrice ordinata di tre valori (come stringhe), in cui ogni valore fornisce contenuto migliorato per un elemento suggerito corrispondente nel campo `RecommendationItems`.

Assegna una policy di funzione Lambda per autorizzare Amazon Pinpoint a elaborare i dati delle raccomandazioni

Prima di poter utilizzare la funzione Lambda per elaborare i dati delle raccomandazioni, devi autorizzare Amazon Pinpoint a richiamare la funzione. Per concedere l'autorizzazione per l'invocazione, assegna una policy di funzione Lambda alla funzione. Una policy di funzione Lambda è una policy di autorizzazione basata su risorse che designa le entità che possono utilizzare la funzione e le azioni che tali entità possono eseguire. Per ulteriori informazioni, consulta Uso delle policy basate sulle risorse per AWS Lambda nella Guida per gli sviluppatori di AWS Lambda .

La seguente policy di esempio consente al responsabile del servizio Amazon Pinpoint di utilizzare l'`lambda:InvokeFunction`azione per una particolare campagna Amazon Pinpoint *`campaignId`* () in un particolare progetto Amazon Pinpoint (): *`projectId`*

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
```

```
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

La policy di funzione richiede un blocco `Condition` che includa una chiave `AWS:SourceArn`. Questa chiave specifica quale risorsa può invocare la funzione. Nell'esempio precedente, la policy consente a una particolare campagna di richiamare la funzione.

Puoi anche scrivere una policy che consenta al responsabile del servizio Amazon Pinpoint di utilizzare l'`lambda:InvokeFunction`azione per tutte le campagne e i percorsi di uno specifico progetto Amazon Pinpoint (). *`projectId`* Di seguito viene illustrato un esempio di policy che mostra questo approccio:

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

A differenza del primo esempio, la chiave `AWS:SourceArn` nel blocco `Condition` di questo esempio consente a un particolare progetto di invocare la funzione. Questa autorizzazione si applica a tutte le campagne e viaggi nel progetto.

Per scrivere una policy più generica, puoi utilizzare un carattere jolly di corrispondenza con più caratteri (*). Ad esempio, puoi utilizzare il seguente blocco `Condition` per consentire a qualsiasi progetto Amazon Pinpoint di richiamare la funzione:

```
"Condition": {
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}
```

Se desideri utilizzare la funzione Lambda con tutti i progetti per l'account Amazon Pinpoint, è consigliabile configurare il blocco `Condition` della policy come indicato in precedenza. Tuttavia, come procedura consigliata, dovresti creare policy che includano solo le autorizzazioni necessarie per eseguire un'operazione specifica su una risorsa specifica.

Autorizza Amazon Pinpoint a richiamare una funzione Lambda utilizzando il comando e AWS CLI il comando Lambda add-permission

Dopo aver assegnato una policy di funzione Lambda a una funzione, puoi aggiungere autorizzazioni che consentono ad Amazon Pinpoint di richiamare la funzione per un progetto, una campagna o un percorso specifico. Puoi farlo usando il comando AWS Command Line Interface (AWS CLI) e Lambda `add-permission`. L'esempio seguente mostra come eseguire questa operazione per un progetto specifico (*`projectId`*):

```
$ aws lambda add-permission \
--function-name function-name \
--statement-id sid \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*
```

L'esempio precedente è formattato per Unix, Linux e macOS. Per Microsoft Windows, sostituire il carattere di continuazione della riga (\) con un accento circonflesso (^).

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "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/*\"}}}"
}
```

Il valore di `Statement` è una versione in formato stringa JSON dell'istruzione aggiunta alla policy della funzione Lambda.

Configura Amazon Pinpoint per richiamare la funzione Lambda per un modello di raccomandazione

Per configurare Amazon Pinpoint per richiamare la funzione Lambda per un modello di raccomandazione, specifica le seguenti impostazioni di configurazione specifiche di Lambda per il modello:

- `RecommendationTransformerUri`: questa proprietà specifica il nome o il nome della risorsa Amazon (ARN) della funzione Lambda.
- `Attributes`: questo oggetto è una mappa che definisce gli attributi consigliati personalizzati che la funzione aggiunge a ciascuna definizione di endpoint. Ciascuno di questi attributi può essere utilizzato come variabile di messaggio in un modello di messaggio.

Puoi specificare queste impostazioni utilizzando la risorsa Modelli di raccomandazione dell'API Amazon Pinpoint (quando crei la configurazione per un modello) o la risorsa Modello di raccomandazione dell'API Amazon Pinpoint (se aggiorni la configurazione per un modello). Puoi inoltre definire queste impostazioni utilizzando la console Amazon Pinpoint.

Per ulteriori informazioni sull'utilizzo dei modelli di raccomandazione con Amazon Pinpoint, consulta Modelli di machine learning nella Guida per l'utente di Amazon Pinpoint.

Elimina il tuo progetto Amazon Pinpoint e rimuovi i dati personali sensibili

A seconda di come viene utilizzato, Amazon Pinpoint può archiviare determinati dati che potrebbero essere considerati personali. Ad esempio, un endpoint in Amazon Pinpoint contiene le informazioni di contatto di un utente finale, come l'indirizzo e-mail o il numero di cellulare.

Puoi utilizzare la console o l'API Amazon Pinpoint per eliminare definitivamente i dati personali. Questo argomento illustra le procedure per l'eliminazione di vari tipi di dati che possono essere considerati dati personali.

Puoi anche chiudere completamente il tuo AWS account. Per ulteriori informazioni, consulta Chiudere un AWS account nella Guida Gestione dell'account AWS di riferimento.

Elimina tutti i dati del progetto Amazon Pinpoint

È possibile eliminare definitivamente tutti i dati archiviati per un progetto Amazon Pinpoint. È possibile farlo eliminando il progetto.

> **Warning**
>
> Se elimini un progetto, Amazon Pinpoint elimina tutte le impostazioni specifiche e i dati del progetto. Le informazioni non possono essere recuperate.

Quando elimini un progetto, Amazon Pinpoint elimina tutte le impostazioni specifiche del progetto per la notifica push e i canali di messaggistica SMS bidirezionale, nonché tutti i segmenti, le campagne, i percorsi e i dati analitici specifici del progetto archiviati in Amazon Pinpoint, ad esempio:

- Segmenti: tutte le impostazioni e i dati dei segmenti. Per segmenti dinamici, questo include i gruppi di segmenti e i filtri definiti. Per i segmenti importati, sono inclusi gli endpoint, gli utenti IDs e gli altri dati che hai importato e tutti i filtri applicati.
- Campagne: tutti i messaggi, i trattamenti e le variabili dei messaggi, i dati analitici, le pianificazioni e altre impostazioni.
- Percorsi: tutte le attività, i dati analitici, gli orari e altre impostazioni.

- Analisi: dati analitici per tutte le metriche di coinvolgimento, ad esempio il numero di messaggi inviati e recapitati per campagne e percorsi e tutte le metriche di esecuzione dei percorsi. Per le app mobili e Web, tutti i dati sugli eventi che non sono stati trasmessi in streaming a un altro AWS servizio come Amazon Kinesis, tutte le canalizzazioni e i dati relativi all'utilizzo delle applicazioni, alle entrate e ai parametri demografici. Prima di eliminare un progetto, ti consigliamo di esportare questi dati in un'altra posizione.

Puoi eliminare un progetto utilizzando la console Amazon Pinpoint. Per ulteriori informazioni, consulta Eliminazione di un progetto nella Guida per l'utente di Amazon Pinpoint. Puoi anche eliminare un progetto a livello di programmazione utilizzando la risorsa App dell'API Amazon Pinpoint.

Esempi di codice per l'utilizzo di Amazon Pinpoint AWS SDKs

I seguenti esempi di codice mostrano come usare Amazon Pinpoint con un kit di sviluppo AWS software (SDK).

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Esempi di codice per l'utilizzo di Amazon Pinpoint AWS SDKs
  - Esempi di base per l'utilizzo di Amazon Pinpoint AWS SDKs
    - Azioni per l'utilizzo di Amazon Pinpoint AWS SDKs
      - Utilizzo CreateApp con un AWS SDK o una CLI
      - Utilizzare CreateCampaign con un SDK AWS
      - Utilizzare CreateExportJob con un SDK AWS
      - Utilizzare CreateImportJob con un SDK AWS
      - Utilizzare CreateSegment con un SDK AWS
      - Utilizzo DeleteApp con un AWS SDK o una CLI
      - Utilizzare DeleteEndpoint con un SDK AWS
      - Utilizzo GetEndpoint con un AWS SDK o una CLI
      - Utilizzare GetSegments con un SDK AWS
      - Utilizzo GetSmsChannel con un AWS SDK o una CLI
      - Utilizzare GetUserEndpoints con un SDK AWS
      - Utilizzo SendMessages con un AWS SDK o una CLI
      - Utilizzare UpdateEndpoint con un SDK AWS
- Esempi di codice per l'API SMS e Voice di Amazon Pinpoint utilizzando AWS SDKs
  - Esempi di base per l'utilizzo dell'API Amazon Pinpoint SMS and Voice AWS SDKs
    - Azioni per l'API SMS e Voice di Amazon Pinpoint utilizzando AWS SDKs
      - Utilizzalo SendVoiceMessage con un AWS SDK

Esempi di codice per l'utilizzo di Amazon Pinpoint AWS SDKs

I seguenti esempi di codice mostrano come usare Amazon Pinpoint con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Esempi di base per l'utilizzo di Amazon Pinpoint AWS SDKs
  - Azioni per l'utilizzo di Amazon Pinpoint AWS SDKs
    - Utilizzo CreateApp con un AWS SDK o una CLI
    - Utilizzare CreateCampaign con un SDK AWS
    - Utilizzare CreateExportJob con un SDK AWS
    - Utilizzare CreateImportJob con un SDK AWS
    - Utilizzare CreateSegment con un SDK AWS
    - Utilizzo DeleteApp con un AWS SDK o una CLI
    - Utilizzare DeleteEndpoint con un SDK AWS
    - Utilizzo GetEndpoint con un AWS SDK o una CLI
    - Utilizzare GetSegments con un SDK AWS
    - Utilizzo GetSmsChannel con un AWS SDK o una CLI
    - Utilizzare GetUserEndpoints con un SDK AWS
    - Utilizzo SendMessages con un AWS SDK o una CLI
    - Utilizzare UpdateEndpoint con un SDK AWS

Esempi di base per l'utilizzo di Amazon Pinpoint AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Amazon AWS SDKs Pinpoint con.

Esempi

- Azioni per l'utilizzo di Amazon Pinpoint AWS SDKs
  - Utilizzo CreateApp con un AWS SDK o una CLI
  - Utilizzare CreateCampaign con un SDK AWS
  - Utilizzare CreateExportJob con un SDK AWS
  - Utilizzare CreateImportJob con un SDK AWS
  - Utilizzare CreateSegment con un SDK AWS
  - Utilizzo DeleteApp con un AWS SDK o una CLI
  - Utilizzare DeleteEndpoint con un SDK AWS
  - Utilizzo GetEndpoint con un AWS SDK o una CLI
  - Utilizzare GetSegments con un SDK AWS
  - Utilizzo GetSmsChannel con un AWS SDK o una CLI
  - Utilizzare GetUserEndpoints con un SDK AWS
  - Utilizzo SendMessages con un AWS SDK o una CLI
  - Utilizzare UpdateEndpoint con un SDK AWS

Azioni per l'utilizzo di Amazon Pinpoint AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Pinpoint con. AWS SDKs Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la documentazione di riferimento dell'API Amazon Pinpoint.

Esempi

- Utilizzo CreateApp con un AWS SDK o una CLI
- Utilizzare CreateCampaign con un SDK AWS
- Utilizzare CreateExportJob con un SDK AWS
- Utilizzare CreateImportJob con un SDK AWS
- Utilizzare CreateSegment con un SDK AWS
- Utilizzo DeleteApp con un AWS SDK o una CLI
- Utilizzare DeleteEndpoint con un SDK AWS

- Utilizzo GetEndpoint con un AWS SDK o una CLI
- Utilizzare GetSegments con un SDK AWS
- Utilizzo GetSmsChannel con un AWS SDK o una CLI
- Utilizzare GetUserEndpoints con un SDK AWS
- Utilizzo SendMessages con un AWS SDK o una CLI
- Utilizzare UpdateEndpoint con un SDK AWS

Utilizzo **`CreateApp`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `CreateApp`.

CLI

AWS CLI

Esempio 1: creazione di un'applicazione

L'esempio `create-app` seguente crea una nuova applicazione (progetto).

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

Esempio 2: creazione di un'applicazione con tag

L'esempio `create-app` seguente crea una nuova applicazione (progetto) e associa un tag (chiave e valore) all'applicazione.

```
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"
        }
    }
}
```

- Per i dettagli sull'API, consulta CreateApp AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateAppRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateAppResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateApplicationRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateApp {
    public static void main(String[] args) {
        final String usage = """

                 Usage:  <appName>

                 Where:
                  appName - The name of the application to create.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
        String appName = args[0];
        System.out.println("Creating an application with name: " + appName);

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

        String appID = createApplication(pinpoint, appName);
        System.out.println("App ID is: " + appID);
        pinpoint.close();
    }

    public static String createApplication(PinpointClient pinpoint, String
 appName) {
        try {
            CreateApplicationRequest appRequest =
 CreateApplicationRequest.builder()
                    .name(appName)
                    .build();

            CreateAppRequest request = CreateAppRequest.builder()
                    .createApplicationRequest(appRequest)
                    .build();

            CreateAppResponse result = pinpoint.createApp(request);
            return result.applicationResponse().id();
```

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

- Per i dettagli sull'API, consulta la CreateAppsezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createApplication(applicationName: String?): String? {
    val createApplicationRequestOb =
        CreateApplicationRequest {
            name = applicationName
        }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.createApp(
                CreateAppRequest {
                    createApplicationRequest = createApplicationRequestOb
                },
            )
        return result.applicationResponse?.id
    }
}
```

- Per i dettagli sull'API, CreateAppconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateCampaign`** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare `CreateCampaign`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea una campagna.

```
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;
    }
}
```

- Per i dettagli sull'API, consulta la CreateCampaignsezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createPinCampaign(
    appId: String,
    segmentIdVal: String,
) {
    val scheduleOb =
        Schedule {
            startTime = "IMMEDIATE"
        }

    val defaultMessageOb =
        Message {
            action = Action.OpenApp
            body = "My message body"
            title = "My message title"
        }

    val messageConfigurationOb =
        MessageConfiguration {
            defaultMessage = defaultMessageOb
        }

    val writeCampaign =
        WriteCampaignRequest {
            description = "My description"
            schedule = scheduleOb
            name = "MyCampaign"
            segmentId = segmentIdVal
            messageConfiguration = messageConfigurationOb
        }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result: CreateCampaignResponse =
```

```
            pinpoint.createCampaign(
                CreateCampaignRequest {
                    applicationId = appId
                    writeCampaignRequest = writeCampaign
                },
            )
        println("Campaign ID is ${result.campaignResponse?.id}")
    }
}
```

- Per i dettagli sull'API, CreateCampaignconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateExportJob`** con un SDK AWS

Il seguente esempio di codice mostra come utilizzare`CreateExportJob`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esporta un endpoint.

```
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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la CreateExportJobsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Pinpoint con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateImportJob`** con un SDK AWS

Il seguente esempio di codice mostra come utilizzare`CreateImportJob`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Importa un segmento.

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

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

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

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

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

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

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

        PinpointClient pinpoint = PinpointClient.builder()
```

```
                .region(Region.US_EAST_1)
                .build();

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

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

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

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

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

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

- Per i dettagli sull'API, consulta la CreateImportJobsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Pinpoint con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateSegment`** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare `CreateSegment`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

/**
```

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

                                Usage:   <appId>

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

                                """;

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

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

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

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

```
                                        .values("Lakers")
                                        .build());

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

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

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

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

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

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

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

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

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

- Per i dettagli sull'API, consulta la CreateSegmentsezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createPinpointSegment(applicationIdVal: String?): String? {
    val segmentAttributes = mutableMapOf<String, AttributeDimension>()
    val myList = mutableListOf<String>()
    myList.add("Lakers")

    val atts =
        AttributeDimension {
            attributeType = AttributeType.Inclusive
            values = myList
        }

    segmentAttributes["Team"] = atts
```

```
    val recencyDimension =
        RecencyDimension {
            duration = Duration.fromValue("DAY_30")
            recencyType = RecencyType.fromValue("ACTIVE")
        }

    val segmentBehaviors =
        SegmentBehaviors {
            recency = recencyDimension
        }

    val segmentLocation = SegmentLocation {}
    val dimensionsOb =
        SegmentDimensions {
            attributes = segmentAttributes
            behavior = segmentBehaviors
            demographic = SegmentDemographics {}
            location = segmentLocation
        }

    val writeSegmentRequestOb =
        WriteSegmentRequest {
            name = "MySegment101"
            dimensions = dimensionsOb
        }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val createSegmentResult: CreateSegmentResponse =
            pinpoint.createSegment(
                CreateSegmentRequest {
                    applicationId = applicationIdVal
                    writeSegmentRequest = writeSegmentRequestOb
                },
            )
        println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
        return createSegmentResult.segmentResponse?.id
    }
}
```

- Per i dettagli sull'API, CreateSegmentconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteApp`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DeleteApp`.

CLI

AWS CLI

Eliminazione di un'applicazione

L'esempio `delete-app` seguente elimina un'applicazione (progetto).

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

Output:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- Per i dettagli sull'API, consulta DeleteApp AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina un'applicazione.

```
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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DeleteAppsezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deletePinApp(appId: String?) {
    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.deleteApp(
                DeleteAppRequest {
                    applicationId = appId
                },
            )
        val appName = result.applicationResponse?.name
        println("Application $appName has been deleted.")
    }
}
```

- Per i dettagli sull'API, DeleteAppconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DeleteEndpoint`** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare `DeleteEndpoint`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Eliminare un endpoint.

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

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

```
                Usage:   <appName> <endpointId >

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

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

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

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

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

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

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

- Per i dettagli sull'API, consulta la DeleteEndpointsezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deletePinEncpoint(
    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")
    }
}
```

- Per i dettagli sull'API, DeleteEndpointconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetEndpoint`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetEndpoint`.

CLI

AWS CLI

Recupero di informazioni sulle impostazioni e gli attributi di un endpoint specifico per un'applicazione

L'esempio `get-endpoint` seguente recupera le informazioni sulle impostazioni e gli attributi di un endpoint specifico per un'applicazione.

```
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"
        }
    }
}
```

- Per i dettagli sull'API, consulta GetEndpoint AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

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

```
                Usage:   <appId> <endpoint>

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

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

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

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

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

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

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

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

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

- Per i dettagli sull'API, consulta la GetEndpointsezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun lookupPinpointEndpoint(
    appId: String?,
    endpoint: String?,
) {
    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.getEndpoint(
                GetEndpointRequest {
                    applicationId = appId
                    endpointId = endpoint
                },
            )
        val endResponse = result.endpointResponse

        // Uses the Google Gson library to pretty print the endpoint JSON.
        val gson: com.google.gson.Gson =
            GsonBuilder()
                .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                .setPrettyPrinting()
```

```
                .create()

        val endpointJson: String = gson.toJson(endResponse)
        println(endpointJson)
    }
}
```

- Per i dettagli sull'API, GetEndpointconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`GetSegments`** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare `GetSegments`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca i segmenti.

```
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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la GetSegmentssezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listSegs(appId: String?) {
    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val response =
            pinpoint.getSegments(
                GetSegmentsRequest {
                    applicationId = appId
                },
            )
        response.segmentsResponse?.item?.forEach { segment ->
            println("Segement id is ${segment.id}")
        }
    }
}
```

- Per i dettagli sull'API, GetSegmentsconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetSmsChannel`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetSmsChannel`.

CLI

AWS CLI

Recupero delle informazioni sullo stato e sulle impostazioni del canale SMS per un'applicazione

L'esempio `get-sms-channel` seguente recupera lo stato e le impostazioni del canale di sms per un'applicazione.

```
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
    }
}
```

- Per i dettagli sull'API, consulta GetSmsChannel AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelResponse;
import software.amazon.awssdk.services.pinpoint.model.GetSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelResponse;

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

                Usage: CreateChannel <appId>

                Where:
                  appId - The name of the application whose channel is updated.

                """;

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

```
        }

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

        SMSChannelResponse getResponse = getSMSChannel(pinpoint, appId);
        toggleSmsChannel(pinpoint, appId, getResponse);
        pinpoint.close();
    }

    private static SMSChannelResponse getSMSChannel(PinpointClient client, String
appId) {
        try {
            GetSmsChannelRequest request = GetSmsChannelRequest.builder()
                    .applicationId(appId)
                    .build();

            SMSChannelResponse response =
client.getSmsChannel(request).smsChannelResponse();
            System.out.println("Channel state is " + response.enabled());
            return response;

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

    private static void toggleSmsChannel(PinpointClient client, String appId,
SMSChannelResponse getResponse) {
        boolean enabled = !getResponse.enabled();
        try {
            SMSChannelRequest request = SMSChannelRequest.builder()
                    .enabled(enabled)
                    .build();

            UpdateSmsChannelRequest updateRequest =
UpdateSmsChannelRequest.builder()
                    .smsChannelRequest(request)
                    .applicationId(appId)
                    .build();
```

```
            UpdateSmsChannelResponse result =
 client.updateSmsChannel(updateRequest);
            System.out.println("Channel state: " +
 result.smsChannelResponse().enabled());

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

- Per i dettagli sull'API, consulta la GetSmsChannelsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Pinpoint con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`GetUserEndpoints`** con un SDK AWS

Il seguente esempio di codice mostra come utilizzare`GetUserEndpoints`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

```
import java.util.List;

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

                Usage:    <applicationId> <userId>

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

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

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

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

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

        try {
            GetUserEndpointsRequest endpointsRequest =
 GetUserEndpointsRequest.builder()
```

```
                    .userId(userId)
                    .applicationId(applicationId)
                    .build();

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

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

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

- Per i dettagli sull'API, consulta la GetUserEndpointssezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Pinpoint con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`SendMessages`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `SendMessages`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Invia un messaggio e-mail.

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

namespace SendEmailMessage;

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

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

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

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
```

```
        string toAddress = configuration["ToAddress"]!;

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

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

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

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

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

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

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

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

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

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

Invia un messaggio 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;
    }
}
```

- Per i dettagli sull'API, consulta la SendMessagessezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Invio di un messaggio SMS utilizzando l'endpoint di un'applicazione

L'esempio `send-messages` seguente invia un messaggio diretto per un'applicazione con un endpoint.

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

Contenuto di `myfile.json`.

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Endpoints": {
```

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

Output:

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

Per ulteriori informazioni, consulta Canale SMS di Amazon Pinpoint nella Guida per l'utente di Amazon Pinpoint.

- Per i dettagli sull'API, consulta SendMessages AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Invia un messaggio e-mail.

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

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

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

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

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

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

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

        """;
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Invio di un messaggio e-mail con valori 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();
        }
    }
}
```

Invia un messaggio SMS.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Invia messaggi SMS in 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.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessageBatch {

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

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

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

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

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

                Usage:   <message> <appId> <originationNumber>
 <destinationNumber> <destinationNumber1>\s

                Where:
                  message - The body of the message to send.
                  appId - The Amazon Pinpoint project/application ID to use when
 you send this message.
                  originationNumber - The phone number or short code that
 you specify has to be associated with your Amazon Pinpoint account. For best
 results, specify long codes in E.164 format (for example, +1-555-555-5654).
                  destinationNumber - The recipient's phone number.  For best
 results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).
                  destinationNumber1 - The second recipient's phone number.  For
 best results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).\s
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String message = args[0];
        String appId = args[1];
        String originationNumber = args[2];
        String destinationNumber = args[3];
        String destinationNumber1 = args[4];
```

```
        System.out.println("Sending a message");
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber, destinationNumber1);
        pinpoint.close();
    }

    public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                                      String originationNumber,
                                      String destinationNumber, String
destinationNumber1) {
        try {
            Map<String, AddressConfiguration> addressMap = new HashMap<String,
AddressConfiguration>();
            AddressConfiguration addConfig = AddressConfiguration.builder()
                    .channelType(ChannelType.SMS)
                    .build();

            // Add an entry to the Map object for each number to whom you want to
send a
            // message.
            addressMap.put(destinationNumber, addConfig);
            addressMap.put(destinationNumber1, addConfig);
            SMSMessage smsMessage = SMSMessage.builder()
                    .body(message)
                    .messageType(messageType)
                    .originationNumber(originationNumber)
                    .senderId(senderId)
                    .keyword(registeredKeyword)
                    .build();

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

            MessageRequest msgReq = MessageRequest.builder()
                    .addresses(addressMap)
                    .messageConfiguration(direct)
```

```
                .build();

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

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

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

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

- Per i dettagli sull'API, consulta la SendMessagessezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Creare il client in un modulo separato ed esportarlo.

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

Invia un messaggio e-mail.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

// The subject line of the email.
const subject = "Amazon Pinpoint Test (AWS SDK for JavaScript in Node.js)";

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

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

// The character encoding for the subject line and message body of the email.
const charset = "UTF-8";

const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
```

```
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

const run = async () => {
  try {
    const { MessageResponse } = await pinClient.send(
      new SendMessagesCommand(params),
    );

    if (!MessageResponse) {
      throw new Error("No message response.");
    }

    if (!MessageResponse.Result) {
      throw new Error("No message result.");
    }

    const recipientResult = MessageResponse.Result[toAddress];

    if (recipientResult.StatusCode !== 200) {
      throw new Error(recipientResult.StatusMessage);
```

```
    }
    console.log(recipientResult.MessageId);
  } catch (err) {
    console.log(err.message);
  }
};

run();
```

Invia un messaggio SMS.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

/* The phone number or short code to send the message from. The phone number
 or short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format. */
const originationNumber = "SENDER_NUMBER"; //e.g., +1XXXXXXXXXX

// The recipient's phone number.  For best results, you should specify the phone
 number in E.164 format.
const destinationNumber = "RECEIVER_NUMBER"; //e.g., +1XXXXXXXXXX

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

/*The Amazon Pinpoint project/application ID to use when you send this message.
Make sure that the SMS channel is enabled for the project or application
that you choose.*/
const projectId = "PINPOINT_PROJECT_ID"; //e.g., XXXXXXXX66e4e9986478cXXXXXXXXX

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

// The registered keyword associated with the originating short code.
```

```
const registeredKeyword = "myKeyword";

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

const senderId = "MySenderID";

// Specify the parameters to pass to the API.
const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

const run = async () => {
  try {
    const data = await pinClient.send(new SendMessagesCommand(params));
    console.log(
      `Message sent!
 ${data.MessageResponse.Result[destinationNumber].StatusMessage}`,
    );
  } catch (err) {
    console.log(err);
  }
};
run();
```

- Per i dettagli sull'API, consulta la SendMessagessezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Invia un messaggio e-mail.

```
"use strict";

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

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

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

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

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

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
----------------------------------------------------
```

```
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

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

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

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

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

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

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

```
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

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

Invia un messaggio SMS.

```
"use strict";

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

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
```

```
var aws_region = "us-east-1";

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

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

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

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

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

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

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

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

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

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

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

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

- Per i dettagli sull'API, consulta la SendMessagessezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */

val body: String =
    """
    Amazon Pinpoint test (AWS SDK for Kotlin)

    This email was sent through the Amazon Pinpoint Email API using the AWS SDK
 for Kotlin.

    """.trimIndent()

suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <subject> <appId> <senderAddress> <toAddress>

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

    if (args.size != 3) {
```

```
        println(usage)
        exitProcess(0)
    }

    val subject = args[0]
    val senderAddress = args[1]
    val toAddress = args[2]
    sendEmail(subject, senderAddress, toAddress)
}

suspend fun sendEmail(
    subjectVal: String?,
    senderAddress: String,
    toAddressVal: String,
) {
    var content =
        Content {
            data = body
        }

    val messageBody =
        Body {
            text = content
        }

    val subContent =
        Content {
            data = subjectVal
        }

    val message =
        Message {
            body = messageBody
            subject = subContent
        }

    val destinationOb =
        Destination {
            toAddresses = listOf(toAddressVal)
        }

    val emailContent =
        EmailContent {
            simple = message
```

```
        }

    val sendEmailRequest =
        SendEmailRequest {
            fromEmailAddress = senderAddress
            destination = destinationOb
            this.content = emailContent
        }

    PinpointEmailClient { region = "us-east-1" }.use { pinpointemail ->
        pinpointemail.sendEmail(sendEmailRequest)
        println("Message Sent")
    }
}
```

- Per i dettagli sull'API, SendMessagesconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Invia un messaggio e-mail.

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

logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
```

```
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

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

```
                            "TextPart": {"Charset": char_set, "Data":
 text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


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

    print("Sending email.")
    message_ids = send_email_message(
```

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


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

Invia un messaggio 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()
```

Invia un messaggio e-mail con un modello di e-mail esistente.

```
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()
```

Invia un messaggio di testo con un modello SMS esistente.

```
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()
```

- Per i dettagli sull'API, consulta SendMessages AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`UpdateEndpoint`** con un SDK AWS

Il seguente esempio di codice mostra come utilizzare`UpdateEndpoint`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.EndpointDemographic;
import software.amazon.awssdk.services.pinpoint.model.EndpointLocation;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
```

```
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.UUID;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Date;

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

                Usage:  <appId>

                Where:
                  appId - The ID of the application to create an endpoint for.

                """;

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

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

        EndpointResponse response = createEndpoint(pinpoint, appId);
        System.out.println("Got Endpoint: " + response.id());
        pinpoint.close();
    }
```

```
   public static EndpointResponse createEndpoint(PinpointClient client, String
appId) {
       String endpointId = UUID.randomUUID().toString();
       System.out.println("Endpoint ID: " + endpointId);

       try {
           EndpointRequest endpointRequest = createEndpointRequestData();
           UpdateEndpointRequest updateEndpointRequest =
UpdateEndpointRequest.builder()
                   .applicationId(appId)
                   .endpointId(endpointId)
                   .endpointRequest(endpointRequest)
                   .build();

           UpdateEndpointResponse updateEndpointResponse =
client.updateEndpoint(updateEndpointRequest);
           System.out.println("Update Endpoint Response: " +
updateEndpointResponse.messageBody());

           GetEndpointRequest getEndpointRequest = GetEndpointRequest.builder()
                   .applicationId(appId)
                   .endpointId(endpointId)
                   .build();

           GetEndpointResponse getEndpointResponse =
client.getEndpoint(getEndpointRequest);
           System.out.println(getEndpointResponse.endpointResponse().address());

System.out.println(getEndpointResponse.endpointResponse().channelType());

System.out.println(getEndpointResponse.endpointResponse().applicationId());

System.out.println(getEndpointResponse.endpointResponse().endpointStatus());

System.out.println(getEndpointResponse.endpointResponse().requestId());
           System.out.println(getEndpointResponse.endpointResponse().user());

           return getEndpointResponse.endpointResponse();

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

```
    }

    private static EndpointRequest createEndpointRequestData() {
        try {
            List<String> favoriteTeams = new ArrayList<>();
            favoriteTeams.add("Lakers");
            favoriteTeams.add("Warriors");
            HashMap<String, List<String>> customAttributes = new HashMap<>();
            customAttributes.put("team", favoriteTeams);

            EndpointDemographic demographic = EndpointDemographic.builder()
                    .appVersion("1.0")
                    .make("apple")
                    .model("iPhone")
                    .modelVersion("7")
                    .platform("ios")
                    .platformVersion("10.1.1")
                    .timezone("America/Los_Angeles")
                    .build();

            EndpointLocation location = EndpointLocation.builder()
                    .city("Los Angeles")
                    .country("US")
                    .latitude(34.0)
                    .longitude(-118.2)
                    .postalCode("90068")
                    .region("CA")
                    .build();

            Map<String, Double> metrics = new HashMap<>();
            metrics.put("health", 100.00);
            metrics.put("luck", 75.00);

            EndpointUser user = EndpointUser.builder()
                    .userId(UUID.randomUUID().toString())
                    .build();

            DateFormat df = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm'Z'"); //
Quoted "Z" to indicate UTC, no timezone
                                                                           //
offset
            String nowAsISO = df.format(new Date());

            return EndpointRequest.builder()
```

```
                    .address(UUID.randomUUID().toString())
                    .attributes(customAttributes)
                    .channelType("APNS")
                    .demographic(demographic)
                    .effectiveDate(nowAsISO)
                    .location(location)
                    .metrics(metrics)
                    .optOut("NONE")
                    .requestId(UUID.randomUUID().toString())
                    .user(user)
                    .build();

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

- Per i dettagli sull'API, consulta la UpdateEndpointsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Pinpoint con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice per l'API SMS e Voice di Amazon Pinpoint utilizzando AWS SDKs

I seguenti esempi di codice mostrano come utilizzare l'API Amazon Pinpoint SMS and Voice con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Esempi di base per l'utilizzo dell'API Amazon Pinpoint SMS and Voice AWS SDKs
  - Azioni per l'API SMS e Voice di Amazon Pinpoint utilizzando AWS SDKs
    - Utilizzalo SendVoiceMessage con un AWS SDK

Esempi di base per l'utilizzo dell'API Amazon Pinpoint SMS and Voice AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le basi dell'API SMS e Voice di Amazon Pinpoint con. AWS SDKs

Esempi

- Azioni per l'API SMS e Voice di Amazon Pinpoint utilizzando AWS SDKs
  - Utilizzalo SendVoiceMessage con un AWS SDK

Azioni per l'API SMS e Voice di Amazon Pinpoint utilizzando AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole azioni API Amazon Pinpoint SMS e Voice con. AWS SDKs Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la documentazione di riferimento dell'API SMS e Voce di Amazon Pinpoint.

Esempi

- Utilizzalo SendVoiceMessage con un AWS SDK

Utilizzalo **`SendVoiceMessage`** con un AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare `SendVoiceMessage`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

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

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

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

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

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

   public static void main(String[] args) {

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

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

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

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

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

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

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

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

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

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

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

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

```
}
```

- Per i dettagli sull'API, SendVoiceMessageconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
"use strict";

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

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

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best
 results, you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

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

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
```

```
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint
 account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

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

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

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

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
```

```
};

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

- Per i dettagli sull'API, SendVoiceMessageconsulta AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
```

```
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your
Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your
Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
message.
    :param ssml_message: The content of the message. This example uses SSML to
control
                         certain aspects of the message, such as the volume and
the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
```

```
        )
        raise
    else:
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


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

- Per i dettagli sull'API, consulta SendVoiceMessage AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Pinpoint con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Sicurezza in Amazon Pinpoint

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili ad Amazon Pinpoint, consulta AWS Services in Scope by Compliance Program Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

La presente documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si utilizza Amazon Pinpoint. Gli argomenti seguenti descrivono come configurare Amazon Pinpoint per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon Pinpoint.

Per ulteriori informazioni sulle architetture di riferimento, consulta il manuale relativo all'architettura resiliente di Amazon Pinpoint.

Argomenti

- Protezione dei dati in Amazon Pinpoint
- Gestione di identità e accessi per Amazon Pinpoint
- Registrazione di log e monitoraggio in Amazon Pinpoint
- Convalida della conformità per Amazon Pinpoint
- Resilienza in Amazon Pinpoint
- Sicurezza dell'infrastruttura in Amazon Pinpoint
- Analisi della configurazione e delle vulnerabilità in Amazon Pinpoint

- Best practice di sicurezza per Amazon Pinpoint

Protezione dei dati in Amazon Pinpoint

Il modello di responsabilità AWS condivisa si applica alla protezione dei dati in Amazon Pinpoint. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con Amazon Pinpoint o altro Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un

server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

A seconda di come configuri e utilizzi il servizio, Amazon Pinpoint può archiviare i seguenti tipi di dati personali tuoi o dei tuoi clienti.

Dati di configurazione

Sono inclusi i dati di configurazione del progetto, ad esempio le credenziali e le impostazioni che definiscono come e quando Amazon Pinpoint invia i messaggi attraverso i canali supportati e i segmenti utente a cui vengono inviati i messaggi. Per inviare messaggi, questi dati possono includere indirizzi IP dedicati per i messaggi e-mail, codici brevi e mittente IDs per i messaggi di testo SMS e credenziali per la comunicazione con servizi di notifica push come il servizio Apple Push Notification (APNs) e Firebase Cloud Messaging (FCM).

Dati relativi a utenti ed endpoint

Sono inclusi gli attributi standard e personalizzati utilizzati per archiviare e gestire i dati relativi agli utenti e agli endpoint per un progetto Amazon Pinpoint. Un attributo può memorizzare informazioni su un utente specifico (ad esempio il nome di un utente) o su un endpoint specifico per un utente (ad esempio l'indirizzo e-mail di un utente, il numero di cellulare o il token del dispositivo mobile). Questi dati possono includere anche utenti esterni IDs che mettono in correlazione gli utenti di un progetto Amazon Pinpoint con gli utenti di un sistema esterno, come un sistema di gestione delle relazioni con i clienti. Per ulteriori informazioni su ciò che questi dati possono includere, consulta gli schemi Utente ed Endpoint nella documentazione di riferimento dell'API Amazon Pinpoint.

Dati analitici

Ciò include i dati per le metriche, note anche come indicatori chiave di prestazione (KPIs), che forniscono informazioni sulle prestazioni di un progetto Amazon Pinpoint per aree come il coinvolgimento degli utenti e l'attività di acquisto. Sono inclusi anche i dati per i parametri che forniscono informazioni dettagliate sui dati demografici degli utenti per un progetto. I dati possono derivare da attributi standard e personalizzati per utenti ed endpoint, ad esempio la città in cui risiede un utente. Possono derivare anche da eventi, ad esempio eventi di apertura e clic per i messaggi e-mail inviati per un progetto.

Dati importati

Sono inclusi i dati di qualsiasi utente, segmentazione e analisi aggiunti o importati da origini esterne e utilizzati in Amazon Pinpoint. Un esempio è un file JSON importato in Amazon Pinpoint (direttamente tramite la console o da un bucket Amazon S3) per creare un segmento statico. Altri

esempi sono i dati degli endpoint aggiunti a livello di programmazione per creare un segmento dinamico, gli indirizzi degli endpoint a cui si inviano i messaggi diretti e gli eventi per cui configuri un'app per segnalarli ad Amazon Pinpoint.

Argomenti

- Crittografia dei dati
- Riservatezza del traffico Internet
- Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint

Crittografia dei dati

I dati Amazon Pinpoint in transito e quelli inattivi sono crittografati. I dati inviati ad Amazon Pinpoint vengono crittografati mentre vengono ricevuti e archiviati. I dati recuperati da Amazon Pinpoint vengono trasmessi mediante i protocolli di sicurezza correnti.

Crittografia a riposo

Amazon Pinpoint crittografa tutti i dati archiviati. Sono inclusi i dati di configurazione, i dati dell'utente e degli endpoint, i dati analitici e tutti i dati aggiunti o importati in Amazon Pinpoint. Per crittografare i tuoi dati, Amazon Pinpoint utilizza chiavi AWS Key Management Service interne AWS KMS() che il servizio possiede e gestisce per tuo conto. Queste chiavi vengono ruotate su base regolare. Per informazioni in merito AWS KMS, consulta la AWS Key Management Service Developer Guide.

Crittografia in transito

Amazon Pinpoint utilizza HTTPS e Transport Layer Security (TLS) 1.2 o versioni successive per comunicare con i client e le applicazioni. Per comunicare con altri AWS servizi, Amazon Pinpoint utilizza HTTPS e TLS 1.2. Inoltre, quando crei e gestisci risorse Amazon Pinpoint utilizzando la console, un AWS SDK o il AWS Command Line Interface, tutte le comunicazioni sono protette tramite HTTPS e TLS 1.2.

Gestione delle chiavi

Per crittografare i dati di Amazon Pinpoint, Amazon Pinpoint utilizza chiavi AWS KMS interne che il servizio possiede e gestisce per tuo conto. Queste chiavi vengono ruotate su base regolare. Non puoi fornire e utilizzare chiavi tue AWS KMS o di altre chiavi per crittografare i dati archiviati in Amazon Pinpoint.

Riservatezza del traffico Internet

La privacy del traffico interrete si riferisce alla protezione delle connessioni e del traffico tra Amazon Pinpoint e i client e le applicazioni locali e tra Amazon Pinpoint e AWS altre risorse nella stessa regione. AWS Le seguenti funzionalità e procedure ti consentono di garantire la riservatezza del traffico Internet per Amazon Pinpoint.

Traffico tra Amazon Pinpoint e applicazioni e client on-premise

Per stabilire una connessione privata tra Amazon Pinpoint e i client e le applicazioni nella rete on-premise, puoi utilizzare AWS Direct Connect. Consente di collegare la rete a una posizione AWS Direct Connect utilizzando un cavo Ethernet standard in fibra ottica. Un'estremità del cavo è collegata al router. L'altra estremità è collegata a un router. AWS Direct Connect Per ulteriori informazioni, consulta Che cos'è AWS Direct Connect? nella Guida per l'utente di AWS Direct Connect .

Per garantire l'accesso sicuro ad Amazon Pinpoint tramite la pubblicazione APIs, ti consigliamo di rispettare i requisiti di Amazon Pinpoint per le chiamate API. Amazon Pinpoint richiede ai client l'utilizzo di Transport Layer Security (TLS) 1.2 o versione successiva. I client devono inoltre supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). La maggior parte dei sistemi moderni come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale AWS Identity and Access Management (IAM) per il tuo AWS account. In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per firmare le richieste.

Traffico tra Amazon Pinpoint e altre risorse AWS

Per proteggere le comunicazioni tra Amazon Pinpoint e altre AWS risorse nella stessa AWS regione, Amazon Pinpoint utilizza HTTPS e TLS 1.2 per impostazione predefinita.

Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint

È possibile stabilire una connessione privata tra il cloud privato virtuale (VPC) e un endpoint in Amazon Pinpoint creando un endpoint VPC di interfaccia.

Gli endpoint di interfaccia sono alimentati da AWS PrivateLink, una tecnologia che consente di accedere in modo privato ad Amazon APIs Pinpoint senza un gateway Internet, un dispositivo NAT,

una connessione VPN o. AWS Direct Connect Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per comunicare con Amazon APIs Pinpoint con cui si integra. AWS PrivateLink

Per ulteriori informazioni, consulta la Guida per AWS PrivateLink.

Creazione di un endpoint VPC di interfaccia

Puoi creare un endpoint di interfaccia utilizzando la console Amazon VPC o AWS Command Line Interface ().AWS CLI Per ulteriori informazioni, consulta Creare un endpoint di interfaccia nella Guida. AWS PrivateLink

Amazon Pinpoint supporta i seguenti nomi di servizio:

- `com.amazonaws.region.pinpoint`
- `com.amazonaws.region.pinpoint-sms-voice-v2`

Se attivi il DNS privato per un endpoint di interfaccia, puoi effettuare richieste API ad Amazon Pinpoint utilizzando il nome DNS predefinito per Regione AWS, ad esempio,. `com.amazonaws.us-east-1.pinpoint` Per ulteriori informazioni, consulta Hostname DNS nella Guida per l'utente di AWS PrivateLink .

Per l'elenco di tutte le regioni e di tutti gli endpoint nei quali Amazon Pinpoint è disponibile, consulta la sezione relativa agli endpoint del servizio AWS nella Riferimenti generali di Amazon Web Services.

Creazione di una policy degli endpoint VPC

È possibile collegare una policy degli endpoint all'endpoint VPC che controlla l'accesso. La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire azioni.

Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai servizi con policy di endpoint nella Guida di AWS PrivateLink .

Esempio di policy di endpoint VPC

La seguente policy degli endpoint VPC concede l'accesso alle azioni Amazon Pinpoint elencate per tutti i principali su tutte le risorse.

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

Gestione di identità e accessi per Amazon Pinpoint

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (chi può effettuare l'accesso) e autorizzato (chi dispone delle autorizzazioni) per utilizzare le risorse Amazon Pinpoint. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso con policy
- Come Amazon Pinpoint funziona con IAM
- Azioni di Amazon Pinpoint per le policy IAM
- Esempi di policy basate su identità Amazon Pinpoint
- Ruoli IAM per attività comuni di Amazon Pinpoint
- Risoluzione dei problemi relativi alla gestione di identità e accessi di Amazon Pinpoint

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon Pinpoint.

Utente del servizio: se utilizzi il servizio Amazon Pinpoint per eseguire il processo, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità Amazon Pinpoint utilizzate per il lavoro, potrebbero essere necessarie ulteriori autorizzazioni. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in Amazon Pinpoint, consulta Risoluzione dei problemi relativi alla gestione di identità e accessi di Amazon Pinpoint .

Amministratore del servizio: se sei il responsabile delle risorse Amazon Pinpoint presso la tua azienda, probabilmente disponi dell'accesso completo ad Amazon Pinpoint. Sarà tua responsabilità determinare le funzionalità e le risorse di Amazon Pinpoint a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon Pinpoint, consulta Come Amazon Pinpoint funziona con IAM.

Amministratore IAM: se sei un amministratore IAM, potresti essere interessato a ottenere informazioni su come scrivere policy per gestire l'accesso ad Amazon Pinpoint. Per visualizzare esempi di policy basate su identità di Amazon Pinpoint che possono essere utilizzate in IAM, consulta Esempi di policy basate su identità Amazon Pinpoint.

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se

non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo consigliato per la firma delle richieste, consulta Signature Version 4 AWS per le richieste API nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA)AWS in IAM nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali

temporanee. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi passare da un ruolo utente a un ruolo IAM (console). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Create a role for a third-party identity provider (federation) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.

- Sessioni di accesso inoltrato (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.
- Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.
- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. Questa soluzione è preferibile alla memorizzazione delle chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella IAM User Guide.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Amazon Pinpoint supporta l'utilizzo di policy basate su identità per controllare l'accesso alle risorse Amazon Pinpoint.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Amazon Pinpoint supporta l'utilizzo di policy basate su risorse per controllare l'accesso alle risorse Amazon Pinpoint.

Elenchi di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Amazon Pinpoint non supporta l'uso di ACLs per controllare l'accesso alle risorse Amazon Pinpoint.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo del servizio (SCPs): SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più di proprietà dell' Account AWS azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS Per ulteriori informazioni su Organizations and SCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.

- Politiche di controllo delle risorse (RCPs): RCPs sono politiche JSON che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le politiche IAM allegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations e RCPs, incluso un elenco di Servizi AWS tale supporto RCPs, vedere Resource control policies (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Amazon Pinpoint supporta l'utilizzo di questi tipi di policy per controllare l'accesso alle risorse Amazon Pinpoint.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta la logica di valutazione delle policy nella IAM User Guide.

Come Amazon Pinpoint funziona con IAM

Per utilizzare Amazon Pinpoint, gli utenti del tuo AWS account richiedono autorizzazioni che consentano loro di visualizzare dati di analisi, creare progetti, definire segmenti di utenti, distribuire campagne e altro ancora. Se integri un'app per dispositivi mobili o Web con Amazon Pinpoint, anche gli utenti dell'app richiedono l'accesso ad Amazon Pinpoint. Questo accesso consente all'app di registrare gli endpoint e di segnalare i dati di utilizzo ad Amazon Pinpoint. Per concedere l'accesso alle funzionalità di Amazon Pinpoint, crea policy AWS Identity and Access Management (IAM) che consentano azioni Amazon Pinpoint per identità IAM o risorse Amazon Pinpoint.

IAM è un servizio che aiuta gli amministratori a controllare in modo sicuro l'accesso alle risorse. AWS Le policy IAM includono istruzioni che consentono o negano azioni specifiche per risorse o utenti specifici. In Amazon Pinpoint è disponibile una serie di azioni che è possibile utilizzare nelle policy

IAM per specificare autorizzazioni granulari per utenti e risorse Amazon Pinpoint. Questo significa che puoi concedere il livello appropriato di accesso ad Amazon Pinpoint senza creare policy troppo permissive che potrebbero esporre dati importanti o compromettere le tue risorse. Ad esempio, puoi concedere l'accesso illimitato a un amministratore Amazon Pinpoint e in sola lettura a coloro che devono accedere solo a uno specifico progetto.

Prima di utilizzare IAM per gestire l'accesso ad Amazon Pinpoint, è necessario comprendere quali funzionalità IAM sono disponibili per l'uso con Amazon Pinpoint. Per avere una visione di alto livello di come Amazon Pinpoint e AWS altri servizi funzionano con IAM, AWS consulta i servizi che funzionano con IAM nella IAM User Guide.

Argomenti

- Policy basate su identità Amazon Pinpoint
- Policy di autorizzazione basate su risorse di Amazon Pinpoint
- Autorizzazione basata sui tag Amazon Pinpoint
- Ruoli IAM Amazon Pinpoint

Policy basate su identità Amazon Pinpoint

Con le policy basate su identità di IAM, è possibile specificare quali azioni e risorse sono consentite o rifiutate, nonché le condizioni in base alle quali le azioni sono consentite o rifiutate. Amazon Pinpoint supporta specifiche azioni, risorse e chiavi di condizione. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Azioni

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Ciò significa che le azioni delle policy controllano ciò che gli utenti possono eseguire sulla console Amazon Pinpoint. Controllano anche ciò che gli utenti possono fare a livello di codice utilizzando AWS SDKs direttamente, the AWS Command Line Interface (AWS CLI) o Amazon APIs Pinpoint.

Le azioni di policy in Amazon Pinpoint utilizzano i seguenti prefissi:

- **`mobiletargeting`**: per le azioni che derivano dall'API Amazon Pinpoint, che è l'API primaria per Amazon Pinpoint.
- **`sms-voice`**: per le azioni che derivano dalla API SMS e Voce di Amazon Pinpoint, che è un'API supplementare che fornisce opzioni avanzate per l'utilizzo e la gestione dei canali SMS e voce in Amazon Pinpoint.

Ad esempio, per concedere l'autorizzazione per visualizzare le informazioni su tutti i segmenti di un progetto, ovvero un'operazione che corrisponde all'operazione `GetSegments` nell'API Amazon Pinpoint, includi l'operazione `mobiletargeting:GetSegments` nella policy. Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. Amazon Pinpoint definisce un proprio set di azioni che descrivono le attività che puoi eseguire.

Per specificare più operazioni in una sola istruzione, separarle con la virgola:

```
"Action": [
      "mobiletargeting:action1",
      "mobiletargeting:action2"
```

Puoi anche specificare più operazioni utilizzando i caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Get`, includi la seguente azione:

```
"Action": "mobiletargeting:Get*"
```

Tuttavia, è consigliabile definire policy in grado di seguire il principio del privilegio minimo. In altre parole, è necessario creare policy che includano solo le autorizzazioni necessarie per eseguire un'operazione specifica.

Per un elenco delle azioni Amazon Pinpoint che puoi utilizzare nelle policy IAM, consulta Azioni di Amazon Pinpoint per le policy IAM.

Risorse

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Ad esempio, l'azione `mobiletargeting:GetSegments` recupera le informazioni su tutti i segmenti associati a un progetto Amazon Pinpoint specifico. Identifica un progetto con un ARN nel seguente formato:

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

Per ulteriori informazioni sul formato di ARNs, consulta Amazon Resource Names (ARNs) nel Riferimenti generali di AWS.

Nelle policy IAM, puoi specificare ARNs i seguenti tipi di risorse Amazon Pinpoint:

- Campagne
- Percorsi
- Modelli di messaggio (denominati modelli in alcuni contesti)
- Progetti (denominati app o applicazioni in alcuni contesti)
- Modelli di raccomandazioni (indicati come raccomandatori in alcuni contesti)
- Segmenti

Ad esempio, per creare un'istruzione di policy per il progetto con ID progetto `810c7aab86d42fb2b56c8c966example`, utilizza il seguente ARN:

```
"Resource": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

Per specificare tutti i progetti che appartengono ad un account specifico, utilizza il carattere jolly (*):

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

Alcune azioni Amazon Pinpoint, ad esempio la creazione di risorse, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (*).

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

Nelle policy IAM, puoi anche specificare ARNs i seguenti tipi di risorse Amazon Pinpoint SMS e Voice:

- Set di configurazione
- Elenco di esclusione
- Numero di telefono
- Pool
- ID mittente

Ad esempio, in una policy per creare un'istruzione per un numero di telefono associato l'ID `phone-12345678901234567890123456789012`, utilizza il seguente ARN:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-12345678901234567890123456789012"
```

Per specificare tutti i numeri di telefono appartenenti a un account specifico, utilizza un carattere jolly (*) al posto dell'ID del numero di telefono:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

Alcune azioni di tipo SMS e Voce di Amazon Pinpoint non vengono eseguite su una risorsa specifica, ad esempio le azioni di gestione delle impostazioni a livello di account come i limiti di spesa. In questi casi, è necessario utilizzare il carattere jolly (*).

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

Molte azioni API Amazon Pinpoint interessano più risorse. Ad esempio, l'operazione `TagResource` può aggiungere un tag a più progetti. Per specificare più risorse in una singola istruzione, separale ARNs con virgole:

```
"Resource": [
      "resource1",
      "resource2"
```

Per visualizzare un elenco dei tipi di risorse Amazon Pinpoint e relativi ARNs, consulta Resources Defined by Amazon Pinpoint nella IAM User Guide. Per informazioni sulle azioni che è possibile specificare con l'ARN di ogni tipo di risorsa, consulta Operazioni definite da Amazon Pinpoint nella Guida per l'utente di IAM.

Chiavi di condizione

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Amazon Pinpoint definisce il proprio set di chiavi di condizione, nonché supporta alcune chiavi di condizione globali. Per visualizzare un elenco di tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM. Per visualizzare l'elenco delle chiavi condizione di Amazon Pinpoint, consulta Chiavi di condizione per Amazon Pinpoint nella Guida dell'utente di IAM. Per informazioni su azioni e risorse con cui è possibile utilizzare una chiave di condizione, consulta Operazioni definite da Amazon Pinpoint nella Guida dell'utente di IAM.

Esempi

Per visualizzare esempi di policy basate su identità Amazon Pinpoint, consulta Esempi di policy basate su identità Amazon Pinpoint.

Policy di autorizzazione basate su risorse di Amazon Pinpoint

Le policy di autorizzazione basate su risorse sono documenti di policy JSON che specificano le azioni che possono essere eseguite da un'entità principale specificata su una risorsa Amazon Pinpoint e in base a quali condizioni. Amazon Pinpoint supporta policy di autorizzazione basate su risorse per campagne, percorsi, modelli di messaggi (modelli), modelli di raccomandazione, progetti (app) e segmenti.

Esempi

Per visualizzare esempi di policy basate su risorse Amazon Pinpoint, consulta the section called “Esempi di policy basate su identità”.

Autorizzazione basata sui tag Amazon Pinpoint

Puoi associare i tag a determinati tipi di risorse Amazon Pinpoint o passare i tag in una richiesta ad Amazon Pinpoint. Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/${TagKey}`, `aws:RequestTag/${TagKey}` o `aws:TagKeys`.

Per informazioni sull'assegnazione di tag alle risorse Amazon Pinpoint, inclusa una policy IAM di esempio, consulta Gestione dei tag delle risorse di Amazon Pinpoint.

Ruoli IAM Amazon Pinpoint

Un ruolo IAM è un'entità all'interno dell'account AWS che dispone di autorizzazioni specifiche.

Utilizzo di credenziali temporanee con Amazon Pinpoint

Puoi utilizzare le credenziali temporanee per effettuare l'accesso utilizzando la federazione, assumere un ruolo IAM o assumere un ruolo tra più account. È possibile ottenere credenziali di sicurezza temporanee chiamando operazioni API AWS Security Token Service (AWS STS) come AssumeRoleo GetFederationToken.

Amazon Pinpoint supporta l'uso di credenziali temporanee.

Ruoli collegati ai servizi

I ruoli collegati ai AWS servizi consentono ai servizi di accedere alle risorse di altri servizi per completare un'azione per conto dell'utente. I ruoli collegati ai servizi sono visualizzati nell'account IAM e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non può modificarle.

Amazon Pinpoint non utilizza ruoli collegati ai servizi.

Ruoli dei servizi

Questa caratteristica consente a un servizio di assumere un ruolo di servizio per conto dell'utente. Questo ruolo consente al servizio di accedere alle risorse in altri servizi per completare un'azione per conto dell'utente. I ruoli dei servizi sono visualizzati nell'account IAM e sono di proprietà dell'account. Ciò significa che un amministratore IAM può modificare le autorizzazioni per questo ruolo. Tuttavia, questo potrebbe pregiudicare la funzionalità del servizio.

Amazon Pinpoint supporta l'utilizzo dei ruoli di servizio.

Azioni di Amazon Pinpoint per le policy IAM

Per gestire l'accesso alle risorse Amazon Pinpoint nel tuo AWS account, puoi aggiungere azioni Amazon Pinpoint AWS Identity and Access Management alle policy (IAM). Con le azioni aggiunte alle policy, puoi controllare ciò che possono fare gli utenti mediante la console Amazon Pinpoint. Puoi anche controllare cosa possono fare gli utenti a livello di codice utilizzando direttamente AWS SDKs, the AWS Command Line Interface (AWS CLI) o Amazon APIs Pinpoint.

In una policy si specificano le singole azioni con lo spazio dei nomi Amazon Pinpoint appropriato seguito da due punti (:) e dal nome dell'operazione, ad esempio `GetSegments`. La maggior parte delle azioni corrisponde a una richiesta all'API Amazon Pinpoint utilizzando un URI e un metodo HTTP specifici. Ad esempio, se consenti l'operazione `mobiletargeting:GetSegments` nella

policy di un utente, l'utente può recuperare le informazioni su tutti i segmenti per un progetto inviando una richiesta HTTP GET all'URI `/apps/projectId/segments`. Questa policy consente inoltre all'utente di visualizzare tali informazioni sulla console e recuperarle utilizzando un AWS SDK o il. AWS CLI

Ogni azione viene eseguita su una specifica risorsa Amazon Pinpoint, identificata in un'istruzione della policy mediante il relativo nome della risorsa Amazon (ARN). Ad esempio, l'operazione `mobiletargeting:GetSegments` viene eseguita su un progetto specifico, identificato con l'ARN `arn:aws:mobiletargeting:region:accountId:apps/projectId`.

In questo argomento vengono descritte le azioni Amazon Pinpoint che puoi aggiungere alle policy IAM per l'account AWS . Per esempi che illustrano come utilizzare le azioni nelle policy per gestire l'accesso alle risorse Amazon Pinpoint, consulta Esempi di policy basate su identità Amazon Pinpoint.

Argomenti

- Azioni dell'API Amazon Pinpoint
- Azioni dell'API SMS e Voce versione 1 di Amazon Pinpoint

Azioni dell'API Amazon Pinpoint

In questa sezione vengono descritte le azioni associate alle funzionalità disponibili nell'API Amazon Pinpoint, che rappresenta l'API primaria di Amazon Pinpoint. Per ulteriori informazioni su questa API, consulta la documentazione di riferimento dell'API Amazon Pinpoint.

Categorie:

- Analisi e parametri
- Campagne
- Canali
- Endpoints
- Flussi di eventi
- Eventi
- Processi di esportazione
- Processi di importazione
- Percorsi

- Modelli dei messaggi
- Messaggi
- Codici OTP (One-Time Password)
- Convalida del numero di telefono
- Progetti
- Modelli della funzione di suggerimento
- Segmenti
- Tag
- Utenti

Analisi e parametri

Le seguenti autorizzazioni fanno riferimento alla visualizzazione dei dati di analisi nella console Amazon Pinpoint. Sono inoltre correlati al recupero (interrogazione) di dati aggregati per metriche standard, note anche come indicatori chiave di performance (KPIs), che si applicano a progetti, campagne e percorsi.

`mobiletargeting:GetReports`

Visualizza i dati di analisi nella console Amazon Pinpoint. Questa autorizzazione è necessaria anche per creare segmenti contenenti attributi personalizzati mediante la console Amazon Pinpoint. È inoltre necessario ottenere una stima delle dimensioni di un segmento nella console Amazon Pinpoint.

- URI: non applicabile
- Metodo: non applicabile
- ARN risorsa: `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:*`

`mobiletargeting:GetApplicationDateRangeKpi`

Recupera (esegue query) i dati aggregati per un parametro dell'applicazione standard. Si tratta di un parametro che si applica a tutte le campagne o ai messaggi transazionali associati a un progetto.

- URI: `/apps/`*`projectId`*`/kpis/daterange/`*`kpi-name`*
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:apps/`*`projectId`*`/kpis/daterange/`*`kpi-name`*

mobiletargeting:GetCampaignDateRangeKpi

Recupera (esegue query) i dati aggregati per un parametro di campagna standard. Si tratta di un parametro che si applica a una singola campagna.

- URI: /apps/*projectId*/campaigns/*campaignId*/kpis/daterange/*kpi-name*
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/campaigns/`*campaignId*`/kpis/daterange/`*kpi-name*

mobiletargeting:GetJourneyDateRangeKpi

Recupera (esegue query) i dati aggregati per un parametro di coinvolgimento viaggi standard. Si tratta di una metrica di coinvolgimento che si applica a un singolo percorso, ad esempio il numero di messaggi aperti dai partecipanti per tutte le attività di un percorso.

- URI: /apps/*projectId*/journeys/*journeyId*/kpis/daterange/*kpi-name*
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/journeys/`*journeyId*`/kpis/daterange/`*kpi-name*

mobiletargeting:GetJourneyExecutionMetrics

Recupera (mediante query) i dati aggregati per le metriche di esecuzione standard applicabili a un singolo percorso, ad esempio il numero di partecipanti coinvolti attivamente in tutte le attività di un percorso.

- URI: /apps/*projectId*/journeys/*journeyId*/execution-metrics
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/journeys/`*journeyId*`/execution-metrics`

mobiletargeting:GetJourneyExecutionActivityMetrics

Recupera (mediante query) i dati aggregati per le metriche di esecuzione standard applicabili a una singola attività in un percorso, ad esempio il numero di partecipanti che hanno iniziato o completato un'attività.

- URI: /apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics
- Metodo – GET

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/activities/journey-activity-id/execution-metrics`

Campagne

Le autorizzazioni seguenti fanno riferimento alla gestione delle campagne nell'account Amazon Pinpoint.

`mobiletargeting:CreateCampaign`

Creare una campagna per un progetto.

- URI: `/apps/projectId/campaigns`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns`

`mobiletargeting:DeleteCampaign`

Eliminare una determinata campagna.

- URI: `/apps/projectId/campaigns/campaignId`
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:GetCampaign`

Recuperare informazioni su una determinata partizione.

- URI: `/apps/projectId/campaigns/campaignId`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:GetCampaignActivities`

Recuperare informazioni sulle attività eseguite da una campagna.

- URI: `/apps/projectId/campaigns/campaignId/activities`
- Metodo – GET

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:GetCampaigns`

Recuperare informazioni su tutte le campagne per un progetto.

- URI: `/apps/projectId/campaigns`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetCampaignVersion`

Recuperare informazioni su una determinata versione di una campagna.

- URI: `/apps/projectId/campaigns/campaignId/versions/versionId`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:GetCampaignVersions`

Recuperare informazioni sulla versione corrente e sulle versioni precedenti di una campagna.

- URI: `/apps/projectId/campaigns/campaignId/versions`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

`mobiletargeting:UpdateCampaign`

Aggiornare una determinata campagna.

- URI: `/apps/projectId/campaigns/campaignId`
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

Canali

Le autorizzazioni seguenti fanno riferimento alla gestione dei canali nell'account Amazon Pinpoint. In Amazon Pinpoint, per canali si intendono i metodi utilizzati per contattare i clienti, come l'invio di messaggi e-mail, SMS o notifiche push.

mobiletargeting:DeleteAdmChannel

Disabilitare il canale Amazon Device Messaging (ADM) per un progetto.

- URI: /apps/*projectId*/channels/adm
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:GetAdmChannel

Recuperare informazioni sul canale ADM per un progetto.

- URI: /apps/*projectId*/channels/adm
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:UpdateAdmChannel

Abilitare o aggiornare il canale ADM per un progetto.

- URI: /apps/*projectId*/channels/adm
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:DeleteApnsChannel

Disattiva il canale del servizio Apple Push Notification () APNs per un progetto.

- URI: /apps/*projectId*/channels/apns
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:GetApnsChannel

Recupera informazioni sul APNs canale di un progetto.

- URI: /apps/*projectId*/channels/apns
- Metodo – GET

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

`mobiletargeting:UpdateApnsChannel`

Abilita o aggiorna il APNs canale per un progetto.

- URI: /apps/*projectId*/channels/apns
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns`

`mobiletargeting:DeleteApnsSandboxChannel`

Disattiva il canale APNs sandbox per un progetto.

- URI: /apps/*projectId*/channels/apns_sandbox
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:GetApnsSandboxChannel`

Recupera informazioni sul canale APNs sandbox per un progetto.

- URI: /apps/*projectId*/channels/apns_sandbox
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:UpdateApnsSandboxChannel`

Abilita o aggiorna il canale APNs sandbox per un progetto.

- URI: /apps/*projectId*/channels/apns_sandbox
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:DeleteApnsVoipChannel`

Disattiva il canale APNs VoIP per un progetto.

- URI: /apps/*projectId*/channels/apns_voip

- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:GetApnsVoipChannel`

Recupera informazioni sul canale APNs VoIP per un progetto.

- URI: /apps/*projectId*/channels/apns_voip
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:UpdateApnsVoipChannel`

Abilita o aggiorna il canale APNs VoIP per un progetto.

- URI: /apps/*projectId*/channels/apns_voip
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:DeleteApnsVoipSandboxChannel`

Disattiva il canale sandbox APNs VoIP per un progetto.

- URI: /apps/*projectId*/channels/apns_voip_sandbox
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

`mobiletargeting:GetApnsVoipSandboxChannel`

Recupera informazioni sul canale sandbox APNs VoIP per un progetto.

- URI: /apps/*projectId*/channels/apns_voip_sandbox
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

`mobiletargeting:UpdateApnsVoipSandboxChannel`

Abilita o aggiorna il canale sandbox APNs VoIP per un progetto.

- URI: /apps/*projectId*/channels/apns_voip_sandbox
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip_sandbox`

`mobiletargeting:DeleteBaiduChannel`

Disabilitare il canale Baidu Cloud Push per un progetto.

- URI: /apps/*projectId*/channels/baidu
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/baidu`

`mobiletargeting:GetBaiduChannel`

Recuperare informazioni sul canale Baidu Cloud Push per un progetto.

- URI: /apps/*projectId*/channels/baidu
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/baidu`

`mobiletargeting:UpdateBaiduChannel`

Abilitare o aggiornare il canale Baidu Cloud Push per un progetto.

- URI: /apps/*projectId*/channels/baidu
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/baidu`

`mobiletargeting:DeleteEmailChannel`

Disabilitare il canale e-mail per un progetto.

- URI: /apps/*projectId*/channels/email
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/email`

`mobiletargeting:GetEmailChannel`

Recuperare informazioni sul canale e-mail per un progetto.

- URI: /apps/*projectId*/channels/email
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/email`

`mobiletargeting:UpdateEmailChannel`

Abilitare o aggiornare il canale e-mail per un progetto.

- URI: /apps/*projectId*/channels/email
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/email`

`mobiletargeting:DeleteGcmChannel`

Disabilita il canale Firebase Cloud Messaging (FCM) per un progetto. Questo canale consente a Amazon Pinpoint di inviare notifiche push a un'app Android tramite il servizio FCM, che sostituisce il servizio Google Cloud Messaging (GCM).

- URI: /apps/*projectId*/channels/gcm
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/gcm`

`mobiletargeting:GetGcmChannel`

Recupera le informazioni sul canale FCM per un progetto. Questo canale consente a Amazon Pinpoint di inviare notifiche push a un'app Android tramite il servizio FCM, che sostituisce il servizio Google Cloud Messaging (GCM).

- URI: /apps/*projectId*/channels/gcm
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/gcm`

`mobiletargeting:UpdateGcmChannel`

Attiva o aggiorna il canale FCM per un progetto. Questo canale consente a Amazon Pinpoint di inviare notifiche push a un'app Android tramite il servizio FCM, che sostituisce il servizio Google Cloud Messaging (GCM).

- URI: /apps/*projectId*/channels/gcm
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:DeleteSmsChannel

Disabilitare il canale SMS per un progetto.

- URI: /apps/*projectId*/channels/sms
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetSmsChannel

Recuperare informazioni sul canale SMS per un progetto.

- URI: /apps/*projectId*/channels/sms
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:UpdateSmsChannel

Abilitare o aggiornare il canale SMS per un progetto.

- URI: /apps/*projectId*/channels/sms
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetChannels

Recupera le informazioni sulla cronologia e lo stato di ciascun canale per un'applicazione.

- URI: /apps/*application-id*/channels
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/channels

mobiletargeting:DeleteVoiceChannel

Disattiva il canale vocale per un'applicazione ed elimina tutte le impostazioni esistenti per il canale.

- URI: /apps/*application-id*/channels/voice
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectid*`/channels/voice`

mobiletargeting:GetVoiceChannel

Recupera le informazioni sullo stato e sulle impostazioni del canale vocale per un'applicazione.

- URI: /apps/*application-id*/channels/voice
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectid*`/channels/voice`

mobiletargeting:UpdateVoiceChannel

Abilita il canale vocale per un'applicazione o aggiorna lo stato e le impostazioni del canale vocale per un'applicazione.

- URI: /apps/*application-id*/channels/voice
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectid*`/channels/voice`

Endpoints

Le autorizzazioni seguenti fanno riferimento alla gestione degli endpoint nell'account Amazon Pinpoint. In Amazon Pinpoint, per endpoint si intende una singola destinazione per i messaggi. Ad esempio, un endpoint può essere l'indirizzo e-mail di un cliente, un numero di telefono o un token di dispositivo mobile.

mobiletargeting:DeleteEndpoint

Eliminare un endpoint.

- URI: /apps/*projectId*/endpoints/*endpointId*
- Metodo: DELETE

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

`mobiletargeting:GetEndpoint`

Recuperare informazioni su un determinato endpoint.

- URI: `/apps/projectId/endpoints/endpointId`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

`mobiletargeting:RemoveAttributes`

Rimuove uno o più attributi dello stesso tipo da tutti gli endpoint associati a un'applicazione.

- URI: `apps/application-id/attributes/attribute-type`
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/attributes/attribute-type`

`mobiletargeting:UpdateEndpoint`

Creare un endpoint o aggiornare le informazioni per un endpoint.

- URI: `/apps/projectId/endpoints/endpointId`
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

`mobiletargeting:UpdateEndpointsBatch`

Creare o aggiornare endpoint con un'operazione batch.

- URI: `/apps/projectId/endpoints`
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Flussi di eventi

Le autorizzazioni seguenti fanno riferimento alla gestione dei flussi di eventi nell'account Amazon Pinpoint.

mobiletargeting:DeleteEventStream

Eliminare il flusso di eventi per un progetto.

- URI: /apps/*projectId*/eventstream/
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:GetEventStream

Recuperare informazioni sul flusso di eventi per un progetto.

- URI: /apps/*projectId*/eventstream/
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:PutEventStream

Creare o aggiornare un flusso di eventi per un progetto.

- URI: /apps/*projectId*/eventstream/
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

Eventi

Le autorizzazioni seguenti fanno riferimento alla gestione dei processi relativi agli eventi nell'account Amazon Pinpoint. In Amazon Pinpoint, crei processi di importazione per creare segmenti in base alle definizioni di endpoint archiviate in un bucket Amazon S3.

mobiletargeting:PutEvents

Crea un nuovo evento da registrare per gli endpoint oppure crea o aggiorna i dati degli endpoint a cui sono associati gli eventi esistenti.

- URI: /apps/*application-id*/events
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/events

Processi di esportazione

Le autorizzazioni seguenti fanno riferimento alla gestione dei processi di esportazione nell'account Amazon Pinpoint. In Amazon Pinpoint, crei processi di esportazione per inviare informazioni sugli endpoint a un bucket Amazon S3 per l'archiviazione o per l'analisi.

`mobiletargeting:CreateExportJob`

Crea un processo di esportazione per esportare le definizioni di endpoint in Amazon S3.

- URI: `/apps/projectId/jobs/export`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

`mobiletargeting:GetExportJob`

Recuperare informazioni su un determinato processo di esportazione per un progetto.

- URI: `/apps/projectId/jobs/export/jobId`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export/jobId`

`mobiletargeting:GetExportJobs`

Recuperare un elenco di tutti i processi di esportazione per un progetto.

- URI: `/apps/projectId/jobs/export`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

Processi di importazione

Le autorizzazioni seguenti fanno riferimento alla gestione dei processi di importazione nell'account Amazon Pinpoint. In Amazon Pinpoint, crei processi di importazione per creare segmenti in base alle definizioni di endpoint archiviate in un bucket Amazon S3.

`mobiletargeting:CreateImportJob`

Importa le definizioni di endpoint da Amazon S3 per creare un segmento.

- URI: /apps/*projectId*/jobs/import
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetImportJob

Recuperare informazioni su un determinato processo di importazione per un progetto.

- URI: /apps/*projectId*/jobs/import/*jobId*
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/jobs/import/*jobId*

mobiletargeting:GetImportJobs

Recuperare informazioni su tutti i processi di importazione per un progetto.

- URI: /apps/*projectId*/jobs/import
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

Percorsi

Le autorizzazioni seguenti fanno riferimento alla gestione dei percorsi nell'account Amazon Pinpoint.

mobiletargeting:CreateJourney

Creare un viaggio per un progetto.

- URI: /apps/*projectId*/journeys
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys

mobiletargeting:GetJourney

Recuperare informazioni su un determinato viaggio.

- URI: /apps/*projectId*/journeys/*journeyId*
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

mobiletargeting:ListJourneys

Recuperare informazioni sui viaggi per un progetto.

- URI: /apps/*projectId*/journeys
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys

mobiletargeting:UpdateJourney

Aggiorna la configurazione e altre impostazioni per un viaggio specifico.

- URI: /apps/*projectId*/journeys/*journeyId*
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

mobiletargeting:UpdateJourneyState

Annullare un viaggio attivo.

- URI: /apps/*projectId*/journeys/*journeyId*/state
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/state

mobiletargeting:DeleteJourney

Eliminare un viaggio specifico.

- URI: /apps/*projectId*/journeys/*journeyId*
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

Modelli dei messaggi

Le seguenti autorizzazioni fanno riferimento alla creazione e alla gestione dei modelli di messaggio per l'account Amazon Pinpoint. Un modello di messaggio è un insieme di contenuti e impostazioni che puoi definire, salvare e riutilizzare nei messaggi inviati per ognuno dei tuoi progetti Amazon Pinpoint.

mobiletargeting:ListTemplates

Recupera le informazioni su tutti i modelli di messaggio associati all'account Amazon Pinpoint.

- URI: /templates
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates

mobiletargeting:ListTemplateVersions

Recuperare informazioni su tutte le versioni di un modello di messaggio specifico.

- URI: /templates/*template-name*/*template-type*/versions
- Metodo – GET
- ARN risorsa: non applicabile

mobiletargeting:UpdateTemplateActiveVersion

Designare una versione specifica di un modello di messaggio come versione attiva del modello.

- URI: /templates/*template-name*/*template-type*/active-version
- Metodo – GET
- ARN risorsa: non applicabile

mobiletargeting:GetEmailTemplate

Recuperare informazioni su un modello di messaggio per i messaggi inviati attraverso il canale di posta elettronica.

- URI: /templates/*template-name*/email
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:CreateEmailTemplate

Creare un modello di messaggio per i messaggi inviati attraverso il canale di posta elettronica.

- URI: /templates/*template-name*/email
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:UpdateEmailTemplate

Aggiornare un modello di messaggio esistente per i messaggi inviati attraverso il canale di posta elettronica.

- URI: /templates/*template-name*/email
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:DeleteEmailTemplate

Eliminare un modello di messaggio per i messaggi inviati attraverso il canale di posta elettronica.

- URI: /templates/*template-name*/email
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:GetPushTemplate

Recuperare informazioni su un modello di messaggio per i messaggi inviati tramite un canale di notifica push.

- URI: /templates/*template-name*/push
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:CreatePushTemplate

Creare un modello di messaggio per i messaggi inviati tramite un canale di notifica push.

- URI: /templates/*template-name*/push
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:UpdatePushTemplate

Aggiornare un modello di messaggio esistente per i messaggi inviati tramite un canale di notifica push.

- URI: /templates/*template-name*/push
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:DeletePushTemplate

Eliminare un modello di messaggio per i messaggi inviati tramite un canale di notifica push.

- URI: /templates/*template-name*/push
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:GetSmsTemplate

Recuperare informazioni su un modello di messaggio per i messaggi inviati tramite il canale SMS.

- URI: /templates/*template-name*/sms
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:CreateSmsTemplate

Creare un modello di messaggio per i messaggi inviati attraverso il canale SMS.

- URI: /templates/*template-name*/sms
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:UpdateSmsTemplate

Aggiornare un modello di messaggio esistente per i messaggi inviati tramite il canale SMS.

- URI: /templates/*template-name*/sms
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:DeleteSmsTemplate

Eliminare un modello di messaggio per i messaggi inviati tramite il canale SMS.

- URI: /templates/*template-name*/sms
- Metodo: DELETE
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:GetVoiceTemplate

Recuperare informazioni su un modello di messaggio per i messaggi inviati tramite il canale vocale.

- URI: /templates/*template-name*/voice
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:CreateVoiceTemplate

Creare un modello di messaggio per i messaggi inviati attraverso il canale vocale.

- URI: /templates/*template-name*/voice
- Metodo – POST

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

`mobiletargeting:UpdateVoiceTemplate`

Aggiornare un modello di messaggio esistente per i messaggi inviati tramite il canale vocale.

- URI: `/templates/template-name/voice`
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

`mobiletargeting:DeleteVoiceTemplate`

Eliminare un modello di messaggio per i messaggi inviati tramite il canale vocale.

- URI: `/templates/template-name/voice`
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

Messaggi

Le autorizzazioni seguenti fanno riferimento all'invio di messaggi e notifiche push dall'account Amazon Pinpoint. È possibile utilizzare le operazioni `SendMessages` e `SendUsersMessages` per inviare messaggi a specifici endpoint senza prima creare segmenti e campagne.

`mobiletargeting:SendMessages`

Inviare un messaggio o una notifica push a determinati endpoint.

- URI: `/apps/projectId/messages`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

`mobiletargeting:SendUsersMessages`

Inviare un messaggio o una notifica push a tutti gli endpoint associati a un determinato ID utente.

- URI: `/apps/projectId/users-messages`
- Metodo – POST

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

Codici OTP (One-Time Password)

Le seguenti autorizzazioni sono relative all'invio e alla convalida di password monouso () in OTPs Amazon Pinpoint.

`mobiletargeting:SendOTPMessage`

Invia un messaggio di testo contenente un codice OTP (One-Time Password).

- URI: `/apps/projectId/otp`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/otp`

`mobiletargeting:VerifyOTPMessage`

Verifica la validità di una password monouso (OTP) generata utilizzando l'operazione di invio. OTPMessage

- URI: `/apps/projectId/verify-otp`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/verify-otp`

Convalida del numero di telefono

Le autorizzazioni seguenti fanno riferimento all'utilizzo del servizio di convalida dei numeri di telefono in Amazon Pinpoint.

`mobiletargeting:PhoneNumberValidate`

Recuperare informazioni su un numero di telefono.

- URI: `/phone/number/validate`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:phone/number/validate`

Progetti

Le autorizzazioni seguenti fanno riferimento alla gestione dei progetti nell'account Amazon Pinpoint. In origine, i progetti erano denominati applicazioni. Nell'ambito di queste operazioni, un'applicazione Amazon Pinpoint coincide con un progetto Amazon Pinpoint.

`mobiletargeting:CreateApp`

Creare un progetto Amazon Pinpoint

- URI: `/apps`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:DeleteApp`

Elimina un progetto Amazon Pinpoint.

- URI: `/apps/projectId`
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApp`

Recupera le informazioni su un progetto Amazon Pinpoint.

- URI: `/apps/projectId`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApps`

Recupera le informazioni su tutti i progetti associati all'account Amazon Pinpoint.

- URI: `/apps`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:GetApplicationSettings`

Recupera le impostazioni predefinite per un progetto Amazon Pinpoint.

- URI: `/apps/projectId/settings`
- Metodo – GET

- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:UpdateApplicationSettings`

Aggiorna le impostazioni predefinite per un progetto Amazon Pinpoint.

- URI: `/apps/projectId/settings`
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Modelli della funzione di suggerimento

Le seguenti autorizzazioni fanno riferimento alla gestione delle configurazioni Amazon Pinpoint per il recupero e l'elaborazione dei dati delle raccomandazioni derivanti dai modelli di raccomandazione. Un modello della funzione di suggerimento è un tipo di modello di machine learning che prevede e genera suggerimenti personalizzati individuando pattern nei dati.

`mobiletargeting:CreateRecommenderConfiguration`

Crea una configurazione Amazon Pinpoint per un modello della funzione di suggerimento.

- URI: `/recommenders`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:recommenders`

`mobiletargeting:GetRecommenderConfigurations`

Recupera le informazioni su tutte le configurazioni del modello di raccomandazione associate all'account Amazon Pinpoint.

- URI: `/recommenders`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:recommenders`

`mobiletargeting:GetRecommenderConfiguration`

Recupera le informazioni su una configurazione di Amazon Pinpoint per un modello di raccomandazione.

- URI: `/recommenders/recommenderId`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

`mobiletargeting:UpdateRecommenderConfiguration`

Aggiorna una configurazione Amazon Pinpoint per un modello della funzione di suggerimento.

- URI: `/recommenders/recommenderId`
- Method – PUT
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

`mobiletargeting:DeleteRecommenderConfiguration`

Elimina una configurazione Amazon Pinpoint per un modello della funzione di suggerimento.

- URI: `/recommenders/recommenderId`
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

Segmenti

Le autorizzazioni seguenti fanno riferimento alla gestione dei segmenti nell'account Amazon Pinpoint. In Amazon Pinpoint, per segmenti si intendono gruppi di destinatari per le campagne che condividono determinati attributi specificati in precedenza.

`mobiletargeting:CreateSegment`

Creare un segmento. Per permettere a un utente di creare un segmento importando i dati degli endpoint dall'esterno di Amazon Pinpoint, è necessario consentire l'azione `mobiletargeting:CreateImportJob`.

- URI: `/apps/projectId/segments`
- Metodo – POST
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:DeleteSegment`

Eliminare un segmento.

- URI: `/apps/projectId/segments/segmentId`
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

mobiletargeting:GetSegment

Recuperare informazioni su un determinato segmento.

- URI: /apps/*projectId*/segments/*segmentId*
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentExportJobs

Recuperare informazioni sui processi che esportano definizioni di endpoint per un segmento.

- URI: /apps/*projectId*/segments/*segmentId*/jobs/export
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*/jobs/export

mobiletargeting:GetSegments

Recuperare informazioni sui segmenti per un progetto.

- URI: /apps/*projectId*/segments
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetSegmentImportJobs

Recupera le informazioni sui processi che creano segmenti mediante l'importazione delle definizioni di endpoint da Amazon S3.

- URI: /apps/*projectId*/segments/*segmentId*/jobs/import
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersion

Recuperare informazioni su una determinata versione di un segmento.

- URI: /apps/*projectId*/segments/*segmentId*/versions/*versionId*
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersions

Recuperare informazioni sulla versione corrente e sulle versioni precedenti di un segmento.

- URI: /apps/*projectId*/segments/*segmentId*/versions
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:UpdateSegment

Aggiornare un determinato segmento.

- URI: /apps/*projectId*/segments/*segmentId*
- Method – PUT
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

Tag

Le autorizzazioni seguenti fanno riferimento alla visualizzazione e alla gestione dei tag per le risorse Amazon Pinpoint.

mobiletargeting:ListTagsForResource

Recuperare informazioni sui tag associati a un progetto, una campagna, un modello di messaggio o un segmento.

- URI: /tags/*resource-arn*
- Metodo – GET
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:*

mobiletargeting:TagResource

Aggiungere uno o più tag a un progetto, una campagna, un modello di messaggio o un segmento.

- URI: /tags/*resource-arn*
- Metodo – POST
- ARN risorsa: arn:aws:mobiletargeting:*region*:*accountId*:*

`mobiletargeting:UntagResource`

Rimuovere uno o più tag da un progetto, una campagna, un modello di messaggio o un segmento.

- URI: `/tags/resource-arn`
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:*`

Utenti

Le autorizzazioni seguenti sono correlate alla gestione degli utenti. In Amazon Pinpoint, per utenti si intendono le persone che ricevono messaggi. Un singolo utente può essere associato a più endpoint.

`mobiletargeting:DeleteUserEndpoints`

Eliminare tutti gli endpoint associati a un ID utente.

- URI: `/apps/projectId/users/userId`
- Metodo: DELETE
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

`mobiletargeting:GetUserEndpoints`

Recuperare informazioni su tutti gli endpoint associati a un ID utente.

- URI: `/apps/projectId/users/userId`
- Metodo – GET
- ARN risorsa: `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

Azioni dell'API SMS e Voce versione 1 di Amazon Pinpoint

In questa sezione vengono identificate le funzionalità disponibili nell'API SMS e Voce di Amazon Pinpoint. Si tratta di un'API supplementare che fornisce opzioni avanzate per l'utilizzo e la gestione dei canali SMS e voce in Amazon Pinpoint. Per ulteriori informazioni su questa API, consulta la documentazione di riferimento dell'API SMS e Voce di Amazon Pinpoint.

`sms-voice:CreateConfigurationSet`

Creare un set di configurazione per l'invio di messaggi vocali.

- URI: `/sms-voice/configuration-sets`
- Metodo – POST
- ARN risorsa: non disponibile. Utilizza `*`.

`sms-voice:DeleteConfigurationSet`

Eliminare un set di configurazione per l'invio di messaggi vocali.

- URI — /sms-voice/configuration-sets/ *`ConfigurationSetName`*
- Metodo: DELETE
- ARN risorsa: non disponibile. Utilizza `*`.

`sms-voice:GetConfigurationSetEventDestinations`

Richiamare informazioni su un set di configurazione e sulle destinazioni degli eventi che contiene.

- URI — /sms-voice/configuration-sets/ *`ConfigurationSetName`* /event-destinazioni-
- Metodo – GET
- ARN risorsa: non disponibile. Utilizza `*`.

`sms-voice:CreateConfigurationSetEventDestination`

Creare una destinazione evento per gli eventi vocali.

- URI — /sms-voice/configuration-sets/ *`ConfigurationSetName`* /event-destinazioni-
- Metodo – POST
- ARN risorsa: non disponibile. Utilizza `*`.

`sms-voice:UpdateConfigurationSetEventDestination`

Aggiornare una destinazione evento per gli eventi vocali.

- URI — /sms-voice/configuration-sets/ *`ConfigurationSetName`* /event-destinazioni/ *`EventDestinationName`*
- Method – PUT

- ARN risorsa: non disponibile. Utilizza *.

`sms-voice:DeleteConfigurationSetEventDestination`

Eliminare una destinazione evento per gli eventi vocali.

- URI — /sms-voice/configuration-sets/ *`ConfigurationSetName`* /event-destinazioni/ *`EventDestinationName`*
- Metodo: DELETE
- ARN risorsa: non disponibile. Utilizza *.

`sms-voice:SendVoiceMessage`

Creare e inviare messaggi vocali.

- URI — /sms- voice/voice/message
- Metodo – POST
- ARN risorsa: non disponibile. Utilizza *.

Esempi di policy basate su identità Amazon Pinpoint

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse Amazon Pinpoint. Inoltre, non possono eseguire attività utilizzando AWS Management Console AWS CLI, o un'API. AWS Un amministratore IAM deve creare policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse di cui hanno bisogno. L'amministratore devi quindi collegare queste policy a utenti o gruppi che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consultare Creazione di policy nella scheda JSON nella Guida per l'utente di IAM.

Argomenti

- Best practice delle policy
- Utilizzo della console Amazon Pinpoint
- Esempio: accesso a un singolo progetto Amazon Pinpoint
- Esempio: visualizzazione di risorse Amazon Pinpoint in base ai tag

- Esempio: consentire agli utenti di visualizzare le loro autorizzazioni
- Esempi: autorizzazione dell'accesso alle azioni dell'API Amazon Pinpoint
- Esempi: autorizzazione dell'accesso alle azioni dell'API SMS e Voce di Amazon Pinpoint
- Esempio: limitazione dell'accesso del progetto Amazon Pinpoint a indirizzi IP specifici
- Esempio: limitazione dell'accesso di Amazon Pinpoint in base ai tag
- Esempio: autorizzazione all'invio di e-mail da parte di Amazon Pinpoint mediante identità verificate in Amazon SES

Best practice delle policy

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon Pinpoint nell'account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100

controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente IAM.

- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console Amazon Pinpoint

Per accedere alla console Amazon Pinpoint, è necessario disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon Pinpoint nel AWS tuo account. Se crei una policy basata su identità che applica autorizzazioni più restrittive rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti e ruoli) associate a tale policy. Per garantire che tali entità possano utilizzare la console Amazon Pinpoint, collega una policy alle entità. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente IAM.

La seguente policy di esempio fornisce l'accesso in sola lettura alla console Amazon Pinpoint in una regione specifica. AWS Include l'accesso in sola lettura ad altri servizi da cui dipende la console Amazon Pinpoint, come Amazon Simple Email Service (Amazon SES), IAM e 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"
                }
            }
        }
    ]
}
```

Nell'esempio di politica precedente, sostituiscilo *region* con il nome di una AWS regione e sostituiscilo *accountId* con l'ID del tuo account. AWS

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Esempio: accesso a un singolo progetto Amazon Pinpoint

È anche possibile creare policy di sola lettura che forniscono l'accesso solo a progetti specifici. La policy di esempio seguente consente agli utenti di effettuare l'accesso alla console e visualizzare una lista di progetti. Consente inoltre agli utenti di visualizzare informazioni sulle risorse correlate per altri servizi AWS da cui dipende la console Amazon Pinpoint, ad esempio Amazon SES, IAM e Amazon Kinesis. Tuttavia, la policy consente agli utenti di visualizzare ulteriori informazioni solo sul progetto specificato nella policy. Puoi modificare questa politica per consentire l'accesso a progetti o AWS regioni aggiuntivi.

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

Nell'esempio precedente, sostituiscilo *region* con il nome di una AWS regione, sostituiscilo *accountId* con l'ID del tuo AWS account e sostituiscilo *projectId* con l'ID del progetto Amazon Pinpoint a cui desideri fornire l'accesso.

Allo stesso modo, puoi creare politiche che concedano a un utente del tuo AWS account un accesso limitato in scrittura a uno dei tuoi progetti Amazon Pinpoint, ad esempio il progetto con l'ID del

`810c7aab86d42fb2b56c8c966example` progetto. In questo caso, vuoi consentire all'utente di visualizzare, aggiungere e aggiornare i componenti del progetto, ad esempio segmenti e campagne, ma non di eliminare alcun componente.

Oltre a concedere le autorizzazioni per le operazioni `mobiletargeting:Get` e `mobiletargeting:List`, crea una policy che conceda le autorizzazioni per le operazioni seguenti: `mobiletargeting:Create`, `mobiletargeting:Update` e `mobiletargeting:Put`. Queste sono le autorizzazioni aggiuntive necessarie per creare e gestire la maggior parte dei componenti del progetto. Per esempio:

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

Esempio: visualizzazione di risorse Amazon Pinpoint in base ai tag

In una policy basata sulle identità puoi utilizzare le condizioni per controllare l'accesso alle risorse Amazon Pinpoint in base ai tag. In questa policy di esempio viene illustrato come creare questo tipo di policy per consentire la visualizzazione delle risorse Amazon Pinpoint. Tuttavia, l'autorizzazione viene concessa solo se il valore del tag del servizio `Owner` è quello del nome utente dell'utente. Questa policy concede anche le autorizzazioni necessarie per completare questa azione nella console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewResourceIfOwner",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
```

```
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:*:*:*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                },
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mobiletargeting:region:accountId:*"
                }
            }
        }
    ]
}
```

Puoi collegare questo tipo di policy agli utenti nel tuo account. Se un utente denominato `richard-roe` tenta di visualizzare una risorsa Amazon Pinpoint, la risorsa deve essere contrassegnata mediante il tag `Owner=richard-roe` o `owner=richard-roe`. In caso contrario l'accesso è negato. La chiave di tag di condizione `Owner` corrisponde a `Owner` e `owner` perché i nomi delle chiavi di condizione non effettuano la distinzione tra maiuscole e minuscole. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.

Esempio: consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o in modo programmatico. AWS CLI AWS

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

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

Esempi: autorizzazione dell'accesso alle azioni dell'API Amazon Pinpoint

In questa sezione vengono fornite policy di esempio che consentono l'accesso alle funzionalità disponibili nell'API Amazon Pinpoint, che è l'API primaria per Amazon Pinpoint. Per ulteriori informazioni su questa API, consulta la documentazione di riferimento dell'API Amazon Pinpoint.

Accesso in sola lettura

La seguente policy di esempio consente l'accesso in sola lettura a tutte le risorse del tuo account Amazon Pinpoint in una regione specifica. AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewAllResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
```

```
            ],
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        }
    ]
}
```

Nell'esempio precedente, sostituiscilo *region* con il nome di una AWS regione e sostituiscilo *accountId* con l'ID del tuo account. AWS

Accesso amministratore

La policy di esempio seguente consente l'accesso completo a tutte le azioni e le risorse Amazon Pinpoint dell'account Amazon Pinpoint:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:*"
            ],
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        }
    ]
}
```

Nell'esempio precedente, sostituiscilo *accountId* con l'ID del tuo AWS account.

Esempi: autorizzazione dell'accesso alle azioni dell'API SMS e Voce di Amazon Pinpoint

In questa sezione vengono fornite policy di esempio che consentono l'accesso alle funzionalità disponibili nell'API SMS e Voce di Amazon Pinpoint. Si tratta di un'API supplementare che fornisce opzioni avanzate per l'utilizzo e la gestione dei canali SMS e voce in Amazon Pinpoint. Per ulteriori informazioni su questa API, consulta la documentazione di riferimento dell'API SMS e Voce di Amazon Pinpoint.

Accesso in sola lettura

La seguente policy di esempio consente l'accesso in sola lettura a tutte le azioni e le risorse dell'API Amazon Pinpoint SMS e Voice nel tuo account: 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:*"
                }
            }
        }
    ]
}
```

Accesso amministratore

La seguente policy di esempio consente l'accesso completo a tutte le azioni e le risorse delle API Amazon Pinpoint SMS e Voice nel tuo AWS account:

```
{
    "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:*"
                }
            }
        }
    ]
}
```

Esempio: limitazione dell'accesso del progetto Amazon Pinpoint a indirizzi IP specifici

La seguente politica di esempio concede le autorizzazioni a qualsiasi utente per eseguire qualsiasi azione Amazon Pinpoint su un progetto specificato (). *projectId* La richiesta deve, tuttavia, avere origine dall'intervallo di indirizzi IP specificati nella condizione.

La condizione in questa dichiarazione identifica l'`54.240.143.*`intervallo di indirizzi Internet Protocol versione 4 (IPv4) consentiti, con una sola eccezione:. `54.240.143.188` Il `Condition` blocco utilizza le `NotIpAddress` condizioni `IpAddress` and e la chiave `aws:SourceIp` condition, che è una chiave di condizione AWS-wide. Per ulteriori informazioni su queste chiavi di condizione, consulta la sezione relativa alla specifica delle condizioni in una policy nella Guida per l'utente di IAM. I `aws:SourceIp` IPv4 valori utilizzano la notazione CIDR standard. Per ulteriori informazioni, consulta Operatori di condizione con indirizzo IP nella Guida per l'utente di 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"
                }
            }
        }
    ]
}
```

Esempio: limitazione dell'accesso di Amazon Pinpoint in base ai tag

La seguente politica di esempio concede le autorizzazioni per eseguire qualsiasi azione di Amazon Pinpoint su un progetto specificato (). *projectId* Tuttavia, le autorizzazioni vengono concesse solo se la richiesta proviene da un utente il cui nome è un valore nel tag `Owner` della risorsa per il progetto, come specificato nella condizione.

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

Esempio: autorizzazione all'invio di e-mail da parte di Amazon Pinpoint mediante identità verificate in Amazon SES

Quando verifichi un'identità e-mail, ad esempio un indirizzo e-mail o un dominio, tramite la console Amazon Pinpoint, tale identità viene configurata automaticamente in modo che possa essere

utilizzata sia da Amazon Pinpoint che da Amazon SES. Tuttavia, se verifichi un'identità e-mail tramite Amazon SES e desideri utilizzare tale identità con Amazon Pinpoint, devi applicare una policy a tale identità.

La seguente policy di esempio concede ad Amazon Pinpoint l'autorizzazione a inviare e-mail utilizzando un'identità e-mail verificata tramite 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/*"
                }
            }
        }
    ]
}
```

Se utilizzi Amazon Pinpoint nella regione AWS GovCloud (Stati Uniti occidentali), utilizza invece il seguente esempio di policy:

```
{
    "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/*"
                }
            }
        }
    ]
}
```

Ruoli IAM per attività comuni di Amazon Pinpoint

Un ruolo IAM è un'identità AWS Identity and Access Management (IAM) che puoi creare nel tuo AWS account e concedere autorizzazioni specifiche. Un ruolo IAM è un' AWS identità con politiche di autorizzazione che determinano ciò che l'identità può e non può fare. AWS Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque.

Inoltre, un ruolo non ha credenziali a lungo termine standard associate. Fornisce invece le credenziali di sicurezza temporanee per una sessione. Puoi utilizzare i ruoli IAM per delegare l'accesso a utenti, app, applicazioni o servizi che normalmente non hanno accesso alle tue AWS risorse.

Per questi motivi, puoi utilizzare i ruoli IAM per l'integrazione di Amazon Pinpoint con determinati servizi e risorse AWS associati al tuo account. Ad esempio, puoi consentire ad Amazon Pinpoint l'accesso alle definizioni di endpoint archiviate in un bucket Amazon Simple Storage Service (Amazon S3) e utilizzarle per i segmenti. Oppure potresti decidere di consentire ad Amazon Pinpoint lo streaming dei dati degli eventi in un flusso Amazon Kinesis per il tuo account. Allo stesso modo, potresti voler utilizzare i ruoli IAM per consentire alle app web o mobili di registrare endpoint o segnalare i dati di utilizzo per i progetti Amazon Pinpoint, senza AWS incorporare chiavi nelle app (dove possono essere difficili da ruotare e gli utenti possono potenzialmente estrarle).

Per questi scenari, è possibile delegare l'accesso ad Amazon Pinpoint utilizzando i ruoli IAM. Questa sezione illustra e fornisce esempi di attività Amazon Pinpoint comuni basate su ruoli IAM per utilizzare altri servizi AWS . Per informazioni sull'utilizzo dei ruoli IAM con app Web e per dispositivi mobili in modo più specifico, consulta Fornire l'accesso a utenti autenticati esternamente (federazione delle identità) nella Guida per l'utente di IAM.

Argomenti

- Ruolo IAM per l'importazione di endpoint o segmenti
- Ruolo IAM per l'esportazione di endpoint o segmenti
- Ruolo IAM per il recupero delle raccomandazioni da Amazon Personalize
- Ruolo IAM per lo streaming di eventi su Kinesis
- Ruolo IAM per l'invio di e-mail con Amazon SES

Ruolo IAM per l'importazione di endpoint o segmenti

Con Amazon Pinpoint, puoi definire un segmento di utenti importando le definizioni degli endpoint da un bucket Amazon Simple Storage Service (Amazon S3) nel tuo account. AWS Prima di eseguire l'importazione, è necessario delegare le autorizzazioni necessarie ad Amazon Pinpoint. A tale scopo, crei un ruolo AWS Identity and Access Management (IAM) e associa le seguenti politiche al ruolo:

- La policy gestita `AmazonS3ReadOnlyAccess` AWS . Questa policy è creata e gestita da AWS e garantisce l'accesso in sola lettura al tuo bucket Amazon S3.
- Una policy di attendibilità che consenta ad Amazon Pinpoint di assumere il ruolo.

Dopo aver creato il ruolo, puoi utilizzare Amazon Pinpoint per importare i segmenti da un bucket Amazon S3. Per informazioni sulla creazione di bucket e file di endpoint e sull'importazione di un segmento tramite la console, consulta Importazione di segmenti nella Guida per l'utente di Amazon Pinpoint. Per un esempio di come importare un segmento a livello di codice utilizzando il, consulta questa guida. AWS SDK per JavaImportazione di segmenti in Amazon Pinpoint

Creazione del ruolo IAM (AWS CLI)

Completa i seguenti passaggi per creare il ruolo IAM utilizzando AWS Command Line Interface ()AWS CLI. Se non hai installato il AWS CLI, consulta Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per creare il ruolo IAM utilizzando il AWS CLI

1. Crea un file JSON contenente la policy di attendibilità per il ruolo e salva il file in locale. È possibile utilizzare la seguente politica di attendibilità.

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

Nell'esempio precedente, procedere nel modo seguente:

- Sostituisci *region* con la AWS regione in cui utilizzi Amazon Pinpoint.
- *accountId*Sostituiscilo con l'ID univoco del tuo AWS account.
- Sostituisci *application-id* con l'ID univoco del progetto.

2. Usare il comando create-role per creare il ruolo e collegare la policy di attendibilità:

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document
 file://PinpointImportTrustPolicy.json
```

Dopo il prefisso `file://` specifica il percorso del file JSON che contiene la policy di attendibilità.

Dopo aver eseguito questo comando, nel terminale viene visualizzato un output simile al seguente:

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
```

```
                    "Action": "sts:AssumeRole",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "pinpoint.amazonaws.com"
                    },
                    "Condition": {
                        "StringEquals": {
                            "aws:SourceAccount": "accountId"
                         },
                         "ArnLike": {
                            "aws:SourceArn":
 "arn:aws:mobiletargeting:region:accountId:apps/application-id"
                         }
                    }
                }
            ]
        },
        "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
        "CreateDate": "2016-12-20T00:44:37.406Z",
        "RoleName": "PinpointSegmentImport",
        "Path": "/",
        "Arn": "arn:aws:iam::accountId:role/PinpointSegmentImport"
    }
}
```

3. Usa il attach-role-policycomando per allegare la politica `AmazonS3ReadOnlyAccess` AWS gestita al ruolo:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport
```

Ruolo IAM per l'esportazione di endpoint o segmenti

È possibile ottenere un elenco di endpoint creando un processo di esportazione. Quando crei un processo di esportazione, devi specificare un ID progetto e, facoltativamente, un ID segmento. Amazon Pinpoint esporta quindi un elenco degli endpoint associati al progetto o al segmento in un bucket Amazon Simple Storage Service (Amazon S3). Il file risultante contiene un elenco di endpoint in formato JSON con i relativi attributi, ad esempio canale, indirizzo, stato del consenso, data di creazione e ID degli endpoint.

Per creare un processo di esportazione, è necessario configurare un ruolo IAM che consenta ad Amazon Pinpoint di scrivere in un bucket Amazon S3. Il processo di configurazione del ruolo prevede due fasi:

1. Creazione di una policy IAM che consenta a un'entità (in questo caso, Amazon Pinpoint) di scrivere in un determinato bucket Amazon S3.
2. Creare un ruolo IAM e collegarvi la policy.

Questo argomento contiene le procedure per completare entrambe le fasi. Queste procedure presuppongono che sia già stato creato un bucket Amazon S3, con una cartella al suo interno, per archiviare i segmenti esportati. Per ulteriori informazioni sulla creazione di un bucket, consulta Creazione di un bucket nella Guida per l'utente di Amazon Simple Storage Service.

Queste procedure si basano anche sul presupposto che l' AWS Command Line Interface (AWS CLI) sia già stata installata e configurata. Per informazioni sulla configurazione di AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Passaggio 1: creare la policy IAM

Una policy IAM definisce le autorizzazioni per un'entità, ad esempio un'identità o una risorsa. Per creare un ruolo per l'esportazione di endpoint Amazon Pinpoint, devi creare una policy che conceda le autorizzazioni di scrittura per una determinata cartella in un determinato bucket Amazon S3. Il seguente esempio di policy segue la pratica di sicurezza che prevede la concessione del privilegio minimo, ovvero concede solo le autorizzazioni necessarie per eseguire una singola attività.

Co,e creare la policy IAM

1. In un editor di testo, crea un nuovo file. Incolla il codice seguente nel 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:::amzn-s3-demo-bucket-example-bucket" ],
            "Condition": {
                "StringEquals": {
                    "s3:delimiter": [ "/" ],
                    "s3:prefix": [
                        "",
                        "Exports/"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-bucket" ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "Exports/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
            "Action": [ "s3:*" ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-bucket/Exports/
*" ]
        }
    ]
}
```

Nel codice precedente, sostituisci tutte le istanze *amzn-s3-demo-bucket-example-bucket* con il nome del bucket Amazon S3 che contiene la cartella in cui desideri esportare le informazioni sul segmento. Inoltre, sostituisci tutte le istanze di *Exports* con il nome della cartella stessa.

Al termine, salva il file come `s3policy.json`.

2. Utilizzando AWS CLI, accedete alla directory in cui si trova il `s3policy.json` file. Immetti il comando seguente per creare la policy:

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
 file://s3policy.json
```

Se la creazione della policy è andata a buon fine, verrà visualizzato un output simile al seguente:

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

Copia l'Amazon Resource Name (ARN) della policy (`arn:aws:iam::123456789012:policy/s3ExportPolicy` nell'esempio precedente). Nella sezione successiva dovrai specificare questo ARN durante la creazione del ruolo.

> **Note**
>
> Se viene visualizzato un messaggio in cui è indicato che l'account non è autorizzato a eseguire l'operazione `CreatePolicy`, all'account utente è necessario collegare una policy che consenta di creare nuovi ruoli e nuove policy IAM. Per ulteriori informazioni,

consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente di IAM .

Passaggio 2: creare il ruolo IAM

Una volta creata una policy IAM, è possibile creare un ruolo e collegare la policy a tale ruolo. Ogni ruolo IAM contiene una policy di attendibilità, ovvero un set di regole che specifica le entità che possono assumere il ruolo specifico. In questa sezione viene creata una policy di attendibilità che consente ad Amazon Pinpoint di assumere il ruolo. In seguito si creerà il ruolo stesso e vi si collegherà la policy creata nella sezione precedente.

Per creare il ruolo IAM

1. In un editor di testo, crea un nuovo file. Incolla il codice seguente nel 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"
                }
            }
        }
    ]
}
```

   Salva il file con nome `trustpolicy.json`.

2. Utilizzando AWS CLI, accedete alla directory in cui si trova il `trustpolicy.json` file. Immetti il comando seguente per creare un nuovo ruolo:

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
 file://trustpolicy.json
```

3. Nella riga di comando immetti il comando seguente per collegare la policy creata nella sezione precedente al ruolo che hai appena creato:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
s3ExportPolicy --role-name s3ExportRole
```

Nel comando precedente, sostituite *arn:aws:iam::123456789012:policy/s3ExportPolicy* con l'ARN della policy creata nella sezione precedente.

Ruolo IAM per il recupero delle raccomandazioni da Amazon Personalize

Puoi configurare Amazon Pinpoint per recuperare i dati delle raccomandazioni da una soluzione Amazon Personalize implementata come campagna Amazon Personalize. Puoi utilizzare questi dati per inviare suggerimenti personalizzati ai destinatari dei messaggi in base agli attributi e al comportamento di ciascun destinatario. Per ulteriori informazioni, consulta Modelli di machine learning nella Guida per l'utente di Amazon Pinpoint.

Prima di poter recuperare i dati delle raccomandazioni da una campagna Amazon Personalize, devi creare un ruolo AWS Identity and Access Management (IAM) che consenta ad Amazon Pinpoint di recuperare i dati dalla campagna. Amazon Pinpoint può creare questo ruolo automaticamente quando utilizzi la console per configurare un modello di raccomandazione in Amazon Pinpoint. In alternativa, è possibile creare questo ruolo manualmente.

Per creare manualmente il ruolo, utilizza l'API IAM per completare le fasi seguenti:

1. Creare una policy IAM che consenta a un'entità (in questo caso, Amazon Pinpoint) di recuperare i dati delle raccomandazioni da una campagna Amazon Personalize.
2. Creare un ruolo IAM e collegarvi la policy IAM.

Questo argomento spiega come completare questi passaggi utilizzando AWS Command Line Interface ()AWS CLI. Si presuppone che la soluzione Amazon Personalize sia già stata creata e implementata come campagna Amazon Personalize. Per informazioni sulla creazione e sull'implementazione di una campagna, consulta l'argomento relativo alla creazione di una campagna nella Guida per gli sviluppatori di Amazon Personalize.

Questo argomento si basa sul presupposto che la AWS CLI sia già stata installata e configurata. Per informazioni sulla configurazione di AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Passaggio 1: creare la policy IAM

Una policy IAM definisce le autorizzazioni per un'entità, ad esempio un'identità o una risorsa. Per creare un ruolo che consenta ad Amazon Pinpoint di recuperare i dati delle raccomandazioni da una campagna Amazon Personalize, devi innanzitutto creare una policy IAM per il ruolo. Questa policy deve consentire ad Amazon Pinpoint di:

- Recuperare le informazioni di configurazione per la soluzione distribuita dalla campagna (`DescribeSolution`).
- Controllare lo stato della campagna (`DescribeCampaign`).
- Recuperare i dati dei suggerimenti dalla campagna (`GetRecommendations`).

Nella procedura seguente, la policy di esempio consente questo accesso per una specifica soluzione Amazon Personalize che è stata implementata da una specifica campagna Amazon Personalize.

Co,e creare la policy IAM

1. In un editor di testo, crea un nuovo file. Incolla il codice seguente nel 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"
            ]
        }
    ]
```

```
}
```

Nell'esempio precedente, sostituisci il *italicized* testo con le tue informazioni:

- *region*— Il nome della AWS regione che ospita la soluzione e la campagna Amazon Personalize.
- *accountId*— Il tuo Account AWS ID.
- *solutionId*— L'ID di risorsa univoco per la soluzione Amazon Personalize implementata dalla campagna.
- *campaignId*— L'ID di risorsa univoco per la campagna Amazon Personalize da cui recuperare i dati delle raccomandazioni.

2. Al termine, salva il file come `RetrieveRecommendationsPolicy.json`.
3. Utilizzando l'interfaccia a riga di comando, passa alla directory in cui è stato salvato il file `RetrieveRecommendationsPolicy.json`.
4. Immetti il comando seguente per creare una policy e denominarla `RetrieveRecommendationsPolicy`. Per usare un nome diverso, scegli *RetrieveRecommendationsPolicy* il nome che desideri.

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document
 file://RetrieveRecommendationsPolicy.json
```

> **Note**
>
> Se ricevi un messaggio in cui è indicato che l'account non è autorizzato a eseguire l'operazione `CreatePolicy`, all'account utente devi collegare una policy che consenta di creare nuovi ruoli e nuove policy IAM per l'account. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente di IAM .

5. Copia l'Amazon Resource Name (ARN) della policy (`arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy` nell'esempio precedente). Nella sezione successiva, avrai bisogno di questo ARN per creare il ruolo IAM.

Passaggio 2: creare il ruolo IAM

Una volta creata la policy IAM, puoi creare un ruolo IAM a cui collegarla.

Ogni ruolo IAM contiene una policy di attendibilità, che è composta da un set di regole che specifica le entità che possono assumere tale ruolo. In questa sezione viene creata una policy di attendibilità che consente ad Amazon Pinpoint di assumere il ruolo. Successivamente, crei il ruolo vero e proprio. Quindi, colleghi tale policy al ruolo.

Per creare il ruolo IAM

1. In un editor di testo, crea un nuovo file. Incolla il codice seguente nel 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. Salva il file con nome `RecommendationsTrustPolicy.json`.
3. Utilizzando l'interfaccia a riga di comando, passa alla directory in cui è stato salvato il file `RecommendationsTrustPolicy.json`.
4. Immetti il comando seguente per creare un nuovo ruolo e denominarlo `PinpointRoleforPersonalize`. Per usare un nome diverso, *`PinpointRoleforPersonalize`* scegli il nome che desideri.

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-
document file://RecommendationsTrustPolicy.json
```

5. Immetti il comando seguente per collegare la policy creata nella sezione precedente al ruolo che hai appena creato:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

   Nel comando precedente, sostituite *arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy* con l'ARN della policy creata nella sezione precedente. Inoltre, sostituiscilo *PinpointRoleforPersonalize* con il nome del ruolo specificato nel passaggio 4, se hai specificato un nome diverso per il ruolo.

Ruolo IAM per lo streaming di eventi su Kinesis

Amazon Pinpoint può inviare automaticamente i dati sull'utilizzo delle app, o dati sugli eventi, dalla tua app a un flusso di dati Amazon Kinesis o a un flusso di distribuzione Amazon Data Firehose nel tuo account. AWS Prima che Amazon Pinpoint possa avviare lo streaming dei dati degli eventi, devi delegare le autorizzazioni necessarie ad Amazon Pinpoint.

Se usi la console per configurare lo streaming degli eventi, Amazon Pinpoint crea automaticamente un ruolo AWS Identity and Access Management (IAM) con le autorizzazioni necessarie. Per ulteriori informazioni, consulta Streaming di eventi Amazon Pinpoint su Kinesis nella Amazon Pinpoint User Guide.

Se vuoi creare manualmente il ruolo, collega le policy seguenti al ruolo:

- Una policy di autorizzazione che consenta ad Amazon Pinpoint di inviare dati di eventi al flusso.
- Una policy di attendibilità che consenta ad Amazon Pinpoint di assumere il ruolo.

Dopo aver creato il ruolo, è possibile configurare Amazon Pinpoint per l'invio automatico degli eventi al flusso. Per ulteriori informazioni sul tagging, consulta Trasmetti in streaming i dati degli eventi delle app tramite Kinesis e Firehose utilizzando Amazon Pinpointin questa guida.

Creazione del ruolo IAM (AWS CLI)

Completa la procedura seguente per creare manualmente un ruolo IAM mediante l' AWS Command Line Interface (AWS CLI). Per informazioni su come creare il ruolo utilizzando la console Amazon Pinpoint, consulta Streaming di eventi Amazon Pinpoint su Kinesis nella Guida per l'utente di Amazon Pinpoint.

Se non hai installato il AWS CLI, consulta Installazione di AWS CLI nella Guida per l'utente.AWS Command Line Interface È inoltre necessario aver creato uno stream Kinesis o uno stream Firehose. Per informazioni sulla creazione di queste risorse, consulta Creating and Managing Streams nella Amazon Kinesis Data Streams Developer Guide o Creazione di un flusso di distribuzione Amazon Data Firehose nella Amazon Data Firehose Developer Guide.

Per creare il ruolo IAM utilizzando il AWS CLI

1. Crea un nuovo file. Incolla la seguente politica nel documento e apporta le seguenti modifiche:

   - Sostituisci *region* con la AWS regione in cui utilizzi Amazon Pinpoint.
   - *accountId*Sostituiscilo con l'ID univoco del tuo AWS account.
   - Sostituisci *applicationId* con l'ID univoco del progetto.

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

   Al termine, salva il file come `PinpointEventStreamTrustPolicy.json`.

2. Usa il comando `create-role` per creare il ruolo e collegare la policy di attendibilità:

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-
document file://PinpointEventStreamTrustPolicy.json
```

3. Crea un nuovo file contenente la policy di autorizzazione per il ruolo.

   Se stai configurando Amazon Pinpoint per inviare dati a un flusso Kinesis, incolla la seguente policy nel file e sostituisci la seguente:

   - Sostituisci *region* con la AWS regione in cui utilizzi Amazon Pinpoint.
   - *accountId*Sostituiscilo con l'ID univoco del tuo AWS account.
   - Sostituiscilo *streamName* con il nome del tuo stream Kinesis.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Action": [
            "kinesis:PutRecords",
            "kinesis:DescribeStream"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:kinesis:region:accountId:stream/streamName"
        ]
    }
}
```

   In alternativa, se stai configurando Amazon Pinpoint per inviare dati a uno stream Firehose, incolla la seguente policy nel file e sostituisci la seguente:

   - Sostituisci *region* con la AWS regione in cui utilizzi Amazon Pinpoint.
   - *accountId*Sostituiscilo con l'ID univoco del tuo AWS account.
   - Sostituisci *delivery-stream-name* con il nome del tuo stream Firehose.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
```

```
          "firehose:PutRecordBatch",
          "firehose:DescribeDeliveryStream"
        ],
        "Resource": [
          "arn:aws:firehose:region:accountId:deliverystream/delivery-stream-name"
     ]
    }
}
```

Al termine, salva il file come `PinpointEventStreamPermissionsPolicy.json`.

4. Utilizza il comando `put-role-policy` per collegare la policy di autorizzazione al ruolo:

```
aws iam put-role-policy --role-name PinpointEventStreamRole --policy-
name PinpointEventStreamPermissionsPolicy --policy-document file://
PinpointEventStreamPermissionsPolicy.json
```

Ruolo IAM per l'invio di e-mail con Amazon SES

Amazon Pinpoint utilizza le tue risorse Amazon SES per inviare e-mail relative alla tua campagna o al tuo percorso. Prima che Amazon Pinpoint possa utilizzare le tue risorse Amazon SES per inviare e-mail, devi concedere le autorizzazioni necessarie ad Amazon Pinpoint. Il tuo account deve disporre delle autorizzazioni `iam:PutRolePolicy` e dei `iam:UpdateAssumeRolePolicy` permessi per aggiornare o creare ruoli IAM.

La console Amazon Pinpoint può creare automaticamente un ruolo AWS Identity and Access Management (IAM) con le autorizzazioni richieste. Per ulteriori informazioni, consulta Creazione di un ruolo di invio per l'orchestrazione delle e-mail nella Guida per l'utente di Amazon Pinpoint.

Se vuoi creare manualmente il ruolo, collega le policy seguenti al ruolo:

- Una politica di autorizzazioni che concede ad Amazon Pinpoint l'accesso alle tue risorse Amazon SES.
- Una policy di attendibilità che consenta ad Amazon Pinpoint di assumere il ruolo.

Dopo aver creato il ruolo, puoi configurare Amazon Pinpoint per utilizzare le tue risorse Amazon SES.

Puoi testare le politiche IAM con il simulatore di policy IAM. Per ulteriori informazioni, consulta Testare le policy IAM con il simulatore di policy IAM nella IAM User Guide.

Creazione del ruolo IAM (AWS Management Console)

Completa i seguenti passaggi per creare manualmente un ruolo IAM per la tua campagna o il tuo percorso di invio di e-mail.

1. Crea una nuova policy di autorizzazione seguendo le istruzioni riportate in Creazione di policy using the JSON editor nella IAM User Guide.

   - Nella fase 5, utilizza la seguente politica di autorizzazione per il ruolo IAM.

     - Sostituisci *partition* con la partizione in cui si trova la risorsa. Per lo standard Regioni AWS, la partizione è. `aws` Se sono presenti risorse in altre partizioni, la partizione è `aws-partitionname`. Ad esempio, la partizione per le risorse negli AWS GovCloud Stati Uniti occidentali è. `aws-us-gov`
     - Sostituisci *region* con il nome di Regione AWS chi ospita il progetto Amazon Pinpoint.
     - *accountId*Sostituiscilo con l'ID univoco del tuo Account 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. Crea una nuova policy di fiducia seguendo le istruzioni riportate nella sezione Creazione di un ruolo utilizzando policy di fiducia personalizzate nella IAM User Guide.

   a. Nel passaggio 4, utilizza la seguente politica di fiducia.

      - Sostituiscilo *accountId* con l'ID univoco del tuo Account AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPinpoint",
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

b. Nel passaggio 11, aggiungi la politica di autorizzazione creata nel passaggio precedente.

Risoluzione dei problemi relativi alla gestione di identità e accessi di Amazon Pinpoint

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon Pinpoint e IAM.

Argomenti

- Non sono autorizzato a eseguire un'azione in Amazon Pinpoint
- Non sono autorizzato a eseguire iam: PassRole
- Desidero consentire a persone esterne al mio AWS account di accedere alle mie risorse Amazon Pinpoint

Non sono autorizzato a eseguire un'azione in Amazon Pinpoint

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` cerca di utilizzare la console per visualizzare i dettagli relativi a un progetto ma non dispone di autorizzazioni `mobiletargeting:GetApp`.

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

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa *my-example-project* utilizzando l'azione `mobiletargeting:GetApp`.

Non sono autorizzato a eseguire iam: PassRole

Se viene visualizzato un errore indicante che non disponi dell'autorizzazione per eseguire l'azione `iam:PassRole`, per poter passare un ruolo ad Amazon Pinpoint dovrai aggiornare le policy.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'azione in Amazon Pinpoint. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

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

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Desidero consentire a persone esterne al mio AWS account di accedere alle mie risorse Amazon Pinpoint

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per consentire alle persone di accedere alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon Pinpoint supporta queste funzionalità, consulta Come Amazon Pinpoint funziona con IAM.
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Registrazione di log e monitoraggio in Amazon Pinpoint

La registrazione di log e il monitoraggio sono importanti per garantire l'affidabilità, la disponibilità e le prestazioni dei progetti Amazon Pinpoint e di altri tipi di risorse Amazon Pinpoint. È necessario registrare e raccogliere dati di monitoraggio da tutte le parti dei progetti e delle risorse Amazon Pinpoint per eseguire più facilmente il debug di un errore multipunto, se si verifica. AWS fornisce diversi strumenti che possono aiutarti a registrare e raccogliere questi dati e a rispondere a potenziali incidenti:

AWS CloudTrail

Amazon Pinpoint si integra con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese in Amazon Pinpoint da un utente, un ruolo o un altro servizio. AWS incluse

le azioni dalla console Amazon Pinpoint e le chiamate programmatiche alle operazioni delle API Amazon Pinpoint. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare quali richieste sono state fatte ad Amazon Pinpoint. Per ogni richiesta, è possibile identificare quando è stata effettuata, l'indirizzo IP da cui è stata effettuata, chi l'ha effettuata e ulteriori dettagli. Per ulteriori informazioni sul tagging, consulta Registra le chiamate API Amazon Pinpoint con AWS CloudTrailin questa guida.

Amazon CloudWatch

Puoi utilizzare Amazon CloudWatch per raccogliere, visualizzare e analizzare diverse metriche importanti relative al tuo account Amazon Pinpoint e ai tuoi progetti. Puoi anche utilizzarlo CloudWatch per creare allarmi che ti avvisano se il valore di una metrica soddisfa determinate condizioni e rientra o supera una soglia da te definita. Se crei un allarme, CloudWatch invia una notifica a un argomento di Amazon Simple Notification Service (Amazon SNS) da te specificato. Per ulteriori informazioni, consulta Monitoring Amazon Pinpoint with amazon CloudWatch nella Amazon Pinpoint User Guide.

AWS Health Pannelli di controllo

Utilizzando i AWS Health dashboard, puoi controllare e monitorare lo stato del tuo ambiente Amazon Pinpoint. Per verificare lo stato generale del servizio Amazon Pinpoint, utilizza il AWS Service Health Dashboard. Per controllare, monitorare e visualizzare i dati storici su eventuali eventi o problemi che potrebbero influire in modo più specifico sull' AWS ambiente, utilizza la AWS Personal Health Dashboard. Per ulteriori informazioni su questi dashboard, consulta la Guida dell'utente di AWS Health.

AWS Trusted Advisor

AWS Trusted Advisor ispeziona l' AWS ambiente e fornisce consigli sulle opportunità per colmare le lacune di sicurezza, migliorare la disponibilità e le prestazioni del sistema e risparmiare denaro. Tutti i AWS clienti hanno accesso a una serie di controlli di Trusted Advisor base. I clienti che dispongono di un piano di supporto Business o Enterprise hanno accesso a Trusted Advisor controlli aggiuntivi.

Molti di questi controlli possono aiutarti a valutare il livello di sicurezza delle tue risorse Amazon Pinpoint nell'ambito generale del AWS tuo account. Ad esempio, il set principale di controlli Trusted Advisor include quanto segue:

- Configurazioni di registrazione per il tuo AWS account, per ogni regione supportata. AWS
- Autorizzazioni di accesso per i bucket Amazon Simple Storage Service (Amazon S3) che potrebbero contenere file importati in Amazon Pinpoint per creare i segmenti.

- Utilizzo di AWS Identity and Access Management utenti, gruppi e ruoli per controllare l'accesso alle risorse di Amazon Pinpoint.
- Configurazioni IAM e impostazioni delle policy che potrebbero compromettere la sicurezza del tuo AWS ambiente e delle risorse Amazon Pinpoint.

Per ulteriori informazioni, consulta AWS Trusted Advisor nella Guida per l'utente di Supporto .

Convalida della conformità per Amazon Pinpoint

Revisori di terze parti valutano la sicurezza e la conformità di Amazon Pinpoint nell'ambito di più programmi di conformità AWS . Questi includono AWS System and Organization Controls (SOC), FedRAMP, HIPAA, 27001:2013 per i controlli di gestione della sicurezza, 27017:2015 per i controlli ISO/IEC specifici del cloud ISO/IEC , 27018:2014 per la protezione dei dati personali ISO/IEC , 9001:2015 per i sistemi di gestione della qualità e altri. ISO/IEC

https://aws.amazon.com/compliance/services-in-scope/ Per informazioni generali, vedere programmi di AWS conformità, programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact Scaricamento dei . AWS

La tua responsabilità di conformità quando utilizzi Amazon Pinpoint è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide introduttive su su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper descrive in che modo le aziende possono utilizzare per creare applicazioni conformi all'HIPAA. AWS
- AWS risorse per la conformità e risorse per la conformità: questa raccolta di riguardare il settore e la località in cui operate.
- Valutazione delle risorse in base alle regole contenute nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.

- AWS Security Hub— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente, AWS che consente di verificare la conformità agli standard e alle best practice del settore della sicurezza.

Amazon Pinpoint è un servizio idoneo alla AWS normativa HIPAA quando i clienti utilizzano i canali di comunicazione appropriati. Per utilizzare Amazon Pinpoint per eseguire carichi di lavoro contenenti informazioni sanitarie protette (PHI) come definito dall'HIPAA e dalla legislazione e dalle normative associate, è necessario utilizzare il canale e-mail, il canale di notifica push o il canale SMS per inviare messaggi contenenti informazioni sanitarie protette. Se utilizzi il canale SMS per inviare messaggi che contengono PHI, devi inviare tali messaggi utilizzando un codice breve dedicato che hai richiesto per il tuo AWS account allo scopo esplicito di inviare messaggi che contengano o possano contenere PHI. Il canale vocale non è idoneo alla AWS normativa HIPAA; non utilizzare il canale vocale per inviare messaggi contenenti PHI.

Resilienza in Amazon Pinpoint

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni sulle architetture di riferimento, consulta il manuale relativo all'architettura resiliente di Amazon Pinpoint.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta infrastruttura globale.AWS

Sicurezza dell'infrastruttura in Amazon Pinpoint

In quanto servizio gestito, Amazon Pinpoint è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi di AWS sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon Pinpoint attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Anche se è possibile eseguire queste chiamate API da qualsiasi percorso di rete, Amazon Pinpoint supporta le policy di accesso basate su risorse. Queste policy possono includere restrizioni basate sull'indirizzo IP di origine. Per ulteriori informazioni su questo tipo di policy, consulta Gestione dell'accesso con policy.

Inoltre, puoi configurare e utilizzare varie funzionalità di AWS sicurezza per controllare l'accesso alle risorse di Amazon Pinpoint da qualsiasi app mobile o Web che integri con Amazon Pinpoint. Sono incluse le restrizioni sulle chiamate API per attività quali l'aggiunta di endpoint, l'aggiornamento dei dati degli endpoint, l'invio dei dati degli eventi e la creazione di report sui dati di utilizzo.

Per utilizzare queste funzionalità, ti consigliamo di utilizzare AWS Mobile SDKs o le AWS Amplify JavaScript librerie per integrare app mobili e Web con Amazon Pinpoint. Per le app Android o iOS, ti consigliamo di utilizzare rispettivamente il AWS Mobile SDK for Android o AWS Mobile SDK for iOS il. Per le app mobili o web JavaScript basate su dispositivi mobili, ti consigliamo di utilizzare la AWS Amplify JavaScript Library for the Web o la AWS Amplify JavaScript Library for React Native. Per ulteriori informazioni su queste risorse, consulta Guida introduttiva ai AWS dispositivi mobili SDKs, Guida introduttiva alla libreria AWS Amplify per il Web e Guida introduttiva alla libreria AWS Amplify per React Native.

Analisi della configurazione e delle vulnerabilità in Amazon Pinpoint

In quanto servizio gestito, Amazon Pinpoint è protetto dalle procedure di sicurezza di rete AWS globali descritte nel white paper Amazon Web Services: panoramica dei processi di sicurezza. Ciò significa che AWS gestisce ed esegue attività e procedure di sicurezza di base per rafforzare, applicare patch, aggiornare e mantenere in altro modo l'infrastruttura sottostante per il tuo account e le tue risorse Amazon Pinpoint. Queste procedure sono state riviste e certificate dalle terze parti appropriate.

Per ulteriori informazioni, consulta le seguenti risorse:

- Convalida della conformità per Amazon Pinpoint
- Modello di responsabilità condivisa
- Amazon Web Services: panoramica dei processi di sicurezza (whitepaper)

Best practice di sicurezza per Amazon Pinpoint

Utilizza gli account AWS Identity and Access Management (IAM) per controllare l'accesso Amazon Pinpoint alle operazioni API, in particolare alle operazioni che creano, modificano o Amazon Pinpoint eliminano risorse. Per quanto riguarda l' Amazon Pinpoint API, tali risorse includono progetti, campagne e viaggi. Per l'API SMS e Voce di Amazon Pinpoint , tali risorse includono numeri di telefono, pool e set di configurazione.

- Crea un utente individuale per ogni persona che gestisce Amazon Pinpoint le risorse, incluso te stesso. Non utilizzare credenziali AWS root per gestire le risorse di Amazon Pinpoint.
- Assegna a ciascun utente un set minimo di autorizzazioni richieste per eseguire le proprie mansioni.
- Utilizza gruppi IAM per gestire in modo efficace le autorizzazioni per più utenti.
- Ruota periodicamente le credenziali IAM.

Per ulteriori informazioni sulla Amazon Pinpoint sicurezza, consulta la sezione Sicurezza in Amazon Pinpoint. Per ulteriori informazioni su IAM, consulta AWS Identity and Access Management. Per informazioni sulle best practice di IAM, consulta Best practice di IAM.

Quote di Amazon Pinpoint

Le sezioni seguenti elencano e descrivono le quote, precedentemente definite limiti, applicabili alle risorse e alle operazioni di Amazon Pinpoint. Alcune quote possono essere aumentate, mentre altre no. Per determinare se è possibile richiedere un aumento per una quota, consulta la colonna Possibilità di incremento in ogni sezione.

Argomenti

- Quote per i progetti
- Quote di richiesta API
- Quote per le campagne
- Quote per l'e-mail
- Quote per gli endpoint
- Quote per l'importazione degli endpoint
- Quote per l'inserimento di eventi
- Quote per i viaggi
- Quote di Lambda
- Quote di machine learning
- Quote per i modelli di messaggio
- Quote per le notifiche push
- Quote per i messaggi in-app
- Quote per i segmenti
- Quote per SMS
- Quote 10DLC
- Quote per i messaggi vocali
- Richiesta di aumento delle quote

Quote per i progetti

Nella tabella seguente sono elencate le quote relative ai progetti in Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Progetti	In ognuno Regione AWS puoi avere fino a 100 progetti.	No

Quote di richiesta API

Amazon Pinpoint implementa quote che limitano la dimensione e il numero di richieste che puoi inviare all'API Amazon Pinpoint dal tuo account. AWS

La dimensione massima di un payload di invocazione (richiesta e risposta) è di 7 MB, se non diversamente specificato per un particolare tipo di risorsa. Per determinare se una risorsa dispone di una quota diversa, vedere la sezione appropriata di questo argomento per quel tipo di risorsa.

Il numero massimo di richieste varia in base al tipo di quota e all'operazione dell'API. Amazon Pinpoint implementa due tipi di quote per le richieste API:

- Quote di velocità: definite anche limiti di frequenza, questo tipo di quote definisce il numero massimo di richieste che è possibile effettuare al secondo per una determinata operazione. Controlla la frequenza di invio o ricezione delle richieste per account.
- Quote di espansione: note anche come limiti di espansione o capacità di espansione, questo tipo di quota definisce il numero massimo di richieste simultaneamente in corso per un account.

Nella tabella seguente sono elencate le quote tariffarie e di espansione per l'API Amazon Pinpoint.

Operazione	Quota di velocità/ espansione predefinita (richieste al secondo)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10
CreateImportJob	300
CreatePushTemplate	10

Operazione	Quota di velocità/ espansione predefinita (richieste al secondo)
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
UpdateVoiceTemplate	10

Operazione	Quota di velocità/ espansione predefinita (richieste al secondo)
Tutte le altre operazioni	300

Nella tabella seguente sono riportate elenca le quote per l'importazione di file per `CreateImportJob`.

Operazione	Quota predefinita	Idoneità all'incremento
Numero massimo di file di importazione	10.000 file per processo di importazione	No

Se si supera una di queste quote, Amazon Pinpoint applica la limitazione della larghezza di banda della rete alla richiesta, ovvero rifiuta una richiesta altrimenti valida e restituisce un errore `TooManyRequests`. La limitazione della larghezza di banda della rete viene calcolata in base al numero totale di richieste effettuate dall'account per un'operazione specifica in una Regione AWS data. Inoltre, le decisioni di throttling vengono calcolate in modo indipendente per ogni operazione. Ad esempio, se Amazon Pinpoint applica la limitazione della larghezza di banda della rete a una richiesta per l'operazione `SendMessages`, una richiesta simultanea per l'operazione `UpdateEndpoint` può essere completata correttamente.

Quote per le campagne

Le quote seguenti si applicano alla risorsa Campagne dell'API Amazon Pinpoint.

Le seguenti quote si applicano singolarmente Regione AWS e alcune possono essere aumentate. Per ulteriori informazioni, consulta Richiesta di un aumento delle quote nella Guida per l'utente di Service Quotas.

Risorsa	Quota predefinita	Idoneità all'incremento
Campagne attive	200 per account	No

Risorsa	Quota predefinita	Idoneità all'incremento
	Note Una campagna attiva è una campagna che non è stata completata o non è andata a buon fine. Lo stato delle campagne attive può essere `SCHEDULED`, `EXECUTING` o `PENDING_NEXT_RUN`.	
Dimensione massima del segmento	Per i segmenti importati: 100.000.000 per campagna Per segmenti dinamici: illimitato	No

Risorsa	Quota predefinita	Idoneità all'incremento
Campagne basate su eventi	Ogni progetto può includere fino a 25 campagne che vengono inviate quando si verificano determinati eventi. Le campagne che utilizzan o trigger basati su eventi devono utilizzare segmenti dinamici. Non possono utilizzare segmenti importati. Se integri la tua app con Amazon Pinpoint utilizzando un SDK per AWS dispositi vi mobili, i messaggi delle campagne basate sugli eventi vengono inviati solo ai clienti le cui app eseguono la AWS Mobile SDK for Android versione 2.7.2 o successiva oppure la versione 2.6.30 o successiva. AWS Mobile SDK for iOS Se non riesce a recapitar e un messaggio da una campagna basata su eventi entro cinque minuti, Amazon Pinpoint elimina il messaggio e non esegue altri tentativi di consegna.	No

Quote per l'e-mail

Le quote nelle sezioni seguenti si applicano al canale e-mail.

Quote per i messaggi e-mail

Risorsa	Quota predefinita	Idoneità all'incremento
Dimensione massima dei messaggi, compresi gli allegati	10 MB per messaggio	No
Numero di identità verificate	10.000 identità **Note** Per identità si intendono indirizzi e-mail o domini o una qualsiasi combinazione di questi. Ogni messaggio e-mail inviato tramite Amazon Pinpoint deve essere inviato da un'identità verificata.	No

Quote per mittenti e destinatari di e-mail

Risorsa	Quota predefinita	Idoneità all'incremento
Indirizzo del mittente	Tutti gli indirizzi o domini di invio devono essere verificati.	No
Indirizzo del destinatario	Se l'account si trova nella sandbox, tutti gli indirizzi e-mail o domini dei destinatari devono essere verificati.	Sì

Risorsa	Quota predefinita	Idoneità all'incremento
	Se l'account è esterno alla sandbox, puoi inviare a qualsiasi indirizzo valido.	
Numero di destinatari per messaggio	50 destinatari per messaggio	No
Numero di identità che è possibile verificare	10.000 identità per regione AWS **Note** Per identità si intendono indirizzi e-mail o domini o una qualsiasi combinazione di questi. Ogni messaggio e-mail inviato tramite Amazon Pinpoint deve essere inviato da un'identità verificata.	No

Quote di invio e-mail

La quota di invio, la velocità di invio e i limiti dell'ambiente di sperimentazione (sandbox) sono condivisi tra i due servizi nella stessa regione. Se utilizzi Amazon SES in us-east-1, sei stato rimosso dall'ambiente di sperimentazione (sandbox) e hai subito un aumento della quota/tariffa di invio, tutte queste modifiche si applicano al tuo account Pinpoint in us-east-1.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero di e-mail che è possibile inviare in un periodo	Se l'account si trova nella sandbox, 200 e-mail nell'arco di 24 ore.	Sì

Risorsa	Quota predefinita	Idoneità all'incremento
di tempo di 24 ore (quota di invio)	Se l'account è esterno alla sandbox, la quota varia in base al caso d'uso specifico. **Note** Questa quota si basa sul numero di destinatari, anziché sul numero di messaggi univoci inviati. Per destinatario si intende qualsiasi indirizzo e-mail presente nella riga A:.	
Numero di e-mail che è possibile inviare al secondo (frequenza di invio)	Se l'account si trova nella sandbox, 1 e-mail al secondo. Se l'account è esterno alla sandbox, la frequenza varia in base al caso d'uso specifico. **Note** Questa frequenza si basa sul numero di destinatari, anziché sul numero di messaggi univoci inviati. Per destinatario si intende qualsiasi indirizzo e-mail presente nella riga A:.	Sì

Quote per gli endpoint

Le quote seguenti si applicano alla risorsa Endpoint dell'API Amazon Pinpoint.

Il numero massimo di attributi supportati per endpoint è 250 e la dimensione massima dell'endpoint è 15 KB. Tuttavia, è possibile che questo numero di attributi sia limitato dalla dimensione totale di un endpoint, che include tutti gli attributi. Se riscontri errori durante l'aggiunta di attributi al modello, valuta la possibilità di ridurre la quantità di dati in ogni attributo o di diminuire il numero di attributi.

Risorsa	Quota predefinita	Idoneità all'incremento
Dimensioni dell'endpoint	Dimensione massima 15 KB	No
Attributi assegnati collettivamente ai parametri `Attributes` , `Metrics` e `UserAttributes`	250 per tutti i parametri degli attributi per applicazione	No
Attributi assegnati al parametro `Attributes`	250 per tutti i parametri degli attributi per applicazione	No
Attributi assegnati al parametro `Metrics`	250 per tutti i parametri degli attributi per applicazione	No
Attributi assegnati al parametro `UserAttributes`	250 per tutti i parametri degli attributi per applicazione	No
Lunghezza del nome degli attributi	50 caratteri	No
Lunghezza del valore degli attributi	100 caratteri	No
Oggetti `EndpointBatchItem` in un payload `EndpointBatchRequest`	100 per payload. La dimensione del payload non può essere superiore a 7 MB.	No

Risorsa	Quota predefinita	Idoneità all'incremento
Endpoint con lo stesso ID utente	15 endpoint univoci per ID utente	No
Valori assegnati agli attributi del parametro `Attributes`	50 per attributo	No
Valori assegnati agli attributi del parametro `UserAttributes`	50 per attributo	No

Quote per l'importazione degli endpoint

Le quote seguenti si applicano all'importazione di endpoint in Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Processi di importazione attivi	10 per account I processi di importazione vengono conteggiati rispetto a questa quota solo se sono in esecuzione. Una volta completato, il processo di importazione non viene più conteggiato rispetto a questa quota.	No
Dimensione di importazione	1 GB per processo di importazione Ad esempio, se ogni endpoint è da 4 KB o meno, è possibile importare 250.000 endpoint.	No

Quote per l'inserimento di eventi

Le seguenti quote si applicano all'inserimento di eventi utilizzando la risorsa AWS Mobile SDKs ed Events dell'API Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di tipi di evento personalizzati	1.500 per app	No
Numero massimo di chiavi di attributi personalizzati	500 per app	No
Numero massimo di valori di attributi personalizzati per chiave di attributo	100.000. Qualsiasi numero superiore a 100.000 è ancora registrato, ma non sarà disponibile nella console di analisi Amazon Pinpoint.	No
Numero massimo di caratteri per chiave di attributo	50	No
Numero massimo di caratteri per valore di attributo	200. Se il numero di caratteri supera i 200, l'evento viene eliminato.	No
Numero massimo di chiavi di parametri personalizzati	500 per app	No
Il numero massimo di eventi in una richiesta	100 per richiesta	No
Dimensione massima di una richiesta	4 MB	No
Dimensione massima di un singolo evento	1.000 KB	No

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di chiavi di attributo e chiavi di parametro per ogni evento	40 per richiesta	No

Quote per i viaggi

Le quote seguenti si applicano ai viaggi.

Le seguenti quote si applicano singolarmente Regione AWS e alcune possono essere aumentate. Per ulteriori informazioni, consulta Richiesta di un aumento delle quote nella Guida per l'utente di Service Quotas.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di viaggi attivi	50 per account	No
Numero massimo di attivi EventTriggeredJourneys	20 per account	No
Numero massimo di attività di viaggio	40 per viaggio	No
Dimensione massima del segmento	Per i segmenti importati: 100.000.000 GB per percorso Per segmenti dinamici: illimitato	No
Numero massimo di attività del contact center	3 per percorso	No
Regole relative al numero massimo di giorni di chiusura	20 per canale	No

Risorsa	Quota predefinita	Idoneità all'incremento
Lunghezza massima del nome della regola relativa ai giorni di chiusura	150 caratteri	No
Numero massimo di giorni tra l'ora di inizio e di fine per una regola dei giorni di chiusura	7 giorni	No
Numero massimo di regole relative agli orari di apertura	4 al giorno	No

Quote di Lambda

Le quote seguenti si applicano alle configurazioni Amazon Pinpoint per il recupero e l'elaborazione dei dati da Lambda.

Risorsa	Quota predefinita	Idoneità all'incremento
Dimensione massima di un payload di richiamo (richiesta e risposta) per una funzione Lambda	6 MB	No
Quantità massima di tempo da attendere per l'elaborazione dei dati da parte di una funzione Lambda	15 secondi	No
Numero massimo di attributi di evento per endpoint	5	No
Numero massimo di caratteri per un nome di attributo di evento	128 caratteri	No

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di caratteri per valore di attributo di evento	128 caratteri	No
Numero massimo di giorni di durata di esecuzione di un percorso	540 giorni	No

Quote di machine learning

Le quote seguenti si applicano alle configurazioni Amazon Pinpoint per il recupero e l'elaborazione dei dati dai modelli di machine learning (ML).

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di configurazioni del modello	1 per modello di messaggio 100 per account	No
Numero massimo di suggerimenti	5 per endpoint o utente	No
Numero massimo di attributi suggeriti per endpoint o utente	1, se i valori degli attributi non vengono elaborati da una funzione AWS Lambda 10, se i valori degli attributi vengono elaborati da una funzione AWS Lambda	No
Lunghezza massima del nome di un attributo suggerito	50 caratteri per un nome di attributo 25 caratteri per il nome visualizzato di un attributo (il nome visualizzato in Attribute	No

Risorsa	Quota predefinita	Idoneità all'incremento
	finder (Finder attributi) sulla console)	
Lunghezza massima di un valore di attributo consigliato recuperato da Amazon Personalize	100 caratteri	No
Dimensione massima di un payload di richiamo (richiesta e risposta) per una funzione Lambda	6 MB	No
Quantità massima di tempo da attendere per l'elaborazione dei dati da parte di una funzione Lambda	15 secondi	No
Numero massimo di tentativi di richiamo una funzione Lambda	3 tentativi	No

A seconda di come configuri Amazon Pinpoint per l'utilizzo di un modello ML, è possibile che vengano applicate quote aggiuntive. Per ulteriori informazioni sulle quote di Amazon Personalize, consulta Quote nella Guida per gli sviluppatori di Amazon Personalize. Per informazioni sulle quote AWS Lambda , consulta Quote nella Guida per gli sviluppatori AWS Lambda .

Quote per i modelli di messaggio

Le quote seguenti si applicano ai modelli di messaggio per l'account Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di modelli di messaggio	20.000 per account	No
Numero massimo di versioni	5.000 per modello	No

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di caratteri in un modello e-mail	600.000 caratteri	No
Numero massimo di caratteri in un modello in-app	200.000 caratteri	No
Numero massimo di caratteri nelle parti del modello di default di un modello di notifica push	4.000 caratteri	No
Numero massimo di caratteri nelle parti del modello specifiche di ADM di un modello di notifica push	6.000 caratteri	No
Numero massimo di caratteri nelle parti APNs specifiche del modello di notifica push	4.000 caratteri	No
Numero massimo di caratteri nelle parti del modello specifiche di Baidu di un modello di notifica push	4.000 caratteri	No
Numero massimo di caratteri nelle parti del modello specifiche di FCM di un modello di notifica push	4.000 caratteri	No
Numero massimo di caratteri in un modello SMS	1.600 caratteri	No
Numero massimo di caratteri in un modello vocale	10,000 caratteri	No

Quote per le notifiche push

Le quote seguenti si applicano ai messaggi inviati da Amazon Pinpoint tramite i canali di notifiche push.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di notifiche push che possono essere inviate al secondo in una campagna	25.000 notifiche al secondo	Sì
Dimensione del payload dei messaggi di Amazon Device Messaging (ADM)	6 KB per messaggio	No
Dimensione del payload dei messaggi del servizio Apple Push Notification (APNs)	4 KB per messaggio	No
APNs dimensione del payload dei messaggi sandbox	4 KB per messaggio	No
Dimensione del payload dei messaggi di Baidu Cloud Push	4 KB per messaggio	No
Dimensione del payload dei messaggi Firebase Cloud Messaging (FCM)	4 KB per messaggio	No

Quote per i messaggi in-app

La seguente quota si applica ai messaggi in-app gestiti mediante Amazon Pinpoint.

Le seguenti quote si applicano singolarmente Regione AWS e alcune possono essere aumentate. Per ulteriori informazioni, consulta Richiesta di un aumento delle quote nella Guida per l'utente di Service Quotas.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di chiamate all'API `GetInAppMessages` al secondo.	5.000 richieste al secondo	Sì
Campagne di messaggistica in-app	Ogni progetto può includere fino a 25 campagne che utilizzano il canale di messaggistica in-app.	Sì, consulta Richiesta di un aumento di quota nella Guida per l'utente di Service Quotas.

Quote per i segmenti

Le quote seguenti si applicano alla risorsa Segmenti dell'API Amazon Pinpoint.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di dimensioni che è possibile utilizzare per creare un segmento	100 per segmento	No
Numero massimo di gruppi di segmenti per segmento	5	No
Numero massimo di segmenti di origine per segmento	5	No
Profondità massima dei segmenti di origine Ad esempio, se un segmento ha un segmento di origine e anche quest'ultimo ha un segmento di origine, la catena di profondità non supera questo limite.	5	No

Quote per SMS

Per le quote SMS, consulta le quote SMS nella guida per l'utente SMS di messaggistica con l'utente AWS finale.

Quote 10DLC

Per le quote di 10 DLC, consulta le quote di 10 DLC nella guida per l'utente di End User Messaging SMS. AWS

Quote per i messaggi vocali

Per le quote vocali, consulta Quote vocali nella guida per l'utente di End User Messaging SMS. AWS

Richiesta di aumento delle quote

Se il valore nella colonna Eligible for Increase (Possibilità di incremento) in una delle tabelle riportate sopra è Yes (Sì), puoi richiedere un incremento di quella quota.

Richiesta di un aumento delle quote

1. Accedi a/ AWS Management Console . https://console.aws.amazon.com
2. Crea un nuovo caso AWS Support all'indirizzo https://console.aws.amazon.com/support/home#/case/create.
3. Nella scheda I tuoi casi di supporto, scegli Create caso.
4. Scegli Stai cercando di aumentare i limiti di servizio?
5. In Aumento della quota di servizio, per Assistenza, scegli una delle seguenti opzioni:

   - Per richiedere un aumento della quota relativa al canale e-mail, scegliere Pinpoint Email (E-mail Pinpoint).
   - Per richiedere un aumento della quota per i limiti di spesa o le tariffe di invio di SMS, scegli SMS Pinpoint. Per tutti gli altri aumenti di quote per SMS, scegli Pinpoint
   - Per richiedere un aumento della quota relativa al canale vocale, scegliere Pinpoint Voice (Voce Pinpoint).
   - Per richiedere un aumento della quota relativa a qualsiasi altra funzionalità di Amazon Pinpoint, scegli Pinpoint.

6. A seconda del servizio scelto, potrebbe esserti chiesto di inserire quanto segue:

   - (Facoltativo) In Fornisci un link al sito o all'app che invierà messaggi SMS, fornisci informazioni relative al sito Web, all'applicazione o al servizio che invierà messaggi SMS.
   - (Facoltativo) In Tipo di messaggi di cui è previsto l'invio, scegli il tipo di messaggio che intendi inviare con codici lunghi:
     - One-Time Password (Password una tantum) - Messaggi che forniscono password che i clienti utilizzano per l'autenticazione a un sito o un'applicazione.
     - Promotional (Promozionale) - Messaggi non critici che promuovono l'azienda o un servizio, ad esempio offerte speciali o annunci.
     - Transactional (Transazionale) - Messaggi informativi importanti che supportano le transazioni con i clienti, come conferme d'ordine o avvisi dell'account. I messaggi transazionali non devono contenere contenuti promozionali o di marketing.
   - (Facoltativo) Da quale AWS regione invierai i messaggi, scegli la regione da cui invierai i messaggi.
   - (Facoltativo) In Paesi a cui si prevede di inviare messaggi, immetti il paese o la regione in cui desideri acquistare codici brevi.
   - (Facoltativo) Nella sezione In che modo i clienti decidono di ricevere messaggi dall'utente, fornisci dettagli sul processo di consenso esplicito.
   - (Facoltativo) Nel campo Fornire il modello di messaggio che si intende utilizzare per inviare messaggi ai clienti, includi il modello che utilizzerai.

7. In Requests (Richieste), procedere come segue:

   - Per la regione scegli la tua Regione AWS.
   - In Resource Type (Tipo di risorsa) scegliere General Limits (Limiti generali). Il campo Tipo di risorsa è presente solo per alcuni Servizi.
   - Per Quota scegli la quota da modificare.
   - In Nuovo valore della quota, inserisci un nuovo valore per la quota.
   - Per richiedere un aumento della stessa quota in un'altra Regione AWS, scegli Aggiungi un'altra richiesta, quindi scegli quella aggiuntiva Regione AWS e compila la nuova richiesta.

8. Scegliere la quota che si vuole aumentare, quindi immettere il nuovo valore desiderato per la quota.

9. Nella sezione Descrizione del caso, spiega perché richiedi l'aumento della quota.

10. In Opzioni di contatto, per Lingua di contatto preferita, scegli la lingua che preferisci usare per comunicare con il team di AWS supporto.
11. Per Metodo di contatto, scegli il metodo di comunicazione preferito con il team di AWS Supporto.
12. Scegli Invia.

Il team di AWS Support fornisce una prima risposta alla tua richiesta entro 24 ore.

Per evitare che i nostri sistemi vengano utilizzati per l'invio di contenuti indesiderati o dannosi, ogni richiesta dovrà essere analizzata attentamente da parte nostra. In seguito a questa valutazione, saremo in grado di gestire la tua richiesta durante le prime 24 ore. Tuttavia, se la risoluzione richiede l'invio di ulteriori informazioni da parte tua, i tempi di gestione della richiesta potranno essere più lunghi.

Potremmo non essere in grado di soddisfare la tua richiesta qualora il tuo caso d'uso non fosse allineato alle nostre policy.

Cronologia dei documenti per Amazon Pinpoint

La tabella seguente descrive le modifiche importanti introdotte in ogni rilascio della Guida per gli sviluppatori di Amazon Pinpoint dopo dicembre 2018. Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile sottoscrivere un feed RSS.

- Ultimo aggiornamento della documentazione: 16 novembre 2023

Modifica	Descrizione	Data
Avviso di fine del supporto	Avviso di fine del supporto: il 30 ottobre 2026 AWS terminerà il supporto per Amazon Pinpoint. Dopo il 30 ottobre 2026, non potrai più accedere alla console Amazon Pinpoint o alle risorse Amazon Pinpoint (endpoint, segmenti, campagne, percorsi e analisi). Per ulteriori informazioni, consulta la pagina relativa alla fine del supporto di Amazon Pinpoint. Nota: per quanto APIs riguarda gli SMS, i comandi vocali, i messaggi push su dispositivi mobili, l'OTP e la convalida del numero di telefono non sono interessati da questa modifica e sono supportati da AWS End User Messaging.	20 maggio 2025
Aggiunto CloudTrail il supporto per l'invio di messaggi	È CloudTrail stato aggiunto il supporto per la PutEvents registrazione e SendUserM essages le chiamate	17 febbraio 2025

	SendMessages API. Vedi Azioni dell'API Amazon Pinpoint supportate CloudTrail nei file di registro.	
Amazon Pinpoint ha aggiornato la documentazione relativa alla guida per l'utente	Per ottenere le informazioni più recenti su come creare, configurare e gestire le risorse Push, consulta la nuova End User Messaging Push User Guide.	22 luglio 2024
Intestazioni di posta elettronica	Puoi aggiungere intestazioni di posta elettronica ai tuoi messaggi di posta elettroni ca. Per ulteriori informazioni, consulta Invio di un'e-mail con intestazioni di annullamento dell'iscrizione.	7 maggio 2024
Orchestrazione della posta elettronica	Amazon Pinpoint ha aggiornato il modo in cui utilizza le risorse Amazon SES per inviare e-mail. Per ulteriori informazioni, consulta il ruolo IAM per l'invio di e-mail con Amazon SES.	30 aprile 2024

Amazon Pinpoint ha aggiornato la documentazione relativa alla guida per l'utente	Gli argomenti sulla gestione delle risorse SMS e vocali vengono ora reindirizzati alla Guida per l'utente SMS di messaggistica per l'utente AWS finale. Per ulteriori informazioni, consulta AWS la Guida per l'utente SMS di messaggistica con l'utente finale.	8 febbraio 2024
Quote di Amazon Pinpoint	Sono state aggiunte quote per le regole relative al numero massimo di giorni di chiusura, alla lunghezza massima del nome della regola dei giorni di chiusura, al numero massimo di giorni tra l'ora di inizio e di fine per una regola dei giorni di chiusura e al numero massimo di regole sugli orari di apertura. Per ulteriori informazioni, consulta Quote di Amazon Pinpoint.	19 dicembre 2023
Amazon Pinpoint ha aggiornato la documentazione della guida per l'utente	Per ottenere le informazioni più recenti su come creare, configurare e gestire le risorse SMS e vocali di messaggistica con l'utente AWS finale, consulta la nuova guida utente SMS per la messaggistica con l'utente AWS finale.	16 novembre 2023

Quote di Amazon Pinpoint	Sono state aggiornate le quote per UpdateEndpointsBatch, UpdateEndpoint, PutEvents DeleteEndpoint, e GetEndpoint. Per ulteriori informazioni, consulta Quote di Amazon Pinpoint.	22 settembre 2023
Quote di Amazon Pinpoint	Sono state aggiornate le quote per CreateEmailTemplate e CreateSmsTemplate, , CreatePushTemplate , CreateInAppTemplate, CreateVoiceTemplate, UpdateEmailTemplate, UpdateSmsTemplate UpdatePushTemplate UpdateInAppTemplate, UpdateVoiceTemplate e. CreateImportJob Per ulteriori informazioni, consulta Quote di Amazon Pinpoint.	12 settembre 2023
Metriche relative all'esecuzione di percorsi e campagne	Sono state aggiunte nuove metriche di analisi per percorsi e campagne. Per ulteriori informazioni, consulta Metriche relative all'esecuzione di percorsi e campagne.	25 aprile 2023
Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint	Amazon Pinpoint ora supporta gli endpoint VPC di interfaccia. Per ulteriori informazioni, consulta Creazione di un endpoint VPC di interfaccia per Amazon Pinpoint.	11 aprile 2023

Crittografia in transito	A partire dal 22/03/2023 Amazon Pinpoint non supporterà più TLS 1.0 ma potrai comunque utilizzare TLS 1.2 o versione successiva. Per maggiori informazioni, consultare Crittografia dei dati in transito.	20 marzo 2023
Quote di Amazon Pinpoint	È stata aggiornata la procedura per richiedere un aumento della quota per le campagne, i percorsi e i messaggi in-app. Per ulteriori informazioni, consulta Quote di Amazon Pinpoint.	16 dicembre 2022
Disponibilità regionale	Amazon Pinpoint è ora disponibile nella seguente regione: Stati Uniti orientali (Ohio).	5 ottobre 2022
Aggiornamenti degli esempi di ruoli IAM	Sono stati aggiornati diversi esempi di ruoli IAM in tutto il documento per allinearli meglio alle best practice della sicurezza.	27 maggio 2022

API SMS e Voce versione 2	Amazon Pinpoint ora include un'API dedicata per l'invio di SMS e messaggi vocali. Questa API include nuove funzionalità, come set di configurazione, pool ed elenchi di opt-out, utili per i clienti che inviano SMS e messaggi vocali a livello transazionale. Per ulteriori informazioni, consulta Utilizzo dell'API SMS e Voce versione 2 di Amazon Pinpoint.	1 aprile 2022
Creazione e convalida di codici OTP (One-Time Password)	Amazon Pinpoint ora include una funzionalità che genera password monouso (OTPs) e le invia agli utenti come messaggi SMS. Include anche un'API per la convalida dei codici OTP quando gli utenti li inseriscono nell'applicazione o nel sito. Per ulteriori informazioni, consulta Invio e convalida di password monouso (). OTPs	26 novembre 2021
Messaggi in-app	Sono state aggiunte informazioni sull'integrazione della funzionalità di messaggistica in-app di Amazon Pinpoint con le app.	10 novembre 2021
Esempi di codice	È stata aggiunta una libreria di esempi di codice per le operazioni più comuni di Amazon Pinpoint.	3 novembre 2021

Quote per i progetti	Il numero massimo di progetti Amazon Pinpoint rimane a 100, ma questa quota può ora essere aumentata aprendo una richiesta di aumento del limite di servizio con. Supporto	11 ottobre 2021
Aggiornamenti delle policy Lambda	Alcune policy di autorizzazione Lambda devono ora includere una condizione `AWS:SourceAccount` . Sono state aggiornate le policy di esempio negli argomenti Creazione di canali personalizzati in Amazon Pinpoint e Personalizzazione dei segmenti con AWS Lambda per soddisfare questo requisito.	7 ottobre 2021
UpdateEndpoint	L'UpdateEndpoint API Amazon Pinpoint è ora registrata da. CloudTrail	16 Novembre 2020
Attributi personalizzati	Amazon Pinpoint ora supporta 250 attributi nei modelli di messaggistica e-mail. Consulta l'argomento Quote.	18 settembre 2020
Disponibilità regionale	Amazon Pinpoint ora è disponibile nelle seguenti regioni: Asia Pacifico (Tokyo), Europa (Londra) e Canada (Centrale). L'API SMS e Voce di Amazon Pinpoint non è disponibile nelle seguenti regioni:	10 settembre 2020

Disponibilità regionale	Amazon Pinpoint ora è disponibile nella regione Asia Pacifico (Tokyo). Tieni presente che l'API SMS e Voce di Amazon Pinpoint non supporta la funzionalità vocale in questa regione.	2 settembre 2020
Eventi delle campagne	Sono state aggiunte informazioni su un nuovo parametro `delivery_type` degli eventi della campagna.	2 agosto 2020
Disponibilità regionale	Amazon Pinpoint ora è disponibile nella regione Asia Pacifico (Seoul). Tieni presente che l'API Amazon Pinpoint non supporta chiamate vocali o SMS in questa regione.	31 luglio 2020
Disponibilità regionale	Amazon Pinpoint è ora disponibile nella AWS GovCloud (US) regione.	30 aprile 2020
Canali personalizzati	Sono state aggiornate le Informazioni sulla creazione di canali personalizzati mediante funzioni Lambda o webhook.	23 aprile 2020
Machine learning	Sono state aggiunte informazioni sul recupero di consigli personalizzati dai modelli di raccomandazione e, facoltativamente, sul miglioramento di tali consigli utilizzando le funzioni. AWS Lambda	4 marzo 2020

Sicurezza	È stato aggiunto un capitolo sulla sicurezza, che fornisce informazioni su vari controlli di sicurezza e funzionalità di Amazon Pinpoint.	4 febbraio 2020
Percorsi	Sono state aggiunte informazioni sull'utilizzo dei percorsi Amazon Pinpoint per sviluppare flussi di lavoro automatizzati che eseguono attività di messaggistica per i progetti. Sono state inoltre aggiunte informazioni sull'interrogazione dei dati analitici per un sottoinsieme di parametri applicabili ai percorsi.	31 ottobre 2019
Analisi	Aggiunte procedure che spiegano come eseguire query sui dati di analisi per campagne e messaggi transazionali e aggiunte informazioni sull'utilizzo dei risultati delle query.	17 ottobre 2019
Analisi	Sono state aggiunte informazioni sull'esecuzione di query sui dati di analisi per un sottoinsieme di parametri applicabili ai messaggi SMS e alle e-mail transazionali.	6 settembre 2019

Esempi di codice	Sono stati aggiunti esempi di codice che possono essere utilizzati per inviare notifiche push transazionali utilizzan do tutti i servizi supportati da Amazon Pinpoint.	30 luglio 2019
Analisi	Aggiunte informazioni sull'esecuzione di query per le attività di analisi dei dati per un sottoinsieme di parametri che si applicano a progetti (applicazioni) e campagne.	24 luglio 2019
Segmenti	È stato aggiunto un tutorial che descrive una soluzione per l'importazione dei dati dei clienti in Amazon Pinpoint da sistemi esterni, ad esempio Salesforce o Marketo.	14 maggio 2019
Disponibilità regionale	Amazon Pinpoint è ora disponibile nelle regioni AWS Asia Pacifico (Mumbai) e Asia Pacifico (Sydney).	25 aprile 2019
Utilizzo di postman con Amazon Pinpoint	È stato aggiunto un tutorial che descrive come utilizzare Postman per l'interazione con l'API Amazon Pinpoint.	8 Aprile 2019
Assegnazione di tag	Sono state aggiunte informazi oni sull'assegnazione di tag alle risorse Amazon Pinpoint.	27 febbraio 2019

Registrazione SMS	Aggiunta un capitolo di tutorial e aggiunto un tutorial che descrive come creare una soluzione che gestisce la registrazione degli utenti tramite SMS.	27 febbraio 2019
Esempi di codice	Aggiunti esempi di codice in vari linguaggi di programma zione che illustrano come inviare messaggi e-mail, SMS e vocali a livello di programma zione.	6 febbraio 2019

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti introdotte in ogni rilascio della Guida per gli sviluppatori di Amazon Pinpoint prima di dicembre 2018.

Modifica	Descrizione	Data
Disponibilità regionale	Amazon Pinpoint è ora disponibile nelle regioni AWS Stati Uniti occidentali (Oregon) ed Europa (Francoforte).	21 dicembre 2018
Canale vocale	È possibile utilizzare il nuovo canale vocale Amazon Pinpoint per creare messaggi vocali e recapitarli ai clienti tramite telefono. Attualmente, è possibile inviare messaggi vocali solo utilizzando l'API API SMS e Voce di Amazon Pinpoint.	15 novembre 2018

Modifica	Descrizione	Data
Disponibilità in Europa (Irlanda)	Amazon Pinpoint è ora disponibile nella regione AWS Europa (Irlanda).	25 ottobre 2018
API Events	È possibile utilizzare l'API Amazon Pinpoint per registrare e eventi e associarli agli endpoint.	7 agosto 2018
Esempi di codice per la definizione e la ricerca di endpoint	Sono stati aggiunti esempi di codice come illustrano come definire, aggiornare, eliminare e cercare endpoint. Vengono forniti esempi per AWS CLI AWS SDK per Java, e l'API Amazon Pinpoint. Per ulteriori informazioni, consulta Usa gli endpoint per rappresentare il tuo pubblico in Amazon Pinpoint.	7 agosto 2018
Autorizzazioni di esportazione degli endpoint	Configura una policy IAM che consente di esportare gli endpoint Amazon Pinpoint in un bucket Amazon S3.	1 maggio 2018
Verifica dei numeri di telefono per SMS	È possibile utilizzare l'API Amazon Pinpoint per verificare un numero di telefono al fine di determinare se è una destinazione valida per i messaggi SMS.	23 aprile 2018

Modifica	Descrizione	Data
Aggiornamento degli argomenti per l'integrazione di Amazon Pinpoint	Integra Amazon Pinpoint con il tuo Android, iOS o JavaScript un'applicazione utilizzando AWS SDKs le nostre librerie.	23 marzo 2018
AWS CloudTrail registrazione	Sono state aggiunte informazioni sulla registrazione delle chiamate API Amazon Pinpoint con. CloudTrail	6 febbraio 2018
Service Quotas aggiornate	Quote aggiornate con ulteriori informazioni sulle quote per l'e-mail.	19 gennaio 2018
Beta pubblica per le estensioni Amazon Pinpoint	Usa AWS Lambda le funzioni per personalizzare i segmenti o creare canali di messaggistica personalizzati.	28 novembre 2017
Quote del payload delle notifiche push	Le quote includono le dimensioni del payload per i messaggi push ai dispositivi mobili.	25 ottobre 2017
Service Quotas aggiornate	Sono state aggiunte informazioni sui canali SMS ed e-mail in Quote.	9 Ottobre 2017
Push per dispositivi mobili ADM e Baidu	È possibile aggiornare il codice dell'app per gestire le notifiche push dai canali push per dispositivi mobili Baidu e ADM.	27 settembre 2017

Modifica	Descrizione	Data
Eventi di utenti IDs e autenticazione con pool di utenti di Amazon Cognito.	Se utilizzi i pool di utenti di Amazon Cognito per gestire l'accesso degli utenti nelle tue app mobili, Amazon Cognito assegna gli utenti IDs agli endpoint e riporta gli eventi di autenticazione ad Amazon Pinpoint.	26 settembre 2017
Utente IDs	Assegna l'utente IDs agli endpoint per monitorare l'utilizzo delle app da parte dei singoli utenti. Vengono forniti esempi per AWS Mobile SDKs e SDK for Java.	31 agosto 2017
Eventi di autenticazione	Il reporting degli eventi di autenticazione consente di scoprire la frequenza con cui gli utenti eseguono l'autenticazione nell'app. Alcuni esempi sono disponibili in Segnala gli eventi di Amazon Pinpoint nella tua applicazione.	31 agosto 2017
Eventi di esempio aggiornati	Gli eventi di esempio includono eventi che Amazon Pinpoint invia in streaming per le attività di e-mail e SMS.	08 giugno 2017
Gestione delle sessioni Android	È possibile gestire le sessioni nelle app Android utilizzando una classe fornita dall'app AWS Mobile Hub di esempio.	20 aprile 2017

Modifica	Descrizione	Data
Esempi di eventi di monetizzazione aggiornati	Il codice di esempio è stato aggiornato per il reporting degli eventi di monetizzazione. .	31 marzo 2017
Flussi di eventi	È possibile configurare Amazon Pinpoint per inviare gli eventi relativi ad app e campagne a un flusso Kinesis.	24 marzo 2017
Autorizzazioni	Come Amazon Pinpoint funziona con IAMPer informazioni su come concedere l'accesso ad Amazon Pinpoint agli utenti del tuo account e agli utenti della tua app AWS per dispositivi mobili, consulta la pagina.	12 gennaio 2017
Disponibilità generale di Amazon Pinpoint	Questo rilascio introduce Amazon Pinpoint.	1° dicembre 2016