aws

Entwicklerhandbuch

Amazon Pinpoint

Amazon Pinpoint: Entwicklerhandbuch

Table of Contents

Was ist Amazon Pinpoint?

Amazon Pinpoint ist ein AWS-Service, über den Sie über mehrere Messaging-Kanäle mit Ihren Kunden interagieren können. Sie können mit Amazon Pinpoint Push-Benachrichtigungen, E-Mails, SMS-Nachrichten oder Sprachnachrichten senden.

Die Informationen in diesem Entwicklerhandbuch richten sich an Anwendungsentwickler. Das Handbuch enthält Informationen zur programmgesteuerten Verwendung der Funktionen von Amazon Pinpoint. Außerdem finden Sie hier interessante Informationen für Entwickler von mobilen Apps wie Verfahren zur Integration von Analyse- und Messaging-Funktionen in Ihre Anwendung.

Es gibt mehrere andere Dokumente, die zu diesem Dokument gehören. Die folgenden Dokumente enthalten Referenzinformationen zu den Amazon-Pinpoint-APIs:

- Amazon-Pinpoint-API-Referenz
- Amazon-Pinpoint-SMS- und -Sprachnachrichten-API

Wenn Sie mit Amazon Pinpoint noch nicht vertraut sind, sollten Sie zunächst das Amazon-Pinpoint-Benutzerhandbuch lesen, bevor Sie mit diesem Dokument fortfahren.

Amazon-Pinpoint-Features

In diesem Abschnitt werden die Hauptfunktionen von Amazon Pinpoint und die Aufgaben beschrieben, die Sie mit deren Verwendung ausführen können.

Definieren von Zielgruppensegmenten

Sie erreichen die richtige Zielgruppe für Ihre Nachrichten, indem Sie Zielgruppensegmente definieren. Ein Segment legt fest, welche Benutzer die Nachrichten erhalten, die von einer Kampagne gesendet werden. Sie können dynamische Segmente basierend auf Daten definieren, die von Ihrer Anwendung gemeldet werden, beispielsweise Informationen zu Betriebssystem oder Typ des Mobilgeräts. Sie können auch statische Segmente importieren, die Sie mit einem anderen Service oder einer anderen Anwendung definieren.

Ansprechen Ihrer Zielgruppe durch Messaging-Kampagnen

Sie sprechen Ihre Zielgruppe an, indem Sie eine Messaging-Kampagne erstellen. Eine Kampagne sendet maßgeschneiderte Nachrichten nach einem von Ihnen festgelegten Zeitplan. Sie können

Kampagnen erstellen, die mobile Push-Nachrichten, E-Mail-Nachrichten oder SMS-Nachrichten senden.

Um mit alternativen Kampagnenstrategien zu experimentieren, konfigurieren Sie Ihre Kampagne als A/B-Test und analysieren die Ergebnisse mit Amazon-Pinpoint-Analysen.

Senden von Transaktionsnachrichten

Sorgen Sie dafür, dass Ihre Kunden informiert bleiben, indem Sie bestimmten Benutzern direkt mobile Transaktions-Push- und SMS-Nachrichten senden, beispielsweise Nachrichten zur Aktivierung neuer Konten, Bestellbestätigungen und Benachrichtigungen zum Passwort, zurücksetzungen. Sie können Transaktionsnachrichten senden, indem Sie die Amazon-Pinpoint-REST-API verwenden.

Analysieren des Benutzerverhaltens

Mithilfe der von Amazon Pinpoint bereitgestellten Analysen erhalten Sie Einsichten in Ihre Zielgruppe und die Effektivität Ihrer Kampagnen. Sie können Trends zum Engagement der Benutzer, zu ihren Kaufaktivitäten zur Demografie usw. anzeigen. Außerdem können Sie den Nachrichtenverkehr überwachen, indem Sie Metriken anzeigen, z. B. die Gesamtzahl der Nachrichten, die für eine Kampagne oder Anwendung geöffnet oder gesendet wurden. Über die Amazon-Pinpoint-API kann die Anwendung benutzerdefinierte Daten in Form von Berichten bereitstellen, die Amazon Pinpoint für Analysen verfügbar macht. Sie können die Analysedaten auf bestimmte Standardmetriken abfragen.

Um Analysedaten außerhalb von Amazon Pinpoint zu analysieren oder zu speichern, können Sie Amazon Pinpoint so konfigurieren, dass die Daten an Amazon Kinesis gestreamt werden.

Regionale Verfügbarkeit

Amazon Pinpoint ist in mehreren AWS-Regionen in Nordamerika, Europa, Asien und Ozeanien verfügbar. In jeder Region unterhält AWS mehrere Availability Zones. Diese Availability Zones sind physisch voneinander isoliert, jedoch durch private, hochredundante Netzwerkverbindungen mit geringer Latenz und hohem Durchsatz miteinander verbunden. Mithilfe dieser Availability Zones können wir ein sehr hohes Maß an Verfügbarkeit und Redundanz bieten und dabei gleichzeitig die Latenz minimieren.

Weitere Informationen zu AWS-Regionen finden Sie unter Verwalten von AWS-Regionen in der Allgemeine Amazon Web Services-Referenz. Eine Liste aller Regionen, in denen Amazon Pinpoint derzeit verfügbar ist, finden Sie unter Amazon-Pinpoint-Endpunkte und -Kontingente und AWS-

Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz. Weitere Informationen über die in jeder Region verfügbare Anzahl von Availability Zones finden Sie unter Globale AWS-Infrastruktur.

Tutorials

Die Tutorials in diesem Abschnitt sollen neuen Amazon-Pinpoint-Benutzern zeigen, wie man einige wichtige Aufgaben erledigt. Wenn Amazon Pinpoint für Sie neu ist, oder wenn Sie nur mit bestimmten Funktionen nicht vertraut sind, sind diese Tutorials ein guter Ausgangspunkt.

Die Tutorials in diesem Handbuch umfassen Aufgaben, die sich an Entwickler oder Systemadministratoren richten. Diese Tutorials zeigen Ihnen, wie Sie Aufgaben mithilfe der Amazon-Pinpoint-API, AWS-SDKs und der AWS CLI ausführen können. Wenn Sie mit Amazon Pinpoint hauptsächlich über die webbasierte Konsole interagieren, lesen Sie den Abschnitt „Tutorials“ im Amazon-Pinpoint-Benutzerhandbuch.

Tutorials

- Tutorial: Verwenden von Postman mit der Amazon-Pinpoint-API
- Tutorial: Einrichten eines SMS-Registrierungssystems

Tutorial: Verwenden von Postman mit der Amazon-Pinpoint-API

Postman ist ein beliebtes Tool zum Testen von APIs in einer benutzerfreundlichen grafischen Umgebung. Sie können mit Postman API-Anfragen an eine REST-API senden und Antworten auf Ihre Anfragen empfangen. Postman bietet eine bequeme Möglichkeit, die Aufrufe an die Amazon-Pinpoint-API zu testen und Fehler zu beheben. Dieses Tutorial behandelt Verfahren zur Einrichtung und Verwendung von Postman mit Amazon Pinpoint.

> **Note**
>
> Postman wird von einem Drittanbieter entwickelt. Es wird von Amazon Web Services (AWS) nicht entwickelt oder unterstützt. Weitere Informationen zur Verwendung von Postman oder Hilfe bei Problemen im Zusammenhang mit Postman erhalten Sie im Support Center auf der Postman-Website.

Über dieses Tutorial

Dieser Abschnitt enthält einen Überblick über dieses Tutorial.

Zielgruppe

Dieses Tutorial richtet sich an Entwickler und Systemimplementierer. Sie müssen nicht mit Amazon Pinpoint oder Postman vertraut sein, um die Schritte in diesem Tutorial ausführen zu können. Sie sollten mit der Verwaltung von IAM-Richtlinien und der Änderung von JSON-Codebeispielen vertraut sein.

Die Verfahren in diesem Tutorial wurden entwickelt, um zu verhindern, dass neue Benutzer API-Operationen verwenden, die Amazon-Pinpoint-Ressourcen dauerhaft löschen können. Fortgeschrittene Benutzer können diese Einschränkung aufheben, indem sie die Richtlinie ändern, die ihren -Benutzern zugeordnet ist.

Verwendete Funktionen

Dieses Tutorial enthält Anwendungsbeispiele für das folgende Amazon-Pinpoint-Feature:

- Interaktion mit der Amazon-Pinpoint-API mithilfe von Postman

Benötigte Zeit

Es sollte etwa 15 Minuten dauern, bis Sie dieses Tutorial abgeschlossen haben.

Regionale Einschränkungen

Es sind keine regionalen Einschränkungen mit der Nutzung dieser Lösung verbunden.

Kosten der Ressourcennutzung

Die Einrichtung eines AWS-Kontos ist kostenlos. Durch die Implementierung dieser Lösung können Ihnen jedoch AWS-Nutzungskosten entstehen, wenn Sie Postman für eine der folgenden Aufgaben einsetzen:

- Senden von Nachrichten per E-Mail, SMS, Mobile Push oder Voice
- Erstellen und Senden von Kampagnen
- Verwenden der Funktion zur Telefonnummernüberprüfung

Weitere Informationen zu den Gebühren, die mit der Nutzung von Amazon Pinpoint verbunden sind, finden Sie unter Amazon Pinpoint – Preise.

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Voraussetzungen erfüllen:

- Sie benötigen ein AWS-Konto. Um ein AWS-Konto zu erstellen, gehen Sie zu https://console.aws.amazon.com/ und wählen Sie Neues AWS-Konto erstellen aus.
- Stellen Sie sicher, dass das Konto, mit dem Sie sich bei der AWS Management Console anmelden, in der Lage ist, neue IAM-Richtlinien und -Rollen zu erstellen.
- Stellen Sie sicher, dass Sie mindestens ein Beispielprojekt erstellt haben, für das E-Mail aktiviert ist und für das eine verifizierte E-Mail-Identität vorliegt. Weitere Informationen finden Sie unter Erstellen eines Amazon-Pinpoint-Projekts mit E-Mail-Unterstützung im Amazon-Pinpoint-Benutzerhandbuch.
- Stellen Sie sicher, dass Sie eine AWS-Konto-ID haben. Ihre AWS-Konto-ID finden Sie in der oberen rechten Ecke der Konsole. Sie können auch die Befehlszeilenschnittstelle (CLI) verwenden. Siehe Wie Sie Ihre AWS-Konto-ID finden.
- Sie müssen Postman herunterladen und auf Ihrem Computer installieren. Sie können Postman von der Postman-Website herunterladen.
- Nachdem Sie Postman auf Ihrem Computer installiert haben, müssen Sie ein Postman-Konto erstellen. Wenn Sie die Postman-Anwendung zum ersten Mal starten, werden Sie aufgefordert, sich anzumelden oder ein neues Konto zu erstellen. Folgen Sie den Anweisungen von Postman, um sich bei Ihrem Konto anzumelden oder ein Konto zu erstellen, sofern Sie noch kein Konto haben.

Schritt 1: Erstellen von IAM-Richtlinien und -Rollen

Wenn Sie Postman verwenden, um die Amazon-Pinpoint-API zu testen, besteht der erste Schritt darin, einen Benutzer zu erstellen. In diesem Abschnitt erstellen Sie eine Richtlinie, die es Benutzern erlaubt, mit allen Amazon-Pinpoint-Ressourcen zu interagieren. Anschließend erstellen Sie einen Benutzer und fügen die Richtlinie direkt an den Benutzer an.

Eine IAM-Richtlinie erstellen

Erfahren Sie mehr dazu, wie Sie eine IAM-Richtlinie erstellen. Benutzer und Rollen, die diese Richtlinie verwenden, können mit allen Ressourcen in der Amazon-Pinpoint-API interagieren. Sie bietet auch Zugriff auf Ressourcen, die mit der Amazon-Pinpoint-E-Mail-API sowie der Amazon-Pinpoint-SMS- und Sprachnachrichten-API verknüpft sind.

So erstellen Sie die Richtlinie

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen).
3. Wählen Sie im Policy-Editor JSON aus. Löschen Sie alle JSON-Dateien, die im Richtlinieneditor aktuell sind, sodass sie leer sind. Kopieren Sie den folgenden JSON-Code und fügen Sie ihn in den Policy-Editor ein. Ersetzen Sie dann im Policy-Editor alle Instanzen von *123456789012* durch Ihre ID. AWS-Konto

   Ihre AWS-Konto ID finden Sie in der oberen rechten Ecke der Konsole, oder Sie können die CLI verwenden, siehe Ihre AWS Konto-ID finden.

   > **Note**
   >
   > Zum Schutz der Daten in Ihrem Amazon-Pinpoint-Konto umfasst diese Richtlinie nur Berechtigungen, die Ihnen erlauben, Ressourcen zu lesen, zu erstellen und zu ändern. Sie beinhaltet nicht die Berechtigung, Ressourcen zu löschen. Sie können diese Richtlinie mithilfe des visuellen Editors in der IAM-Konsole ändern. Weitere Informationen finden Sie unter Verwalten der IAM-Richtlinien im IAM-Benutzerhandbuch. Sie können diese Richtlinie auch mithilfe des CreatePolicyVersionVorgangs in der IAM-API aktualisieren.
   > Diese Richtlinie umfasst auch Berechtigungen, die Ihnen erlauben, mit den Services `ses` und `sms-voice` zu interagieren, zusätzlich zum Service `mobiletargeting`. Die Berechtigungen `ses` und `sms-voice` ermöglichen Ihnen die Interaktion mit der Amazon-Pinpoint-E-Mail-API bzw. der Amazon-Pinpoint-SMS- und Sprachnachrichten-API. Die `mobiletargeting`-Berechtigungen ermöglichen Ihnen die Interaktion mit der Amazon-Pinpoint-API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Update*",
                "mobiletargeting:Get*",
                "mobiletargeting:Send*",
                "mobiletargeting:Put*",
```

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

Wählen Sie Weiter aus.

4. Geben Sie unter Richtlinienname einen Namen für die Richtlinie ein, z. B. **`PostmanAccessPolicy`** Wählen Sie Richtlinie erstellen aus.
5. (Optional) Sie können der Richtlinie Tags hinzufügen, indem Sie Tag hinzufügen auswählen.
6. Wählen Sie Weiter: Prüfen aus.

Erstellen eines IAM-Benutzers

> ⚠ Warning
>
> IAM-Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

Nachdem Sie die Richtlinie erstellt haben, können Sie einen Benutzer erstellen und die Richtlinie an diesen anhängen. Wenn Sie den Benutzer erstellen, stellt Ihnen IAM einen Satz von Anmeldeinformationen bereit, mit dem Sie Postman erlauben können, Amazon-Pinpoint-API-Operationen auszuführen.

So erstellen Sie den Benutzer

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie in der IAM-Konsole im Navigationsbereich Benutzer und dann Benutzer erstellen aus.
3. Geben Sie unter Benutzerdetails für Benutzername einen eindeutigen Namen für den Benutzer ein, z. B. **`PostmanUser`**. Wählen Sie anschließend Weiter.
4. Wählen Sie unter Berechtigungen festlegen für Berechtigungsoptionen die Option Direktes Anfügen von Richtlinien aus.
5. Wählen Sie unter Berechtigungsrichtlinien die Richtlinie (**`PostmanAccessPolicy`**) aus, die Sie unter IAM-Richtlinie erstellen erstellt haben. Wählen Sie anschließend Weiter.
6. Auf der Seite Prüfen und erstellen können Sie optional Tags hinzufügen, um Benutzer leichter zu identifizieren. Weitere Informationen zur Verwendung von Tags finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.
7. Wenn Sie bereit sind, den Benutzer zu erstellen, wählen Sie Create user (Benutzer erstellen) aus.

Zugriffsschlüssel erstellen

> **Warning**
>
> Für dieses Szenario sind IAM-Benutzer mit programmatischem Zugriff und langfristigen Anmeldeinformationen erforderlich, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden. Die Zugriffsschlüssel können bei Bedarf aktualisiert werden. Weitere Informationen finden Sie unter Aktualisieren von Zugriffsschlüsseln im IAM-Benutzerhandbuch.

IAM stellt einen Satz von Anmeldeinformationen bereit, mit dem Sie Postman erlauben können, Amazon-Pinpoint-API-Operationen auszuführen.

So erstellen Sie den Benutzer

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich der IAM-Konsole Benutzer aus. Wählen Sie den Benutzer (**`PostmanUser`**) aus, der unter Einen IAM-Benutzer erstellen erstellt wurde, und wählen Sie dann die Registerkarte Sicherheitsanmeldeinformationen aus.
3. Wählen Sie im Abschnitt Access keys (Zugriffsschlüssel) Create access key (Zugriffsschlüssel erstellen).
4. Wählen Sie auf der Seite „Zugriff auf die wichtigsten bewährten Methoden und Alternativen“ die Option Anwendung, die außerhalb AWS ausgeführt wird aus.

   Wählen Sie anschließend Weiter.
5. (Optional) Sie können der Richtlinie ein Beschreibungs-Tag hinzufügen.
6. Wählen Sie Zugriffsschlüssel erstellen aus.
7. Kopieren Sie auf der Seite Zugriffsschlüssel abrufen die Anmeldeinformationen, die in den Spalten Zugriffsschlüssel und Geheimer Zugriffsschlüssel angezeigt werden.

   > **Note**
   >
   > Sie müssen sowohl die Zugriffsschlüssel-ID als auch den geheimen Zugriffsschlüssel in einem späteren Schritt in diesem Tutorial angeben. Dies ist das einzige Mal, dass Sie

den geheimen Zugriffsschlüssel anzeigen können. Wir empfehlen Ihnen, dass Sie ihn kopieren und an einem sicheren Ort speichern.

8. Nachdem Sie beide Schlüssel gespeichert haben, wählen Sie Fertig aus.

Schritt 2: Einrichten von Postman

Nachdem Sie jetzt einen Benutzer erstellt haben, das auf die Amazon-Pinpoint-API zugreifen kann, können Sie Postman einrichten. In diesem Abschnitt erstellen Sie eine oder mehrere Umgebungen in Postman. Als nächstes importieren Sie eine Sammlung, die für jeden der Vorgänge in der Amazon-Pinpoint-API eine Vorlage für Anfragen enthält.

Postman-Workspace erstellen

In Postman ist ein Workspace ein organisatorischer Container für Projekte und Umgebungen. In diesem Abschnitt erstellen Sie mindestens einen Workspace zur Verwendung mit Amazon Pinpoint.

Einen Workspace erstellen

Wählen Sie in Postman die weiteren Aktionen Datei und dann Neu aus.

1. Wählen Sie im Fenster Neu erstellen die Option Workspace aus.
2. Geben Sie einen Namen und eine Zusammenfassung ein und setzen Sie die Sichtbarkeit auf „Persönlich“. Wählen Sie dann Workspace erstellen aus.

Postman-Umgebungen erstellen

In Postman ist eine Umgebung ein Satz von Variablen, die als Schlüssel-Wert-Paare gespeichert werden. Sie können Umgebungen nutzen, um die Konfiguration der Anfragen, die Sie über Postman stellen, zu ändern, ohne die API-Anfragen selbst ändern zu müssen.

In diesem Abschnitt erstellen Sie mindestens eine Umgebung, die Sie mit Amazon Pinpoint verwenden können. Jede Umgebung, die Sie erstellen, enthält einen Satz von Variablen, die spezifisch für Ihr Konto in einer einzigen AWS-Region sind. Wenn Sie die Verfahren in diesem Abschnitt verwenden, um mehr als eine Umgebung zu erstellen, können Sie zwischen den Regionen wechseln, indem Sie eine andere Umgebung aus dem Menü Umgebung in Postman auswählen.

So erstellen Sie eine Umgebung

1. Wählen Sie in Postman die weiteren Aktionsmenüs Datei und dann Neu aus.
2. Wählen Sie im Fenster Create New (Neu erstellen) die Option Environment (Umgebung) aus.
3. Geben Sie im Fenster MANAGE ENVIRONMENTS (Umgebungen verwalten) unter Environment Name (Name der Umgebung) **`Amazon Pinpoint - Region Name`** ein. Ersetzen Sie *`Region Name`* durch einen der folgenden Werte:

   - USA Ost (Nord-Virginia)
   - USA West (Oregon)
   - Asien-Pazifik (Mumbai)
   - Asien-Pazifik (Sydney)
   - Europe (Frankfurt)
   - Europa (Irland)

   > **Note**
   >
   > Sie müssen mindestens eine Umgebung für ein einzelnes Projekt erstellen AWS-Region, und diese AWS-Region muss ein Projekt enthalten. Wenn Sie kein Projekt in einem der zuvor aufgelisteten Projekte erstellt haben AWS-Regionen, finden Sie weitere Informationen unter Erstellen eines Amazon Pinpoint Pinpoint-Projekts mit E-Mail-Support im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

4. Erstellen Sie sechs neue Variablen: `endpoint`, `region`, `serviceName`, `accountId`, `accessKey` und `secretAccessKey`. Verwenden Sie die folgende Tabelle, um zu bestimmen, welcher Wert in die Spalten Anfangswert und Aktueller Wert für jede Variable eingegeben werden soll.

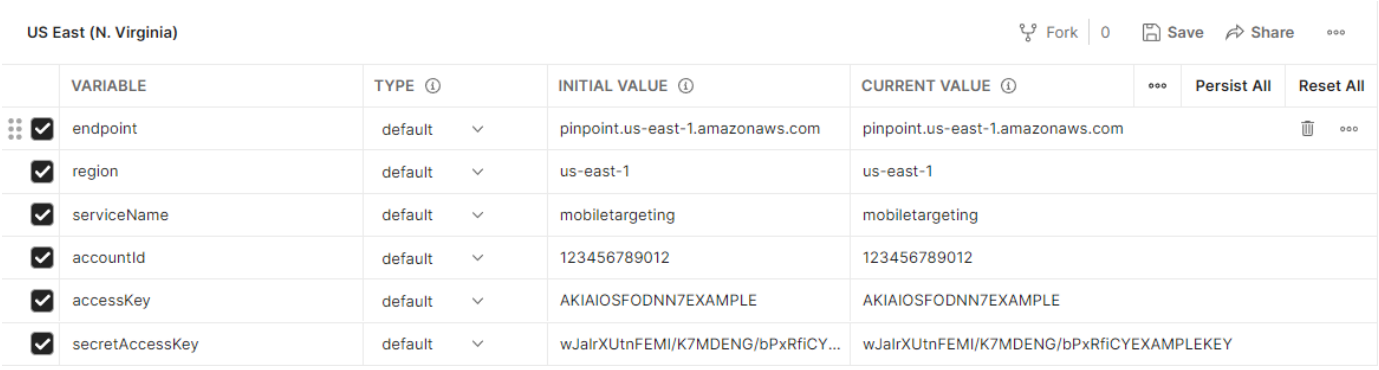
Region	Variable	Anfangswert und aktueller Wert
USA Ost (Nord-Virginia)	`endpoint`	**`pinpoint.us-east-1.amazonaws.com`**
	`region`	**`us-east-1`**

Region	Variable	Anfangswert und aktueller Wert
	`serviceName`	**`mobiletargeting`**
	`accountId`	(Ihre AWS Konto-ID)
	`accessKey`	(die (ID Ihres IAM-Zugriffsschlüssels))
	`secretAccessKey`	(Ihr geheimer IAM-Zugriffsschlüssel)
USA West (Oregon)	`endpoint`	**`pinpoint.us-west-2.amazonaws.com`**
	`region`	**`us-west-2`**
	`serviceName`	**`mobiletargeting`**
	`accountId`	(Ihre AWS Konto-ID)
	`accessKey`	(die (ID Ihres IAM-Zugriffsschlüssels))
	`secretAccessKey`	(Ihr geheimer IAM-Zugriffsschlüssel)
Asien-Pazifik (Mumbai)	`endpoint`	**`pinpoint.ap-south-1.amazonaws.com`**
	`region`	**`ap-south-1`**
	`serviceName`	**`mobiletargeting`**
	`accountId`	(Ihre AWS Konto-ID)

Region	Variable	Anfangswert und aktueller Wert
	accessKey	(die (ID Ihres IAM-Zugriffsschlüssels))
	secretAccessKey	(Ihr geheimer IAM-Zugriffsschlüssel)
Asien-Pazifik (Sydney)	endpoint	**pinpoint.ap-southeast-2.amazonaws.com**
	region	**ap-southeast-2**
	serviceName	**mobiletargeting**
	accountId	(Ihre AWS Konto-ID)
	accessKey	(die (ID Ihres IAM-Zugriffsschlüssels))
	secretAccessKey	(Ihr geheimer IAM-Zugriffsschlüssel)
Europa (Frankfurt)	endpoint	**pinpoint.eu-central-1.amazonaws.com**
	region	**eu-central-1**
	serviceName	**mobiletargeting**
	accountId	(Ihre AWS Konto-ID)
	accessKey	(die (ID Ihres IAM-Zugriffsschlüssels))

Region	Variable	Anfangswert und aktueller Wert
	`secretAccessKey`	(Ihr geheimer IAM-Zugriffsschlüssel)
Europa (Irland)	`endpoint`	**`pinpoint.eu-west-1.amazonaws.com`**
	`region`	**`eu-west-1`**
	`serviceName`	**`mobiletargeting`**
	`accountId`	(Ihre AWS Konto-ID)
	`accessKey`	(die (ID Ihres IAM-Zugriffsschlüssels))
	`secretAccessKey`	(Ihr geheimer IAM-Zugriffsschlüssel)

Nachdem Sie diese Variablen erstellt haben, ähnelt das Fenster MANAGE ENVIRONMENTS (Umgebungen verwalten) dem Beispiel in der folgenden Abbildung.



Wählen Sie Save (Speichern) aus, wenn Sie fertig sind.

> **Important**
>
> Die Zugriffsschlüssel in der vorhergehenden Abbildung sind fiktiv. Geben Sie Ihre IAM-Zugriffsschlüssel nicht an andere weiter.
> Postman enthält Features, mit denen Sie Umgebungen gemeinsam nutzen und exportieren können. Wenn Sie diese Features verwenden, achten Sie darauf, dass Sie Ihre Zugriffsschlüssel-ID und Ihren geheimen Zugriffsschlüssel nicht an Personen weitergeben, die keinen Zugriff auf diese Zugangsdaten haben dürfen.
> Weitere Informationen finden Sie unter Bewährte Methoden für IAM im IAM-Benutzerhandbuch.

5. (Optional) Wiederholen Sie die Schritte 1–4 für jede weitere Umgebung, die Sie erstellen möchten.

> **Tip**
>
> In Postman können Sie so viele Umgebungen erstellen, wie Sie benötigen. Sie können Umgebungen auf folgende Weisen verwenden:
>
> - Erstellen Sie eine separate Umgebung für jede Region, in der Sie die Amazon-Pinpoint-API testen müssen.
> - Erstellen Sie Umgebungen, die mit verschiedenen AWS-Konten verknüpft sind.
> - Erstellen Sie Umgebungen, die Anmeldeinformationen verwenden, die mit anderen Benutzern verknüpft sind.

6. Wenn Sie mit der Erstellung von Umgebungen fertig sind, fahren Sie mit dem nächsten Abschnitt fort.

Erstellen Sie eine Amazon Pinpoint-Sammlung in Postman

In Postman ist eine Sammlung eine Gruppe von API-Anforderungen. Anfragen in einer Sammlung werden in der Regel durch einen gemeinsamen Zweck zusammengefasst. In diesem Abschnitt erstellen Sie eine neue Sammlung, die für jeden Vorgang in der Amazon-Pinpoint-API eine Vorlage für Anfragen enthält.

So erstellen Sie die Amazon Pinpoint-Sammlung

1. Wählen Sie in Postman die weiteren Aktionsmenüs Datei und dann Importieren aus.
2. Wählen Sie im Fenster Import die Option Aus Link importieren und geben Sie dann die folgende URL ein: https://raw.githubusercontent.com/awsdocs/ amazon-pinpoint-developer-guide /master/ Amazon %20pinpoint.postman_collection.json.

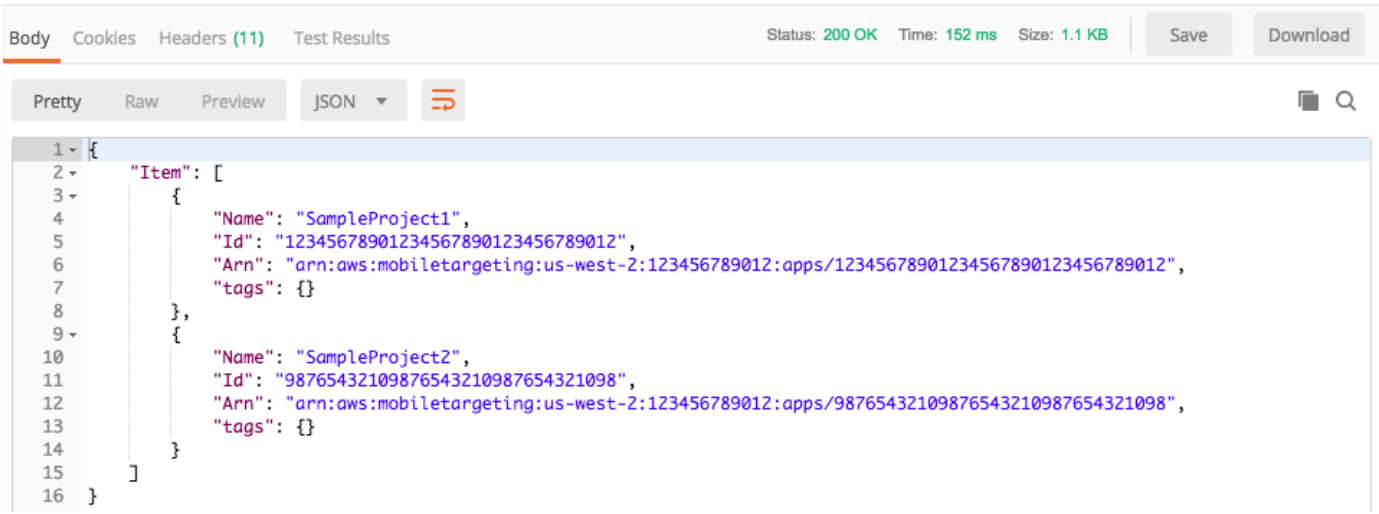
   Wählen Sie Importieren aus. Postman importiert die Amazon-Pinpoint-Sammlung, die 120 Beispielanfragen enthält.

Testen Ihrer Postman-Konfiguration

Nachdem Sie die Amazon-Pinpoint-Sammlung importiert haben, sollten Sie einen schnellen Test durchführen, um sicherzustellen, dass alle Komponenten richtig konfiguriert sind. Sie können Ihre Konfiguration testen, indem Sie eine `GetApps`-Anfrage senden. Diese Anfrage gibt eine Liste aller Projekte zurück, die in Ihrem Amazon-Pinpoint-Konto in der aktuellen AWS-Region-Region existieren. Diese Anfrage erfordert keine zusätzliche Konfiguration, daher ist es eine gute Möglichkeit, Ihre Konfiguration zu testen.

So testen Sie die Konfiguration der Amazon-Pinpoint-Sammlung

1. Wählen Sie im linken Navigationsbereich Sammlungen aus, erweitern Sie die Amazon-Pinpoint-Sammlung und erweitern Sie den Apps-Ordner.
2. GetAppsWählen Sie in der Liste der Anfragen.
3. Verwenden Sie die Umgebungsauswahl, um die Umgebung auszuwählen, die Sie in Create Postman-Umgebungen erstellt haben.
4. Wählen Sie Send (Senden) aus. Wenn die Anfrage erfolgreich gesendet wurde, zeigt der Antwortbereich den Status `200 OK` an. Sie sehen eine Antwort, die dem Beispiel in der folgenden Abbildung ähnelt.



> **Note**
>
> Wenn Sie keine Projekte in der erstellt haben, AWS-Region kehrt Amazon Pinpoint zurück`{ "Item": [] }`.

Diese Antwort zeigt eine Liste aller Amazon-Pinpoint-Projekte, die in Ihrem Konto in der Region existieren, die Sie in Schritt 3 ausgewählt haben.

Fehlerbehebung

Wenn Sie Ihre Anfrage absenden, wird möglicherweise ein Fehler angezeigt. In der folgenden Liste finden Sie einige Fehler, die häufig auftreten können, und Hinweise auf Schritte, die Sie zur Behebung dieser Fehler ergreifen können.

Fehlermeldung	Problem	Auflösung
Could not get any response (Konnte keine Antwort erhalten) There was an error connecting to https://%7B%7Bendpoint%7D%7D/v1/apps. (Es ist ein Fehler aufgetreten bei der	Es gibt keinen aktuellen Wert für die Variable `{{endpoint}}` , die gesetzt wird, wenn Sie eine Umgebung auswählen.	Verwenden Sie das Umgebungsauswahlfeld zur Auswahl einer Umgebung.

Fehlermeldung	Problem	Auflösung
Verbindung zu https://%7B%7Bendpoint%7D%7D/v1/apps.)		
Das Sicherheits-Token der Anfrage ist ungültig.	Postman konnte den aktuellen Wert Ihrer Zugriffsschlüssel-ID oder Ihres geheimen Zugriffsschlüssels nicht finden.	Wählen Sie das Zahnradsymbol neben dem Umgebungsauswahlfeld aus und wählen Sie dann die aktuelle Umgebung aus. Stellen Sie sicher, dass die Werte `accessKey` und `secretAccessKey` sowohl in den Spalten ANFANGSWERT als auch AKTUELLER WERT angezeigt werden und dass Sie die Anmeldeinformationen korrekt eingegeben haben.
„Message“: „Der Benutzer: arn:aws:iam: :123456789012:user/ PinpointPostmanUser ist nicht berechtigt, Folgendes auszuführen: mobiletargeting: on resource: aws:mobiletargeting:us-west-2:123456789012: *“ GetApps	Die IAM-Richtlinie, die Ihrem Benutzer zugeordnet ist, enthält nicht die entsprechenden Berechtigungen.	Vergewissern Sie sich, dass Ihr Benutzer über die unter Erstellen einer IAM-Richtlinie beschriebenen Berechtigungen verfügt und dass Sie die richtigen Anmeldeinformationen angegeben haben, als Sie die Umgebung im Arbeitsbereich Postman erstellen erstellt haben.

Schritt 3: Senden von zusätzlichen Anforderungen

Wenn Sie die Konfiguration und den Test von Postman abgeschlossen haben, können Sie zusätzliche Anfragen an die Amazon-Pinpoint-API senden. Dieser Abschnitt enthält Informationen, die Sie kennen müssen, bevor Sie mit dem Senden von Anfragen beginnen. Es enthält auch zwei Beispielanfragen, in denen beschrieben wird, wie die Amazon-Pinpoint-Sammlung verwendet wird.

> **Important**
>
> Wenn Sie die in diesem Abschnitt beschriebenen Schritte ausführen, senden Sie Anfragen an die Amazon-Pinpoint-API. Diese Anfragen können neue Ressourcen in Ihrem Amazon-Pinpoint-Konto erstellen, bestehende Ressourcen ändern, Nachrichten senden, die Konfiguration Ihrer Amazon-Pinpoint-Projekte ändern und andere Amazon-Pinpoint-Features nutzen. Seien Sie vorsichtig, wenn Sie diese Anfragen ausführen.

Informationen zu den Beispielen in der Amazon-Pinpoint-Postman-Sammlung

Sie müssen die meisten Operationen in der Amazon-Pinpoint-Postman-Sammlung konfigurieren, bevor Sie sie verwenden können. Für `GET`- und `DELETE`-Operationen müssen Sie in der Regel nur die Variablen ändern, die auf der Registerkarte Pre-Request Script (Skript vor Anfrage) eingestellt sind.

> **Note**
>
> Wenn Sie die IAM-Richtlinie verwenden, die unter Eine IAM-Richtlinie erstellen angezeigt wird, können Sie keine der in dieser Sammlung enthaltenen `DELETE` Anforderungen ausführen.

Beispielsweise erfordert die Operation `GetCampaign` die Angabe einer `projectId` und einer `campaignId`. Auf der Registerkarte Pre-Request Script (Skript vor Anfrage) sind beide Variablen vorhanden und mit Beispielwerten ausgefüllt. Löschen Sie die Beispielwerte und ersetzen Sie sie durch die Werte für Ihr Amazon-Pinpoint-Projekt und Ihre Kampagne.

Von diesen Variablen wird die `projectId`-Variable am häufigsten verwendet. Der Wert für diese Variable ist ein eindeutiger Bezeichner für das Projekt, für das Ihre Anforderung gilt. Eine Liste mit Bezeichnern für Ihre Projekte finden Sie in der Antwort auf die `GetApps`-Anforderung, die Sie im vorherigen Schritt dieser Anleitung gesendet haben. In dieser Antwort enthält das `Id`-Feld die eindeutige Kennung für ein Projekt. Weitere Informationen zur `GetApps`-Operation und zur Bedeutung der einzelnen Felder in der Antwort finden Sie unter Apps in der Amazon-Pinpoint-API-Referenz.

> **Note**
>
> In Amazon Pinpoint ist ein „Projekt“ dasselbe wie eine „App“ oder „Anwendung“.

Für `POST`- und `PUT`-Operationen müssen Sie auch den Anfragetext so anpassen, dass er die Werte enthält, die Sie an die API senden möchten. Wenn Sie beispielsweise eine `CreateApp`-Anforderung senden (bei der es sich um eine `POST`-Anforderung handelt), müssen Sie einen Namen für das Projekt angeben, das Sie erstellen. Sie können die Anfrage auf der Registerkarte Body (Text) ändern. Ersetzen Sie in diesem Beispiel den Wert neben `"Name"` durch den Namen des Projekts. Wenn Sie dem Projekt Tags hinzufügen möchten, können Sie diese im `tags`-Objekt angeben. Oder, wenn Sie keine Tags hinzufügen möchten, können Sie das gesamte `tags`-Objekt löschen.

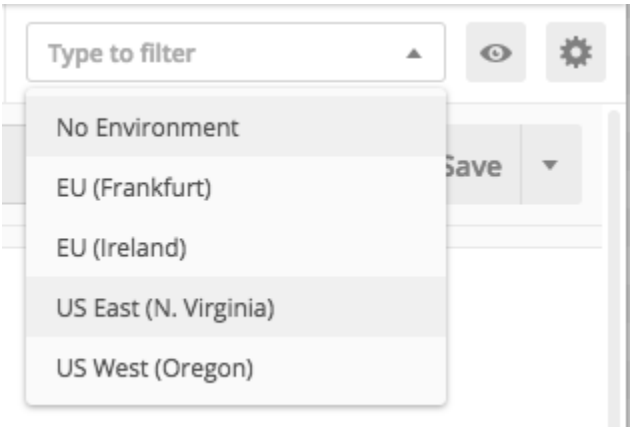
> **Note**
>
> Für die Operation `UntagResource` müssen Sie außerdem URL-Parameter angeben. Sie können diese Parameter auf der Registerkarte Params (Parameter) angeben. Ersetzen Sie die Werte in der Spalte VALUE (Wert) durch die Tags, die Sie für die angegebene Ressource löschen möchten.

Beispiel für eine Anforderung: Erstellen eines Projekts mithilfe der Operation **`CreateApp`**

Bevor Sie Segmente und Kampagnen in Amazon Pinpoint erstellen, müssen Sie zunächst ein Projekt erstellen. In Amazon Pinpoint besteht ein Projekt aus Segmenten, Kampagnen, Konfigurationen und Daten, die durch einen gemeinsamen Zweck miteinander verbunden sind. Beispielsweise könnten Sie in einem Projekt alle Inhalte vereinen, die sich auf eine bestimmte App oder eine bestimmte Marke oder Marketinginitiative beziehen. Wenn Sie Kundeninformationen zu Amazon Pinpoint hinzufügen, werden diese Informationen einem Projekt zugeordnet.

Um ein Projekt durch Senden einer API-Anfrage zu erstellen CreateApp

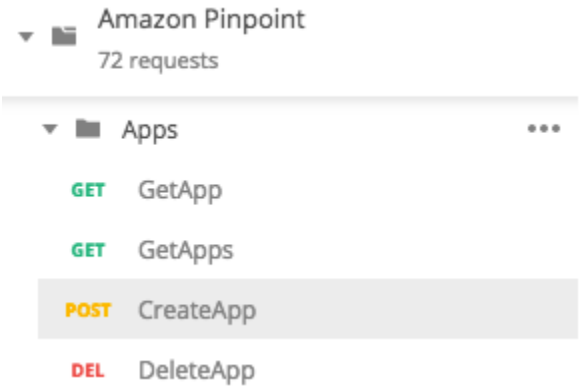
1. Wählen Sie im Menü Umgebungen die aus AWS-Region , in der Sie das Projekt erstellen möchten.



In diesem Beispiel wurde Postman so konfiguriert, dass das Menü Environments (Umgebungen) die folgenden vier Optionen anzeigt:

- USA Ost (Nord-Virginia)
- USA West (Oregon)
- Europe (Frankfurt)
- Europa (Irland)

2. Wählen Sie im Ordner Apps den CreateAppVorgang > aus.



Der Apps-Ordner in der Amazon-Pinpoint-Postman-Sammlung ist erweitert und enthält die folgenden Anfragen:

- `GetApp`
- `GetApps`
- `CreateApp`
- `DeleteApp`

3. Ersetzen Sie auf der Registerkarte Body (Text), neben `"Name"`, den Platzhalterwert (`"string"`) durch einen Namen für die Kampagne, z. B. **`"MySampleProject"`**.
4. Löschen Sie das Komma, das nach dem Kampagnennamen steht, und löschen Sie dann das gesamte `tags`-Objekt in den Zeilen 3 bis 5. Wenn Sie fertig sind, sollte Ihre Anfrage dem Beispiel ähneln, das im folgenden Code-Snippet gezeigt wird.

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

   Postman ist so konfiguriert, dass die Anfrage als unformatierte JSON-Nutzlast gesendet wird.
5. Wählen Sie Send (Senden) aus. Wenn die Kampagne erfolgreich erstellt wurde, zeigt der Antwortbereich den Status `201 Created` an.

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

Beispiel: Senden einer E-Mail mithilfe der Operation **`SendMessages`**

Es ist sehr verbreitet, die Amazon-Pinpoint–`SendMessages`API zu verwenden, um transaktionale Nachrichten zu senden. Ein Vorteil des Sendens von Nachrichten über die `SendMessages`-API (im Gegensatz zum Erstellen von Kampagnen) besteht darin, dass Sie Nachrichten an eine beliebige Adresse senden können (z. B. E-Mail-Adresse, Telefonnummer oder Geräte-Token). Die Adresse, an die Sie Nachrichten senden, muss in Ihrem Amazon-Pinpoint -Konto nicht bereits vorhanden sein. Vergleichen wir dies mit dem Versenden von Nachrichten durch das Erstellen von Kampagnen. Bevor Sie eine Kampagne in Amazon Pinpoint versenden, müssen Sie Ihrem Amazon-Pinpoint-Konto Endpunkte hinzufügen, Segmente erstellen, die Kampagne erstellen und die Kampagne durchführen.

Das Beispiel in diesem Abschnitt zeigt, wie Sie eine transaktionale E-Mail-Nachricht direkt an eine bestimmte E-Mail-Adresse senden können. Sie können diese Anfrage ändern, um Nachrichten über andere Kanäle wie SMS, Mobile Push oder Voice zu senden.

Um eine E-Mail-Nachricht zu senden, indem Sie eine SendMessages Anfrage einreichen

1. Stellen Sie sicher, dass der E-Mail-Kanal für das Projekt aktiviert ist und dass die E-Mail-Adresse oder Domain, die Sie zum Senden und Empfangen der Nachricht verwenden möchten, konfiguriert ist. Weitere Informationen finden Sie unter Aktivieren und Deaktivieren des E-Mail-Kanals und Verifizieren von E-Mail-Identitäten im Amazon-Pinpoint-Benutzerhandbuch.

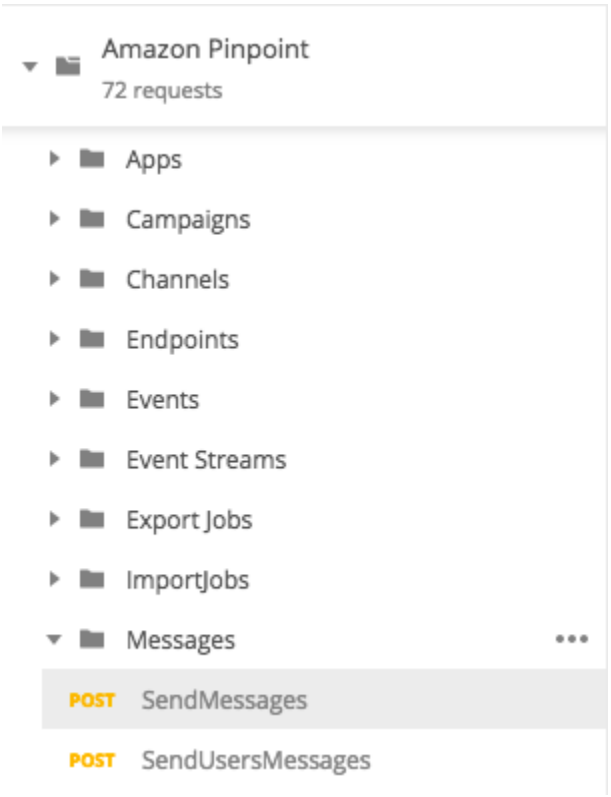
   > **Note**
   >
   > Um das Verfahren in diesem Abschnitt abzuschließen, müssen Sie eine E-Mail-Adresse verifizieren.

2. Wählen Sie im Menü Umgebungen die Umgebung aus AWS-Region , von der aus Sie die Nachricht senden möchten.

   In diesem Beispiel wurde Postman so konfiguriert, dass das Menü Environments (Umgebungen) die folgenden vier Optionen anzeigt:

   - USA Ost (Nord-Virginia)
   - USA West (Oregon)
   - Europe (Frankfurt)
   - Europa (Irland)

3. Wählen Sie im Ordner Nachrichten den SendMessagesVorgang aus.



4. Ersetzen Sie auf der Registerkarte Pre-Request Script (Skript vor Anfrage) den Wert der Variablen `projectId` durch die ID eines Projekts, das bereits in der Region existiert, die Sie in Schritt 2 dieses Abschnitts ausgewählt haben.
5. Löschen Sie auf der Registerkarte Body (Text) die Beispielanfrage, die im Anfrage-Editor angezeigt wird. Fügen Sie folgenden Code ein:

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

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

6. Ersetzen Sie im vorhergehenden Code *sender@example.com* durch Ihre verifizierte E-Mail-Adresse. Ersetzen Sie *recipient@example.com* durch die verifizierte E-Mail-Adresse, an die Sie die Nachricht senden möchten.

> **Note**
>
> Wenn sich Ihr Konto noch in der E-Mail-Sandbox-Umgebung von Amazon Pinpoint befindet, können Sie E-Mails nur an Adressen oder Domänen senden, die in Ihrem Amazon-Pinpoint-Konto verifiziert sind. Weitere Informationen zum Entfernen Ihres Kontos aus der Sandbox-Umgebung finden Sie unter Anfordern von Produktionszugriff für E-Mail im Amazon-Pinpoint-Benutzerhandbuch.

7. Wählen Sie Send (Senden) aus. Wenn die Nachricht erfolgreich gesendet wurde, zeigt der Antwortbereich den Status `200 OK` an.

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

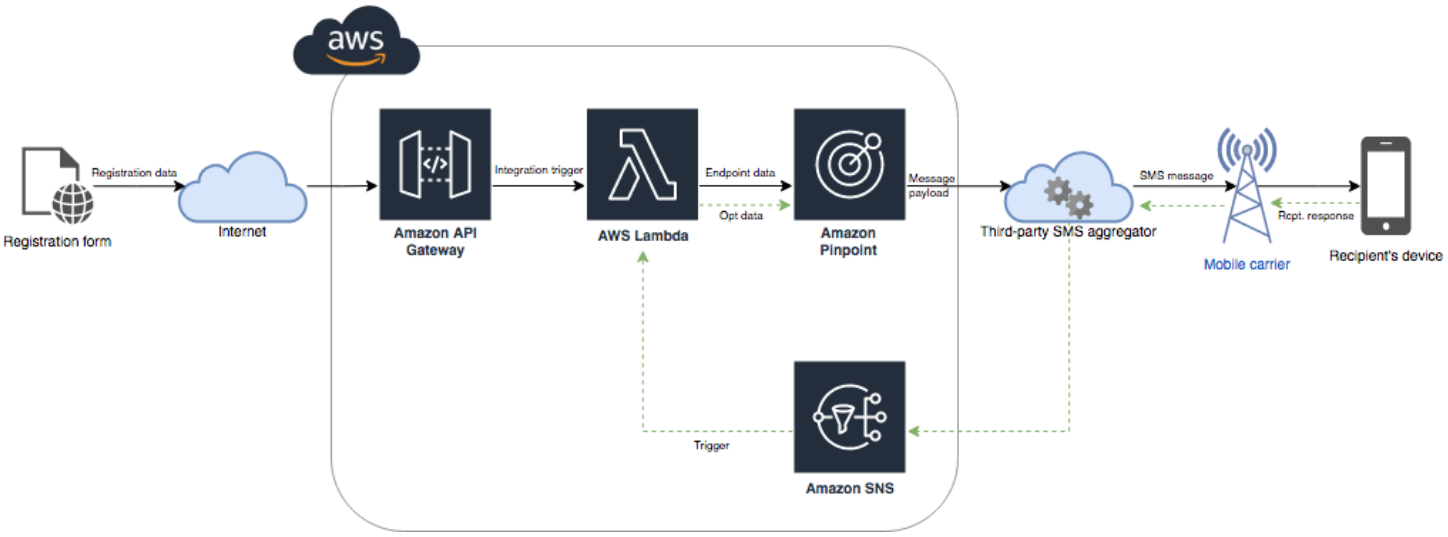
Tutorial: Einrichten eines SMS-Registrierungssystems

SMS-Nachrichten (Textnachrichten) sind eine gute Möglichkeit, zeitkritische Nachrichten an Ihre Kunden zu senden. Heutzutage haben viele Menschen ihr Telefon immer in der Nähe. Außerdem neigen Menschen dazu, SMS-Nachrichten mehr Aufmerksamkeit zu schenken als Push-Benachrichtigungen, E-Mails oder Anrufen.

Eine gängige Methode zur Erfassung der Mobiltelefonnummern von Kunden ist die Verwendung eines webbasierten Formulars. Nachdem Sie die Telefonnummer des Kunden überprüft und sein Abonnement bestätigt haben, können Sie damit beginnen, SMS-Nachrichten zu Werbe-, Transaktions- und Informationszwecken an diesen Kunden zu senden.

Dieses Tutorial zeigt Ihnen, wie Sie ein Webformular einrichten, um die Kontaktinformationen der Kunden zu erfassen. Das Webformular sendet diese Informationen an Amazon Pinpoint. Als Nächstes überprüft Amazon Pinpoint, ob die Telefonnummer gültig ist, und erfasst andere Metadaten, die sich auf die Telefonnummer beziehen. Danach sendet Amazon Pinpoint dem Kunden eine Nachricht, in der er aufgefordert wird, sein Abonnement zu bestätigen. Nachdem der Kunde sein Abonnement bestätigt hat, wählt Amazon Pinpoint ihn für den Empfang Ihrer Nachrichten aus.

Das folgende Architekturdiagramm zeigt den Datenfluss in dieser Lösung.



Informationen zu Double-Opt-in

Dieses Tutorial zeigt Ihnen, wie Sie ein Double-Opt-In-System in Amazon Pinpoint einrichten, das bidirektionale SMS-Nachrichten verwendet.

In einem SMS-Double-Opt-In-System stellt Ihnen ein Kunde seine Telefonnummer zur Verfügung, indem er sie in einem Webformular oder in Ihrer App übermittelt. Wenn Sie die Anfrage des Kunden erhalten, erstellen Sie einen neuen Endpunkt in Amazon Pinpoint. Der neue Endpunkt sollte von Ihrer Kommunikation ausgeschlossen werden. Als nächstes senden Sie eine Nachricht an diese Telefonnummer. In Ihrer Nachricht bitten Sie den Empfänger, sein Abonnement zu bestätigen, indem er mit einem bestimmten Wort oder Satz (z. B. „Ja“ oder „Bestätigen“) antwortet. Wenn der Kunde auf die Nachricht mit dem von Ihnen angegebenen Wort oder der von Ihnen angegebenen Phrase antwortet, ändern Sie den Status des Endpunkts in „Opted-in“. Andernfalls, wenn der Kunde nicht antwortet oder mit einem anderen Wort oder einer anderen Phrase antwortet, können Sie für den Endpunkt den Status „Opted-out“ bestehen lassen.

Informationen zu dieser Lösung

Dieser Abschnitt enthält Informationen über die Lösung, die Sie in diesem Tutorial erstellen.

Zielgruppe

Dieses Tutorial richtet sich an Entwickler und Systemimplementierer. Sie müssen nicht mit Amazon Pinpoint vertraut sein, um die Schritte in diesem Tutorial ausführen zu können. Sie sollten jedoch mit der Verwaltung von IAM-Richtlinien, der Erstellung von Lambda-Funktionen in Node.js und der Bereitstellung von Webinhalten vertraut sein.

Verwendete Funktionen

Dieses Tutorial enthält Anwendungsbeispiele für die folgenden Amazon-Pinpoint-Funktionen:

- Senden von transaktionalen SMS-Nachrichten
- Abrufen von Informationen über Telefonnummern durch die Verwendung der Telefonnummernüberprüfung
- Empfangen von eingehenden SMS-Nachrichten mithilfe von Zwei-Wege-SMS-Messaging
- Erstellen von dynamischen Segmenten
- Erstellen von Kampagnen
- Interaktion mit der Amazon Pinpoint API mithilfe von AWS Lambda

Benötigte Zeit

Bis zum Abschluss dieses Tutorials dauert es etwa eine Stunde. Nachdem Sie diese Lösung implementiert haben, können Sie weitere Schritte unternehmen, um die Lösung an Ihren individuellen Anwendungsfall anzupassen.

Regionale Einschränkungen

In diesem Tutorial müssen Sie mithilfe der Amazon-Pinpoint-Konsole eine Langwahlnummer leasen. Sie können die Amazon-Pinpoint-Konsole verwenden, um dedizierte Langwahlnummern zu leasen, die in mehreren Ländern registriert sind. Allerdings können nur Langwahlnummern verwendet werden, die in Kanada registriert sind, um SMS-Nachrichten zu senden. (Sie können Langwahlnummern, die in anderen Ländern und Regionen registriert sind, zum Senden von Sprachnachrichten verwenden.)

Wir haben die Codebeispiele in diesem Tutorial unter Berücksichtigung dieser Einschränkung entwickelt. Die Code-Beispiele gehen beispielsweise davon aus, dass die Telefonnummer des Empfängers immer 10 Ziffern und den Ländercode 1 hat. Wenn Sie diese Lösung in anderen Ländern oder Regionen als den USA oder Kanada implementieren, müssen Sie die Codebeispiele entsprechend anpassen.

Kosten der Ressourcennutzung

Für die Erstellung eines AWS Kontos fallen keine Gebühren an. Durch die Implementierung dieser Lösung können Ihnen jedoch folgende Kosten entstehen:

- Leasingkosten für Langwahlnummern: Um dieses Tutorial abzuschließen, müssen Sie eine Langwahlnummer leasen. Langwahlnummern, die in Kanada registriert sind, kosten 1,00 USD pro Monat.
- Nutzung der Telefonnummernüberprüfung: Die Lösung in diesem Tutorial verwendet das Feature der Telefonnummernüberprüfung von Amazon Pinpoint, um zu überprüfen, ob jede Nummer, die Sie erhalten, gültig und korrekt formatiert ist, und um zusätzliche Informationen über die Telefonnummer zu erhalten. Sie zahlen 0,006 USD für jede Anfrage zur Überprüfung der Telefonnummer.
- Kosten für den Versand von Nachrichten: Die Lösung in diesem Tutorial sendet ausgehende SMS-Nachrichten Sie zahlen für jede Nachricht, die Sie über Amazon Pinpoint senden. Der Preis, den Sie für jede Nachricht zahlen, hängt vom Land oder der Region des Empfängers ab. Wenn Sie Nachrichten an Empfänger in den USA (außer US-Territorien) senden, zahlen Sie 0,00645 USD pro Nachricht. Wenn Sie Nachrichten an Empfänger in Kanada senden, zahlen Sie 0,00109–0,02 USD, je nach Netzbetreiber und Standort des Empfängers.

- Kosten für den Empfang von Nachrichten: Diese Lösung empfängt und verarbeitet auch eingehende SMS-Nachrichten. Sie bezahlen für jede eingehende Nachricht, die an Telefonnummern gesendet wird, die mit Ihrem Amazon-Pinpoint-Konto verknüpft sind. Der Preis, den Sie zahlen, hängt davon ab, wo die empfangende Telefonnummer registriert ist. Wenn die empfangende Telefonnummer in den USA registriert ist (ausgenommen US-Territorien), zahlen Sie 0,0075 USD pro eingehende Nachricht. Wenn Ihre Nummer in Kanada registriert ist, zahlen Sie 0,00155 USD pro eingehende Nachricht.
- Lambda-Nutzung: Diese Lösung verwendet zwei Lambda-Funktionen, die mit der Amazon-Pinpoint-API interagieren. Wenn Sie eine Lambda-Funktion aufrufen, werden Ihnen die Kosten nach der Anzahl der Anforderungen für Ihre Funktionen, nach der Zeit, die Ihr Code benötigt, um ausgeführt zu werden, und nach der Menge an Speicher, die Ihre Funktionen verwenden, berechnet. Die Funktionen in diesem Tutorial verbrauchen sehr wenig Speicher und laufen typischerweise 1–3 Sekunden lang. Ihre Nutzung dieser Lösung fällt möglicherweise teilweise oder vollständig unter das kostenlose Lambda-Nutzungskontingent. Weitere Informationen finden Sie unter Lambda – Preise.
- API-Gateway-Nutzung: Das Webformular in dieser Lösung ruft eine API auf, die von API Gateway verwaltet wird Für jede Million Aufrufe an API Gateway zahlen Sie 3,50 bis 3,70 USD, je nachdem, in welcher AWS Region Sie Amazon Pinpoint verwenden. Weitere Informationen finden Sie unter API-Gateway-Preise.
- Webhostingkosten: Diese Lösung umfasst ein webbasiertes Formular, das Sie auf Ihrer Website hosten müssen. Die Höhe der Kosten für das Hosten dieser Inhalte hängt von Ihrem Webhosting-Anbieter ab.

> **Note**
>
> Alle in dieser Liste aufgeführten Preise sind in US-Dollar (USD) angegeben.

Nächster Schritt: Voraussetzungen

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Voraussetzungen erfüllen:

- Sie müssen über ein AWS-Konto verfügen. Um ein AWS-Konto zu erstellen, gehen Sie zu https://console.aws.amazon.com/ und wählen Sie Neues AWS-Konto erstellen aus.

- Das Konto, das Sie verwenden, um sich bei der AWS Management Console anzumelden, muss die folgenden Aufgaben ausführen können:
  - Neue IAM-Richtlinien und -Rollen erstellen
  - Neue Amazon-Pinpoint-Projekte erstellen
  - Neue Lambda-Funktionen erstellen
  - Neue APIs in API Gateway erstellen
- Sie müssen über eine Methode zum Hosting von Webseiten verfügen und Sie sollten wissen, wie Sie Webseiten veröffentlichen können. Obwohl Sie AWS-Services zum Hosten Ihrer Webseiten verwenden können, müssen Sie das nicht tun.

> **Tip**
>
> Weitere Informationen zum Hosten von Webseiten mithilfe von AWS-Services finden Sie unter Hosten einer statischen Webseite.

Weiter: Einrichten von Amazon Pinpoint

Schritt 1: Einrichten von Amazon Pinpoint

Der erste Schritt bei der Implementierung dieser Lösung ist die Einrichtung von Amazon Pinpoint. In diesem Abschnitt führen Sie folgenden Aufgaben aus:

- Erstellen Sie ein Amazon-Pinpoint-Projekt
- Aktivieren Sie den SMS-Kanal und leasen Sie eine Telefonnummer
- Konfigurieren Sie Zwei-Wege-SMS-Messaging

Bevor Sie mit diesem Tutorial beginnen, sollten Sie die Voraussetzungen überprüfen.

Erstellen Sie ein Amazon-Pinpoint-Projekt

Um zu beginnen, müssen Sie ein Amazon-Pinpoint-Projekt erstellen. In Amazon Pinpoint besteht ein Projekt aus Segmenten, Kampagnen, Konfigurationen und Daten, die durch einen gemeinsamen Zweck miteinander verbunden sind. Beispielsweise könnten Sie in einem Projekt alle Inhalte vereinen, die sich auf eine bestimmte App oder eine bestimmte Marke oder Marketinginitiative beziehen. Wenn Sie Kundeninformationen zu Amazon Pinpoint hinzufügen, werden diese Informationen einem Projekt zugeordnet.

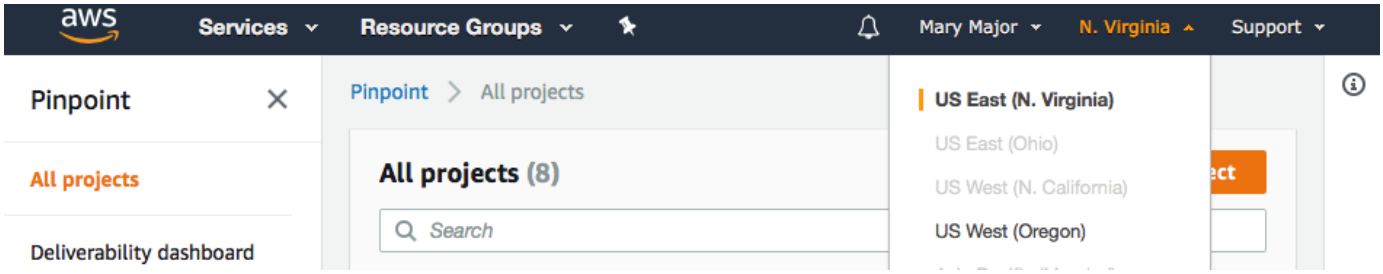
Die Schritte zum Erstellen eines neuen Projekts unterscheiden sich je nachdem, ob Sie zuvor ein Projekt in Amazon Pinpoint erstellt haben.

Erstellen eines Projekts (neue Amazon-Pinpoint-Benutzer)

Diese Schritte beschreiben den Prozess der Erstellung eines neuen Amazon Pinpoint Pinpoint-Projekts, falls Sie in der aktuellen AWS Region noch nie ein Projekt erstellt haben.

Ein Projekt erstellen

1. Melden Sie sich bei der Amazon Pinpoint Pinpoint-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/pinpoint/.
2. Verwenden Sie die Regionsauswahl, um die AWS Region auszuwählen, die Sie verwenden möchten, wie in der folgenden Abbildung gezeigt. Wenn Sie sich nicht sicher sind, wählen Sie die Region aus, die Ihnen am nächsten liegt.



3. Geben Sie unter Get started (Erste Schritte) unter Name einen Namen für die Kampagne ein (z. B. **`SMSRegistration`**) und wählen Sie dann Create project (Projekt erstellen) aus.
4. Wählen Sie auf der Seite Configure features (Funktionen konfigurieren) die Option Skip this step (Diesen Schritt überspringen) aus.
5. Wählen Sie im Navigationsbereich die Option Alle Projekte (Alle Projekte) aus.
6. Kopieren Sie auf der Seite All projects (Alle Projekte) neben dem Projekt, das Sie gerade erstellt haben, den Wert, der in der Spalte Project ID (Projekt-ID) angezeigt wird.

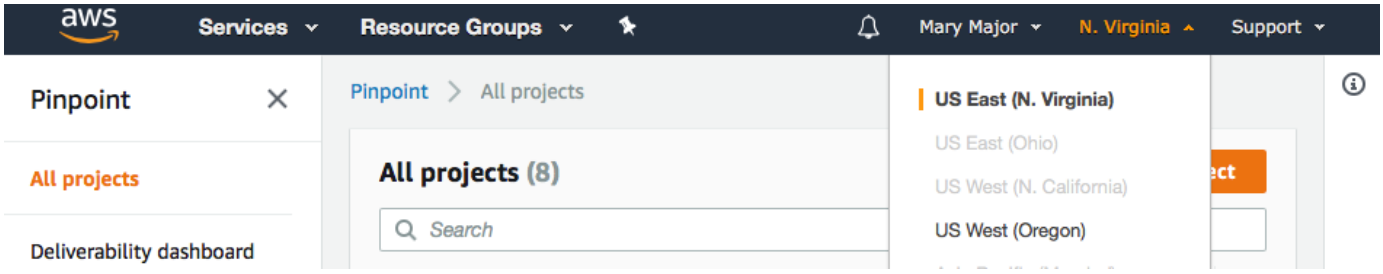
> **Tip**
>
> Sie müssen diese ID an verschiedenen Stellen in diesem Tutorial verwenden. Bewahren Sie die Projekt-ID an einem geeigneten Ort auf, sodass Sie sie später kopieren können.

Erstellen eines Projekts (bestehende Amazon-Pinpoint-Benutzer)

Diese Schritte beschreiben den Prozess der Erstellung eines neuen Amazon Pinpoint Pinpoint-Projekts, falls Sie bereits Projekte in der aktuellen AWS Region erstellt haben.

Ein Projekt erstellen

1. Melden Sie sich bei der Amazon Pinpoint Pinpoint-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/pinpoint/.
2. Verwenden Sie die Regionsauswahl, um die AWS Region auszuwählen, die Sie verwenden möchten, wie in der folgenden Abbildung gezeigt. Wenn Sie sich nicht sicher sind, wählen Sie die Region aus, die Ihnen am nächsten liegt.



3. Klicken Sie auf der Seite Alle Projekte auf Projekt erstellen.
4. Geben Sie im Fenster Create a project (Erstellen eines Projekts) unter Project name (Projektname) einen Namen für das Projekt ein (z. B. **`SMSRegistration`**). Wählen Sie Erstellen.
5. Wählen Sie auf der Seite Configure features (Funktionen konfigurieren) die Option Skip this step (Diesen Schritt überspringen) aus.
6. Wählen Sie im Navigationsbereich die Option Alle Projekte (Alle Projekte) aus.
7. Kopieren Sie auf der Seite All projects (Alle Projekte) neben dem Projekt, das Sie gerade erstellt haben, den Wert, der in der Spalte Project ID (Projekt-ID) angezeigt wird.

> **Tip**
>
> Sie müssen diese ID an verschiedenen Stellen in diesem Tutorial verwenden. Bewahren Sie die Projekt-ID an einem geeigneten Ort auf, sodass Sie sie später kopieren können.

Besorgen Sie sich eine dedizierte Telefonnummer

> **Note**
>
> Amazon Pinpoint hat die Dokumentation seines Benutzerhandbuchs aktualisiert. Aktuelle Informationen zur Erstellung, Konfiguration und Verwaltung Ihrer Amazon-Pinpoint-SMS- und -Sprachressourcen finden Sie im neuen Amazon-Pinpoint-SMS-Benutzerhandbuch.

Nachdem Sie ein Projekt erstellt haben, können Sie damit beginnen, Funktionen in diesem Projekt zu konfigurieren. In diesem Abschnitt aktivieren Sie den SMS-Kanal und rufen eine dedizierte Telefonnummer ab, die Sie beim Senden von SMS-Nachrichten verwenden können.

> **Note**
>
> In diesem Abschnitt wird davon ausgegangen, dass Sie nach der Registrierung der Marke und Kampagne eine 10DLC-Telefonnummer in den USA, eine gebührenfreie Nummer in den USA oder eine kanadische Langwahlnummer leasen. Wenn Sie die in diesem Abschnitt beschriebenen Verfahren befolgen, aber ein anderes Land als die Vereinigten Staaten oder Kanada auswählen, können Sie diese Nummer nicht verwenden, um SMS-Nachrichten zu versenden. Weitere Informationen zum Leasing von SMS-fähigen Langcodes in anderen Ländern als den USA oder Kanada finden Sie unter Unterstützte Länder und Regionen (SMS-Kanal) im Amazon-Pinpoint-SMS-Benutzerhandbuch.

Gehen Sie wie folgt vor, um den SMS-Kanal über die Amazon-Pinpoint-Konsole zu aktivieren:

SMS-Kanal aktivieren

1. Melden Sie sich bei der Amazon Pinpoint Pinpoint-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/pinpoint/.
2. Klicken Sie im Navigationsbereich unter Settings (Einstellungen) auf SMS and voice (SMS und Stimme).
3. Klicken Sie neben SMS settings (SMS-Einstellungen) auf Edit (Bearbeiten).
4. Wählen Sie unter General settings (Allgemeine Einstellungen) die Option Enable the SMS channel for this project (Den SMS-Kanal für dieses Projekt aktivieren) und anschließend Save changes (Änderungen speichern) aus.

Gehen Sie zum Anfordern einer Telefonnummer mit der Amazon-Pinpoint-Konsole wie folgt vor:

Anfordern einer Telefonnummer (Konsole)

1. Öffnen Sie die Amazon-Pinpoint-SMS-Konsole unter https://console.aws.amazon.com/sms-voice/.

   > **Note**
   >
   > Stellen Sie sicher, dass Sie eine Telefonnummer genauso anfordern AWS-Region , in der Sie Ihr Amazon Pinpoint Pinpoint-Projekt erstellt haben.

2. Wählen Sie im Navigationsbereich unter Konfigurationen die Option Telefonnummern und dann Absender anfordern aus.
3. Wählen Sie auf der Seite Land auswählen für das Land des Nachrichtenziels entweder die USA oder Kanada aus. Wählen Sie Weiter aus.
4. Geben Sie im Abschnitt Anwendungsfall für Messaging Folgendes ein:

   - Wählen Sie unter Rufnummernfunktionen die Option SMS

     > **Important**
     >
     > Die Funktionen für SMS und Sprache können nach dem Kauf der Telefonnummer nicht mehr geändert werden.

   - Wählen Sie für Bidirektionales Messaging Ja.
5. Wählen Sie Weiter aus.
6. Wählen Sie unter Absendertyp auswählen entweder Long Code oder 10DLC aus.

   Wenn Sie 10DLC auswählen und bereits eine registrierte Kampagne haben, können Sie die Kampagne unter Mit registrierter Kampagne verbinden auswählen.
7. Wählen Sie Weiter aus.
8. Unter Überprüfung und Anfrage können Sie Ihre Afrage überprüfen und bearbeiten, bevor Sie sie absenden. Wählen Sie Request (Anfrage).
9. Abhängig von der Art der von Ihnen angeforderten Telefonnummer wird möglicherweise das Fenster Registrierung erforderlich angezeigt. Ihre Telefonnummer ist mit dieser Registrierung

verbunden und kann erst dann Nachrichten senden, wenn Ihre Registrierung genehmigt wurde. Weitere Informationen zu den Registrierungsanforderungen finden Sie unter Registrierungen.

a. Geben Sie unter Name des Anmeldeformulars einen benutzerfreundlichen Namen ein.

b. Wählen Sie Registrierung beginnen, um die Registrierung der Telefonnummer abzuschließen, oder Später registrieren.

> **Important**
>
> Ihre Telefonnummer kann erst dann Nachrichten senden, wenn Ihre Registrierung genehmigt wurde.
> Ihnen wird weiterhin die wiederkehrende monatliche Leasinggebühr für die Telefonnummer in Rechnung gestellt, unabhängig vom Registrierungsstatus.
> Weitere Informationen zu den Registrierungsanforderungen finden Sie unter Registrierungen.

Aktivieren bidirektionaler SMS-Nachrichten

Jetzt, da Sie eine eigene Telefonnummer haben, können Sie Zwei-Wege-SMS einrichten. Die Aktivierung von Zwei-Wege-SMS ermöglicht es Ihren Kunden, auf die SMS-Nachrichten zu antworten, die Sie ihnen senden. In dieser Lösung verwenden Sie Zwei-Wege-SMS, um Ihren Kunden eine Möglichkeit zu geben, zu bestätigen, dass sie Ihr SMS-Programm abonnieren möchten.

Gehen Sie wie folgt vor, um bidirektionale SMS über die Amazon-Pinpoint-SMS-Konsole zu aktivieren:

Aktivieren bidirektionaler SMS-Nachrichten

1. Öffnen Sie die Amazon-Pinpoint-SMS-Konsole unter https://console.aws.amazon.com/sms-voice/.
2. Wählen Sie im Navigationsbereich unter Konfigurationen die Option Telefonnummern.
3. Wählen Sie auf der Seite Telefonnummern verwalten eine Telefonnummer.
4. Wählen Sie auf der Registerkarte Bidirektionale SMS die Schaltfläche Einstellungen bearbeiten.
5. Wählen Sie auf der Seite Einstellungen bearbeiten die Option Bidirektionale Nachricht aktivieren aus.
6. Wählen Sie für Zielart Amazon SNS aus.

- Neues Amazon-SNS-Thema – Amazon Pinpoint SMS erstellt ein Thema in Ihrem Konto. Das Thema wird automatisch mit allen erforderlichen Berechtigungen erstellt. Weitere Informationen zu Amazon-SNS-Themen finden Sie unter Konfigurieren von Amazon SNS im Amazon-SNS-Entwicklerhandbuch.
- Geben Sie unter Ziel für eingehende Nachrichten einen Themennamen ein, z. B. **`SMSRegistrationFormTopic`**.

7. Wählen Sie für Bidirektionale Kanalrolle die Option SNS-Themenrichtlinien verwenden aus.
8. Wählen Sie Änderungen speichern aus.

Verwenden Sie die Amazon-Pinpoint-SMS-Konsole, um Ihrer Telefonnummer Stichwörter hinzuzufügen, die Ihnen Kunden zur Bestätigung ihrer Abonnements senden (z. B. **`Yes`** oder **`Confirm`**).

Hinzufügen eines Schlüsselworts

1. Öffnen Sie die Amazon-Pinpoint-SMS-Konsole unter https://console.aws.amazon.com/sms-voice/.
2. Wählen Sie im Navigationsbereich unter Konfigurationen die Option Telefonnummer.
3. Wählen Sie auf der Seite Telefonnummern die Telefonnummer aus, der ein Stichwort hinzugefügt werden soll.
4. Wählen Sie auf der Registerkarte Stichwörter die Schaltfläche Stichwort hinzufügen.
5. Fügen Sie im Bereich Benutzerdefiniertes Stichwort Folgendes hinzu:
   - Stichwort – Das neue Stichwort, das hinzugefügt werden soll (z. B. **`Yes`** oder **`Confirm`**).
   - Antwortnachricht – Die Nachricht, die an den Empfänger zurückgesendet werden soll.
   - Stichwortaktion – Die Aktion, die ausgeführt werden soll, wenn das Stichwort empfangen wird. Wählen Sie Automatische Antwort.
6. Wählen Sie Stichwort hinzufügen.

Weiter: Erstellen von IAM-Richtlinien und -Rollen

Schritt 2: Erstellen von IAM-Richtlinien und -Rollen

Der nächste Schritt bei der Implementierung der SMS-Registrierungslösung besteht darin, eine Richtlinie und eine Rolle in AWS Identity and Access Management (IAM) zu konfigurieren. Für diese

Lösung müssen Sie eine Richtlinie erstellen, die den Zugriff auf bestimmte Ressourcen ermöglicht, die sich auf Amazon Pinpoint beziehen. Anschließend erstellen Sie eine Rolle und fügen die Richtlinie daran an. Später in diesem Tutorial erstellen Sie eine AWS Lambda Funktion, die diese Rolle verwendet, um bestimmte Operationen in der Amazon Pinpoint Pinpoint-API aufzurufen.

Eine IAM-Richtlinie erstellen

In diesem Abschnitt wird beschrieben, wie Sie eine IAM-Richtlinie erstellen. Benutzer und Rollen, die diese Richtlinie verwenden, können Folgendes tun:

- Die Funktion zur Telefonnummernüberprüfung verwenden
- Amazon-Pinpoint-Endpunkte anzeigen, erstellen und aktualisieren
- Nachrichten an Endpunkte von Amazon Pinpoint senden

In diesem Tutorial möchten Sie Lambda die Möglichkeit geben, diese Aufgaben auszuführen. Allerdings verwendet diese Richtlinie, um die Sicherheit zu erhöhen, das Prinzipal des Erteilens von geringsten Rechten. Mit anderen Worten, es werden nur die Berechtigungen erteilt, die zum Abschließen dieser Lösung erforderlich sind, und nicht mehr. Diese Richtlinie ist wie folgt eingeschränkt:

- Sie können sie nur verwenden, um die Telefonnummernüberprüfungs-API in einer bestimmten Region aufzurufen.
- Sie können sie nur verwenden, um Endpunkte anzuzeigen, zu erstellen oder zu aktualisieren, die mit einem bestimmten Amazon-Pinpoint-Projekt verknüpft sind.
- Sie können sie nur verwenden, um Nachrichten an Endpunkte zu senden, die mit einem bestimmten Amazon-Pinpoint-Projekt verknüpft sind.

So erstellen Sie die Richtlinie

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen).
3. Fügen Sie auf der Registerkarte JSON den folgenden Code ein.

```
{
    "Version": "2012-10-17",
```

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

Gehen Sie im vorhergehenden Beispiel wie folgt vor:

- Ersetzen Sie *Region* durch die AWS Region, in der Sie Amazon Pinpoint verwenden, z. B. `us-east-1` oder `eu-central-1`.

> **Tip**
>
> Eine vollständige Liste der AWS Regionen, in denen Amazon Pinpoint verfügbar ist, finden Sie unter AWS Regionen und Endpunkte in der. Allgemeine AWS-Referenz

- Ersetzen Sie *`accountId`* durch die eindeutige ID für Ihr AWS Konto.
- Ersetzen Sie *`ProjectID`* durch die eindeutige ID des Projekts, das Sie in Erstellen Sie ein Amazon Pinpoint Pinpoint-Projekt dieses Tutorials erstellt haben.

> **Note**
>
> Die `logs` Aktionen ermöglichen es Lambda, seine Ausgabe in Logs zu CloudWatch protokollieren.

4. Wählen Sie Weiter aus.
5. Geben Sie unter Richtlinienname einen Namen für die Richtlinie ein, z. B. **`RegistrationFormPolicy`** Wählen Sie Richtlinie erstellen aus.

Erstellen einer IAM-Rolle

So erstellen Sie die Rolle

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Rollen und wählen Sie dann Rolle erstellen aus.
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS Service und dann für Service oder Benutzerfall Lambda aus der Drop-down-Liste aus.
4. Wählen Sie Weiter aus.
5. Wählen Sie unter Berechtigungsrichtlinien die Richtlinie aus, die Sie im vorherigen Abschnitt erstellt haben, oder suchen Sie danach, und klicken Sie dann auf Weiter.
6. Geben Sie unter Rollendetails für Rollenname einen Namen für die Rolle ein, z. **`SMSRegistrationForm`** B. Wählen Sie Rolle erstellen aus.

Weiter: Erstellen von Lambda-Funktionen

Schritt 3: Erstellen von Lambda-Funktionen

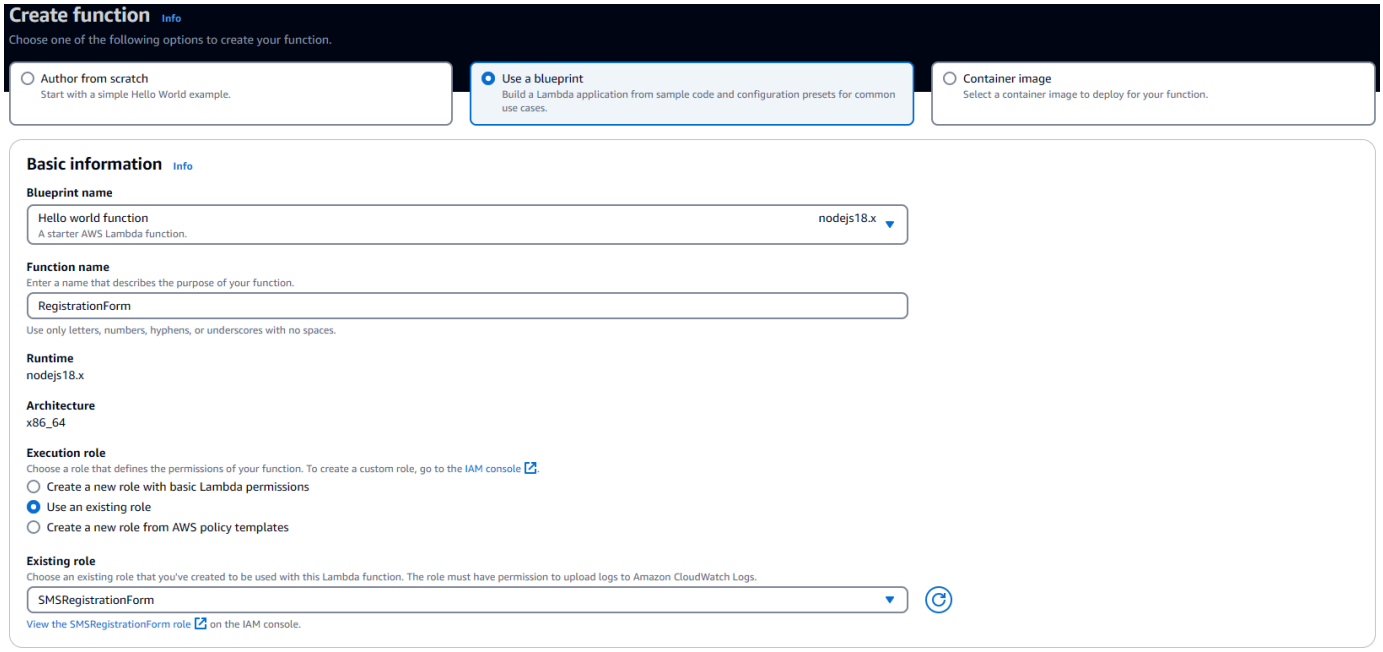
Diese Lösung verwendet zwei Lambda-Funktionen. In diesem Abschnitt erfahren Sie, wie Sie diese Funktionen anlegen und konfigurieren. Später richten Sie API Gateway und Amazon Pinpoint ein, um diese Funktionen auszuführen, wenn bestimmte Ereignisse eintreten. Beide Funktionen erstellen und aktualisieren Endpunkte im von Ihnen angegebenen Amazon-Pinpoint-Projekt. Die erste Funktion verwendet auch die Funktion zur Telefonnummernüberprüfung.

Erstellen der Funktion, die Kundendaten überprüft und Endpunkte erstellt

Die erste Funktion verwendet Eingaben aus Ihrem Registrierungsformular, die sie von Amazon API Gateway erhält. Sie verwendet diese Informationen, um mithilfe der Funktion zur Überprüfung der Telefonnummer von Amazon Pinpoint Informationen über die Telefonnummer des Kunden zu erhalten. Die Funktion verwendet dann die validierten Daten, um einen neuen Endpunkt in dem von Ihnen angegebenen Amazon-Pinpoint-Projekt zu erstellen. Standardmäßig wird der Endpunkt, den die Funktion erstellt, von zukünftigen Nachrichten von Ihnen ausgeschlossen, aber dieser Status kann durch die zweite Funktion geändert werden. Schließlich sendet diese Funktion dem Kunden eine Nachricht, in der er aufgefordert wird, zu überprüfen, ob er SMS-Nachrichten von Ihnen erhalten möchte.

So erstellen Sie die Lambda-Funktion:

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie Funktion erstellen.
3. Wählen Sie unter Funktion erstellen die Option Blueprint verwenden aus.
4. Geben Sie im Suchfeld **`hello`** ein und drücken Sie dann die Eingabetaste. Wählen Sie in der Ergebnisliste die Node.js-Funktion `hello-world` aus, wie im folgenden Bild gezeigt.



5. Führen Sie unter Basic information (Grundlegende Informationen) die folgenden Schritte aus:

   - Geben Sie für Name einen Namen für die Funktion ein, z. B. **`RegistrationForm`**.
   - Wählen Sie für Role (Rolle) die Option Choose an existing role (Eine vorhandene Rolle wählen) aus.
   - Wählen Sie für Bestehende Rolle die RegistrationFormSMS-Rolle aus, die Sie unter IAM-Rolle erstellen erstellt haben.

   Wenn Sie fertig sind, klicken Sie auf Create function (Funktion erstellen).

6. Löschen Sie unter Codequelle die Beispielfunktion im Code-Editor und fügen Sie dann den folgenden Code ein:

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

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

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

```
var originationNumber = process.env.originationNumber;

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

var messageType = "TRANSACTIONAL";

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

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

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

async function createEndpoint(data, firstName, lastName, source) {
```

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

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

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

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

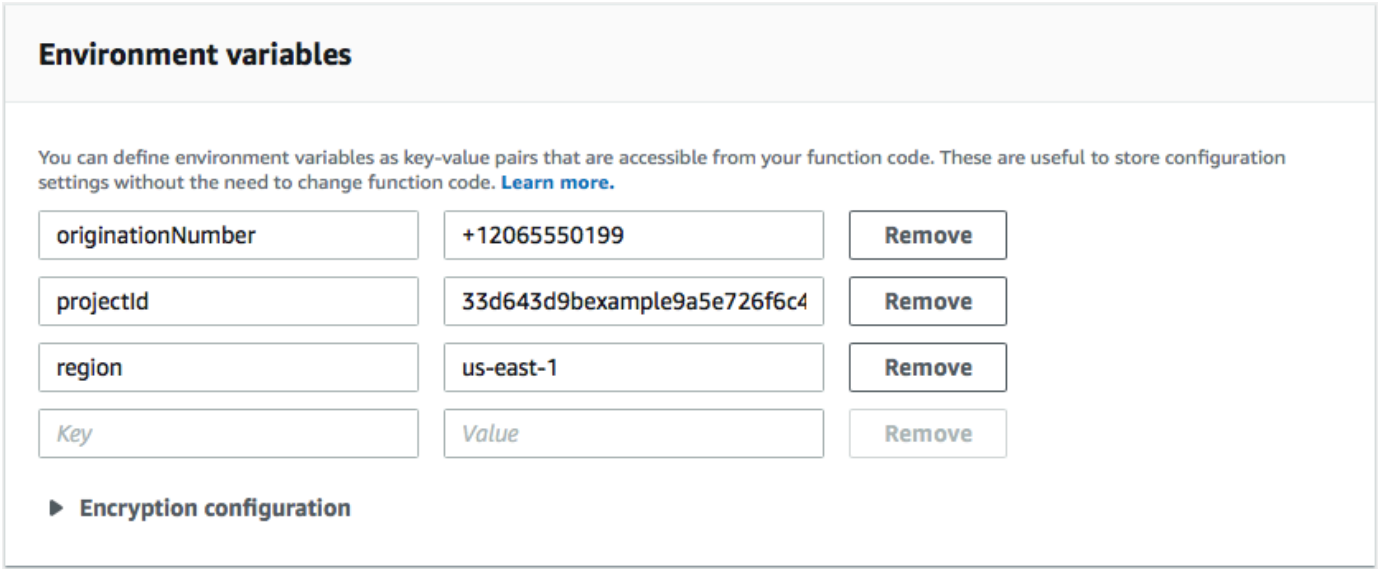
7. Wählen Sie auf der Registerkarte Konfiguration für Umgebungsvariablen die Option Bearbeiten und dann Umgebungsvariable hinzufügen aus. Gehen Sie wie folgt vor:

- Erstellen Sie in der ersten Zeile eine Variable mit dem Schlüssel **`originationNumber`**. Stellen Sie anschließend den Wert auf die Telefonnummer der dedizierten Langwahlnummer ein, die Sie in Schritt 1.2 erhalten haben.

  > **Note**
  >
  > Achten Sie darauf, dass Sie das Pluszeichen (+) und die Landesvorwahl für die Telefonnummer angeben. Fügen Sie keine weiteren Sonderzeichen wie Bindestriche (-), Punkte (.) oder Klammern hinzu.

- Erstellen Sie in der zweiten Zeile eine Variable mit dem Schlüssel **`projectId`**. Als nächstes setzen Sie den Wert auf die eindeutige ID des Projekts, das Sie in Schritt 1.1 erstellt haben.
- Erstellen Sie in der dritten Zeile eine Variable mit einem Schlüssel von **`region`**. Stellen Sie als Nächstes den Wert auf die Region ein, in der Sie Amazon Pinpoint verwenden, z. B. **`us-east-1`** oder **`us-west-2`**.

Wenn Sie fertig sind, sollte der Abschnitt Environment Variables (Umgebungsvariablen) dem in der folgenden Abbildung gezeigten Beispiel entsprechen.



8. Wählen Sie oben auf der Seite Save aus.

Testen der Funktion

Nachdem Sie die Funktion erstellt haben, sollten Sie sie testen, um sich zu vergewissern, dass sie richtig konfiguriert ist. Stellen Sie außerdem sicher, dass die von Ihnen erstellte IAM-Rolle über die entsprechenden Berechtigungen verfügt.

So testen Sie die -Funktion

1. Wählen Sie die Registerkarte Test.
2. Wählen Sie Neues Ereignis erstellen und gehen Sie wie folgt vor:
   - Geben Sie für Event name (Ereignisname) einen Namen für das Testereignis ein, z. B. **`MyPhoneNumber`**.
   - Löschen Sie den Beispielcode im Code-Editor. Fügen Sie folgenden Code ein:

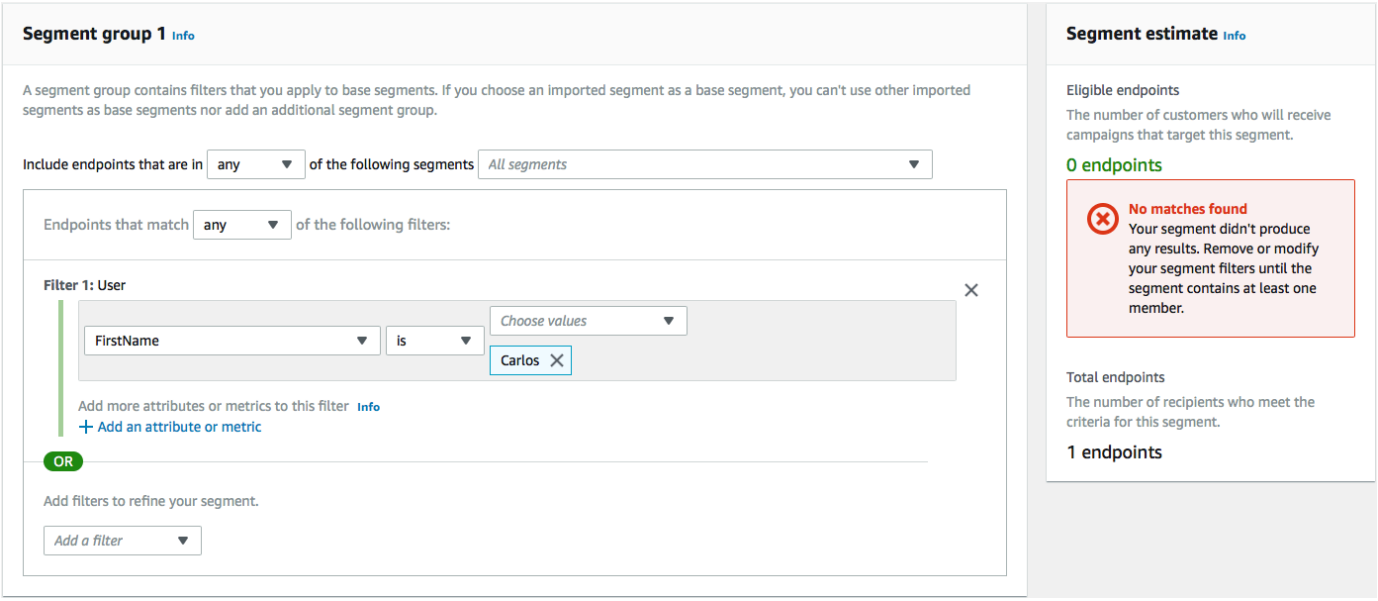
     ```
     {
       "destinationNumber": "+12065550142",
       "firstName": "Carlos",
       "lastName": "Salazar",
       "source": "Registration form test"
     }
     ```

   - Ersetzen Sie im vorhergehenden Codebeispiel die Werte der Attribute `destinationNumber`, `firstName` und `lastName` durch die Werte, die Sie zum Testen verwenden möchten, wie beispielsweise Ihre persönlichen Kontaktdaten. Wenn Sie diese Funktion testen, sendet sie eine SMS an die Telefonnummer, die Sie im Attribut `destinationNumber` angeben. Stellen Sie sicher, dass die von Ihnen angegebene Telefonnummer in der Lage ist, SMS-Nachrichten zu empfangen.
   - Wählen Sie Erstellen.
3. Wählen Sie Test aus.
4. Wählen Sie unter Execution result: succeeded (Ausführungsergebnis: erfolgreich) die Option Details aus. Überprüfen Sie im Abschnitt Log output (Protokollausgabe) die Ausgabe der Funktion. Stellen Sie sicher, dass die Funktion fehlerfrei ausgeführt wurde.

   Überprüfen Sie das Gerät, das der von Ihnen angegebenen `destinationNumber` zugeordnet ist, um sicherzustellen, dass es die Testnachricht empfangen hat.
5. Öffnen Sie die Amazon Pinpoint-Konsole unter https://console.aws.amazon.com/pinpoint/.

6. Wählen Sie auf der Seite Alle Projekte das Projekt aus, das Sie unter Amazon Pinpoint Pinpoint-Projekt erstellen erstellt haben.
7. Wählen Sie im Navigationsbereich die Option Segments (Segmente) aus. Wählen Sie auf der Seite Segments (Segmente) die Option Create a segment (Ein Segment erstellen) aus.
8. Wählen Sie in Segment group 1 (Segmentgruppe 1) unter Add filters to refine your segment (Filter zur Optimierung des Segments hinzufügen die Option Filter by user (Nach Benutzer filtern) aus.
9. Wählen Sie unter Benutzerattribut auswählen die Option FirstName. Wählen Sie dann unter Choose values (Werte auswählen) den Vornamen aus, den Sie im Testereignis angegeben haben.

   Im Abschnitt Segment estimate (Segmentschätzung) sollte angezeigt werden, dass es keine qualifizierten Endpunkte und insgesamt einen Endpunkt gibt, wie in der folgenden Abbildung gezeigt. Dieses Ergebnis wird erwartet. Wenn die Funktion einen neuen Endpunkt erstellt, ist der Endpunkt deaktiviert. Segmente in Amazon Pinpoint schließen automatisch deaktivierte Endpunkte aus.



Erstellen der Funktion, die Kunden für Ihre Kommunikation aktiviert

Die zweite Funktion wird nur ausgeführt, wenn ein Kunde auf die Nachricht antwortet, die von der ersten Funktion gesendet wurde. Wenn die Antwort des Kunden das Schlüsselwort enthält, das Sie unter Bidirektionale SMS aktivieren angegeben haben, aktualisiert die Funktion seinen

Endpunktdatensatz, um ihn für future Mitteilungen anzumelden. Amazon Pinpoint antwortet auch automatisch mit der Nachricht, die Sie unter Bidirektionale SMS aktivieren angegeben haben.

Wenn der Kunde nicht antwortet oder seine Antwort das angegebene Schlüsselwort nicht enthält, dann passiert nichts. Der Endpunkt des Kunden bleibt in Amazon Pinpoint, aber Segmente können ihn nicht anvisieren.

So erstellen Sie die Lambda-Funktion:

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie Funktion erstellen.
3. Wählen Sie unter Create function (Funktion erstellen) die Option Blueprints aus.
4. Geben Sie im Suchfeld **`hello`** ein und drücken Sie dann die Eingabetaste. Wählen Sie in der Ergebnisliste die Node.js-Funktion `hello-world` aus, wie im folgenden Bild gezeigt. Wählen Sie Konfigurieren aus.
5. Führen Sie unter Basic information (Grundlegende Informationen) die folgenden Schritte aus:

   - Geben Sie für Name einen Namen für die Funktion ein, z. B. **`RegistrationForm_OptIn`**.
   - Wählen Sie für Role (Rolle) die Option Choose an existing role (Eine vorhandene Rolle wählen) aus.
   - Wählen Sie unter Existierende Rolle die RegistrationForm SMS-Rolle aus, die Sie unter IAM-Rolle erstellen erstellt haben.

   Wenn Sie fertig sind, klicken Sie auf Create function (Funktion erstellen).
6. Löschen Sie die Beispielfunktion im Code-Editor und fügen Sie dann den folgenden Code ein:

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

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

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

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

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

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

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

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

    try {
```

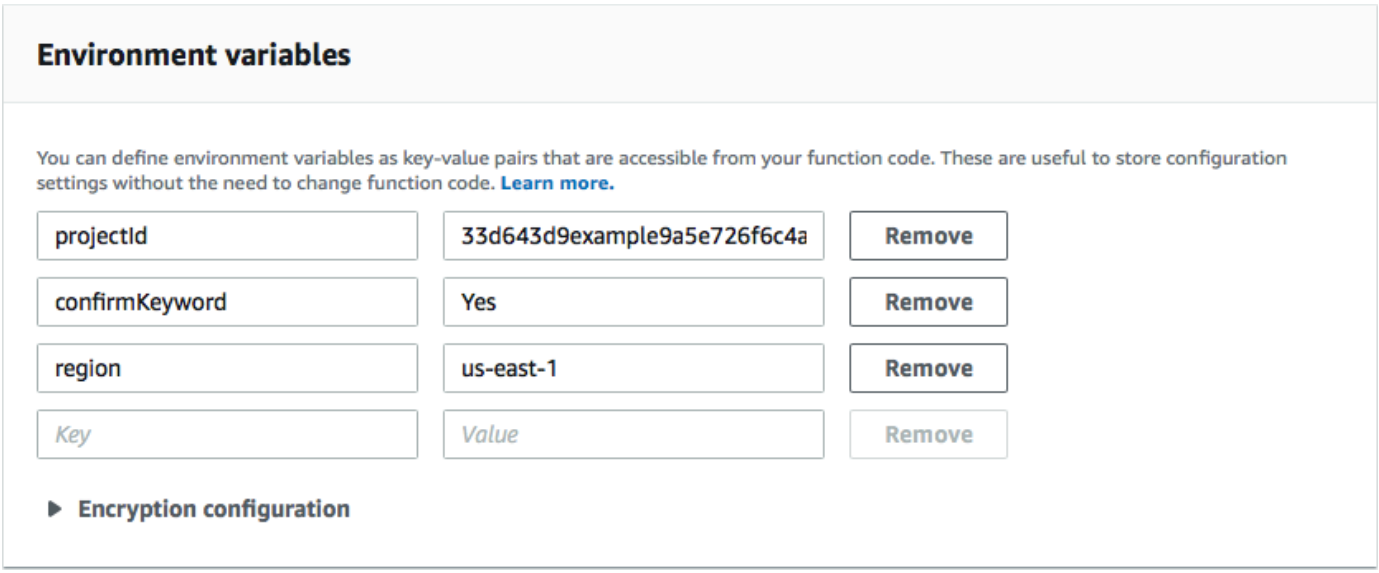
```
        // Send the UpdateEndpointCommand to update the endpoint's opt-in status
        const updateEndpointResponse = await pinClient.send(new
 UpdateEndpointCommand(params));
        console.log(updateEndpointResponse);
        console.log(`Successfully changed the opt status of endpoint ID
 ${endpointId}`);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

7. Gehen Sie unter Environment variables (Umgebungsvariablen) wie folgt vor:

   - Erstellen Sie in der ersten Zeile eine Variable mit dem Schlüssel **`projectId`**. Als Nächstes setzen Sie den Wert auf die eindeutige ID des Projekts, das Sie unter Amazon Pinpoint Pinpoint-Projekt erstellen erstellt haben.
   - Erstellen Sie in der zweiten Zeile eine Variable mit dem Schlüssel **`region`**. Stellen Sie als Nächstes den Wert auf die Region ein, in der Sie Amazon Pinpoint verwenden, z. B. **`us-east-1`** oder **`us-west-2`**.
   - Erstellen Sie in der dritten Zeile eine Variable mit einem Schlüssel von **`confirmKeyword`**. Stellen Sie als Nächstes den Wert auf das Bestätigungsschlüsselwort ein, das Sie unter Bidirektionale SMS aktivieren erstellt haben.

     > **Note**
     >
     > Das Schlüsselwort wird nicht nach Groß- und Kleinschreibung unterschieden. Diese Funktion wandelt die eingehende Nachricht in Kleinbuchstaben um.

   Wenn Sie fertig sind, sollte der Abschnitt Environment Variables (Umgebungsvariablen) dem in der folgenden Abbildung gezeigten Beispiel entsprechen.



8. Wählen Sie oben auf der Seite Save aus.

Testen der Funktion

Nachdem Sie die Funktion erstellt haben, sollten Sie sie testen, um sich zu vergewissern, dass sie richtig konfiguriert ist. Stellen Sie außerdem sicher, dass die von Ihnen erstellte IAM-Rolle über die entsprechenden Berechtigungen verfügt.

So testen Sie die -Funktion

1. Wählen Sie Test aus.
2. Führen Sie im Fenster Configure test event (Testereignis konfigurieren) die folgenden Schritte aus:

   a. Wählen Sie Create new test event aus.

   b. Geben Sie für Event name (Ereignisname) einen Namen für das Testereignis ein, z. B. **`MyResponse`**.

   c. Löschen Sie den Beispielcode im Code-Editor. Fügen Sie folgenden Code ein:

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

```
      }
    }
  ]
}
```

Ersetzen Sie im vorigen Codebeispiel die Werte des Attributs `originationNumber` durch die Telefonnummer, die Sie verwendet haben, als Sie die vorherige Lambda-Funktion getestet haben. Ersetzen Sie den Wert von `messageBody` durch das bidirektionale SMS-Schlüsselwort, das Sie unter Bidirektionale SMS aktivieren angegeben haben. Optional können Sie den Wert von `Timestamp` durch das aktuelle Datum und die aktuelle Uhrzeit ersetzen.

d. Wählen Sie Erstellen.

3. Wählen Sie erneut Test (Testen) aus.
4. Wählen Sie unter Execution result: succeeded (Ausführungsergebnis: erfolgreich) die Option Details aus. Überprüfen Sie im Abschnitt Log output (Protokollausgabe) die Ausgabe der Funktion. Stellen Sie sicher, dass die Funktion fehlerfrei ausgeführt wurde.
5. Öffnen Sie die Amazon Pinpoint-Konsole unter https://console.aws.amazon.com/pinpoint/.
6. Wählen Sie auf der Seite Alle Projekte das Projekt aus, das Sie unter Amazon Pinpoint Pinpoint-Projekt erstellen erstellt haben.
7. Wählen Sie im Navigationsbereich die Option Segments (Segmente) aus. Wählen Sie auf der Seite Segments (Segmente) die Option Create a segment (Ein Segment erstellen) aus.
8. Wählen Sie in Segment group 1 (Segmentgruppe 1) unter Add filters to refine your segment (Filter zur Optimierung des Segments hinzufügen die Option Filter by user (Nach Benutzer filtern) aus.
9. Wählen Sie unter Benutzerattribut auswählen die Option FirstName. Wählen Sie dann unter Choose values (Werte auswählen) den Vornamen aus, den Sie im Testereignis angegeben haben.

   Im Abschnitt Segment estimate (Segmentschätzung) sollte angezeigt werden, dass es einen qualifizierten Endpunkt und insgesamt einen Endpunkt gibt.

Weiter: Einrichten von Amazon API Gateway

Schritt 4: Einrichten von Amazon API Gateway

In diesem Abschnitt erstellen Sie mithilfe von Amazon API Gateway eine neue API. Das Registrierungsformular, das Sie in dieser Lösung bereitstellen, ruft diese API auf. API Gateway leitet dann die Informationen, die auf dem Registrierungsformular erfasst wurden, an die Lambda-Funktion weiter, die Sie unter Lambda-Funktionen erstellen erstellt haben.

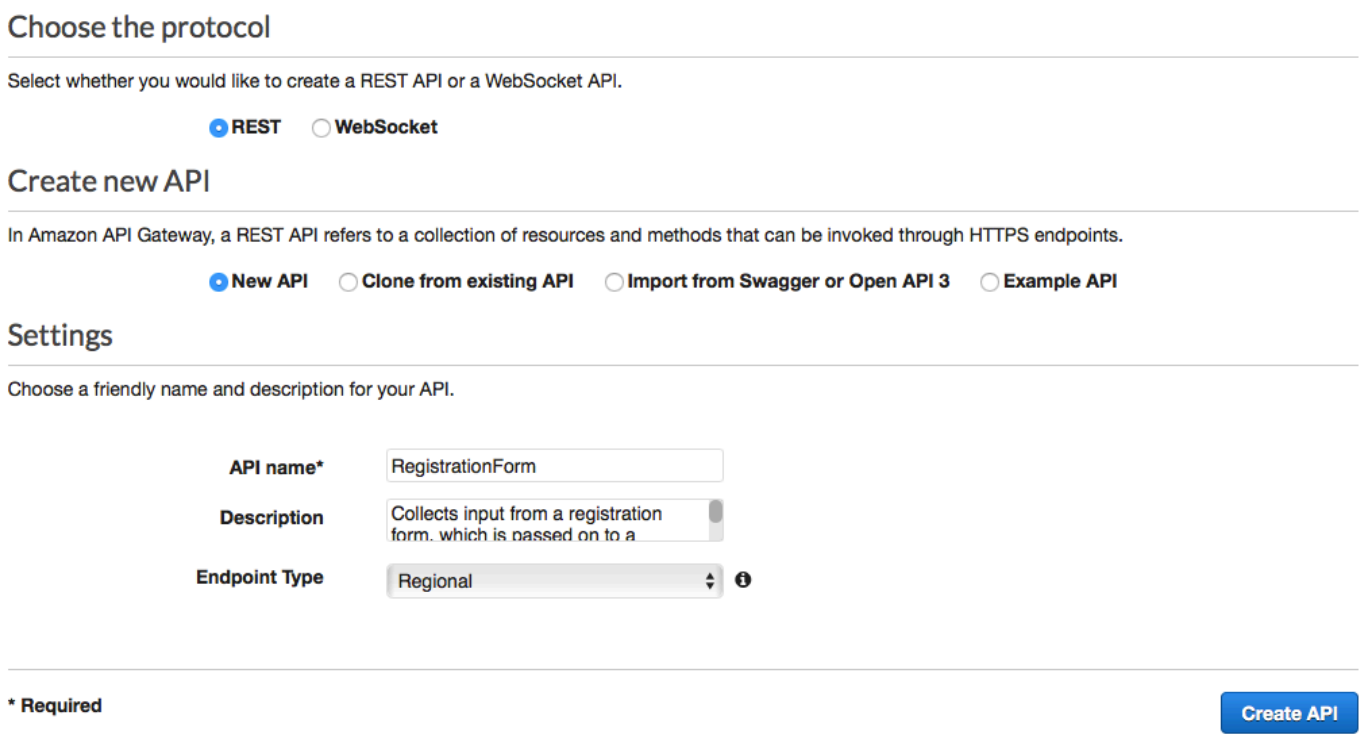
Erstellen der -API

Zunächst müssen Sie eine neue API in API Gateway erstellen. Das folgende Verfahren zeigt, wie Sie eine neue REST-API erstellen.

So erstellen Sie eine neue API

1. Öffnen Sie die API Gateway-Konsole unter https://console.aws.amazon.com/apigateway/.
2. Wählen Sie Create API (API erstellen) aus. Treffen Sie die folgende Auswahl:
   - Wählen Sie unter Choose the protocol (Das Protokoll auswählen) die Option REST aus.
   - Wählen Sie unter Create new API (Neue API erstellen) die Option New API (Neue API) aus.
   - Geben Sie unter Settings (Einstellungen) für Name einen Namen ein, z. B. **`RegistrationForm`**. Geben Sie optional im Feld Description (Beschreibung) einen Text zum Verwendungszweck der API ein. Wählen Sie für Endpoint Type (Endpunkttyp) die Option Regional aus. Wählen Sie dann die Option Create API (API erstellen) aus.

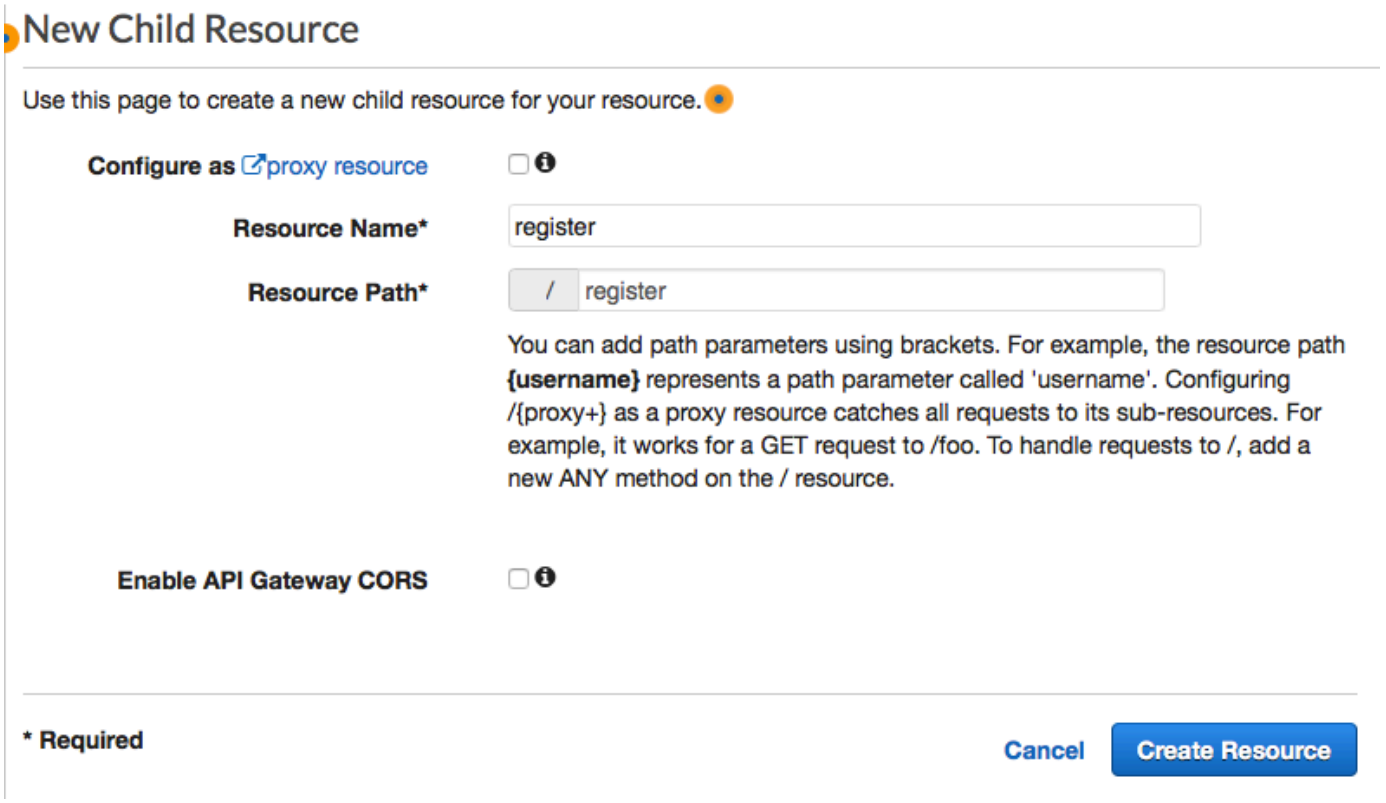
   Ein Beispiel für diese Einstellungen sehen Sie in der folgenden Abbildung.



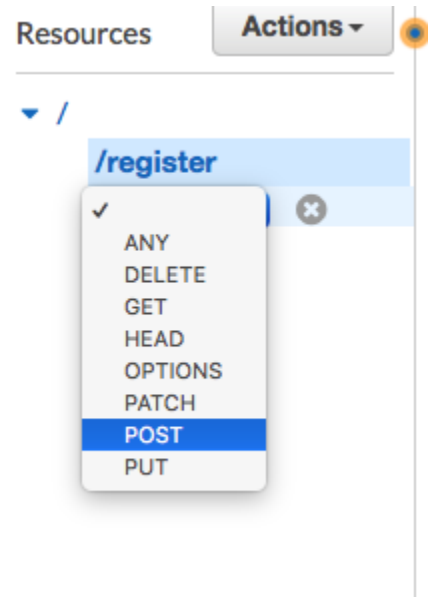
Erstellen einer Ressource

Nachdem Sie eine API erstellt haben, können Sie damit beginnen, Ressourcen hinzuzufügen. Fügen Sie danach eine POST-Methode zur Ressource hinzu und weisen API Gateway an, die Daten, die Sie über diese Methode empfangen, an Ihre Lambda-Funktion zu übergeben.

1. Wählen Sie im Menü Actions (Aktionen) die Option Create Ressource (Ressource erstellen) aus. Geben Sie im Bereich New Child Resource (Neue untergeordnete Ressource) unter Resource Name (Name der Ressource) **`register`** ein, wie in der folgenden Abbildung dargestellt. Wählen Sie Create Resource (Ressource erstellen) aus.



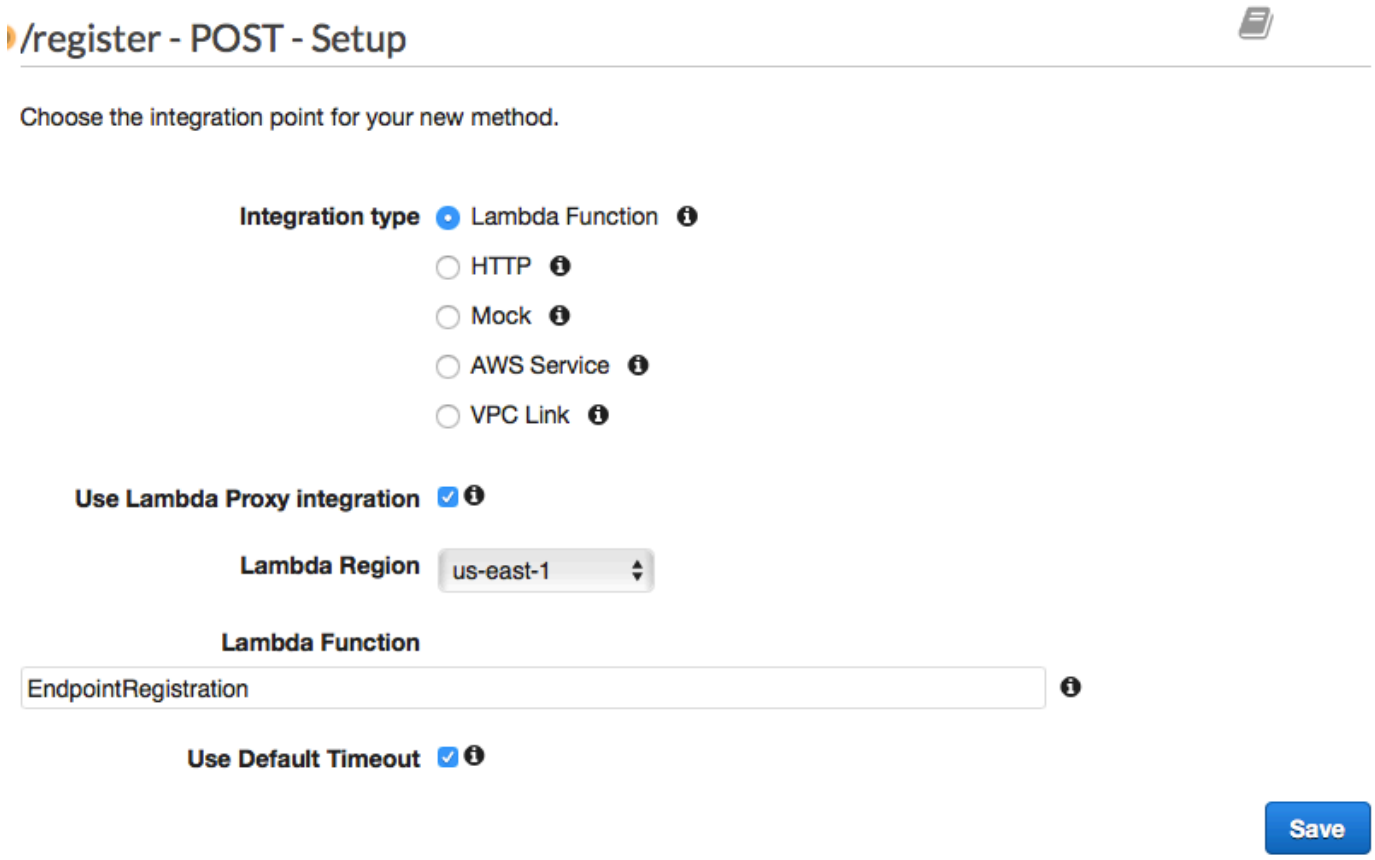
2. Wählen Sie im Menü Actions (Aktionen) die Option Create Method (Methode erstellen) aus. Wählen Sie im angezeigten Menü die Option POST aus, wie in der folgenden Abbildung dargestellt. Klicken Sie dann auf die Häkchen-Schaltfläche ( ).



3. Wählen Sie im Bereich /register – POST – Setup folgende Optionen aus:

- Für Integration type (Integrationstyp) wählen Sie Lambda Function (Lambda-Funktion).
- Wählen Sie die Option Lambda-Proxy-Integration verwenden aus.
- Wählen Sie für die Lambda-Region die Region aus, in der Sie die Lambda-Funktion erstellt haben.
- Wählen Sie für Lambda-Funktion die RegisterEndpoint Funktion aus, die Sie unter Lambda-Funktionen erstellen erstellt haben.

Ein Beispiel für diese Einstellungen sehen Sie in der folgenden Abbildung.



Wählen Sie Speichern. Klicken Sie im Fenster, das angezeigt wird, auf OK, um API Gateway die Berechtigung zu erteilen, Ihre Lambda-Funktion auszuführen.

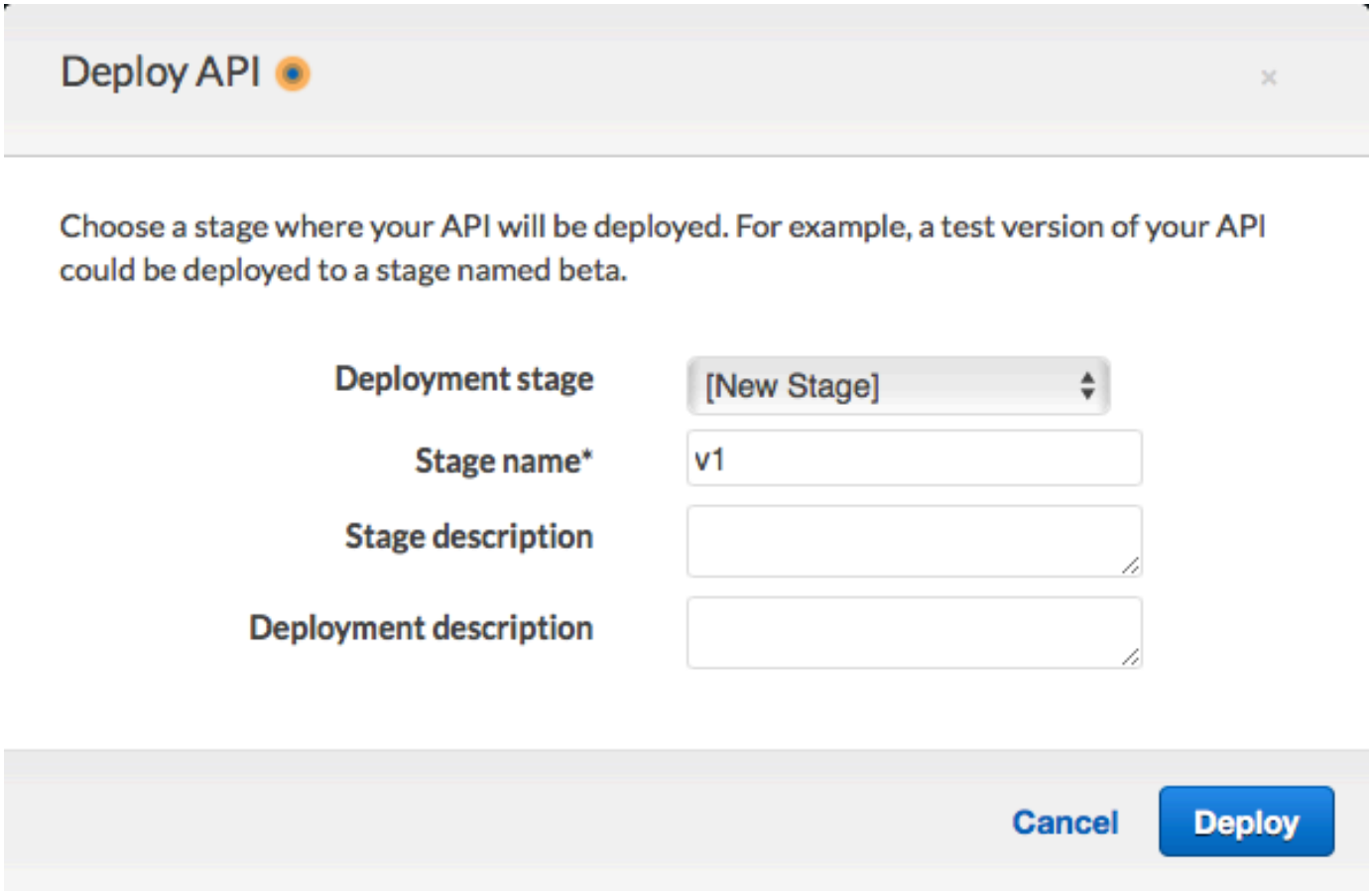
Bereitstellen der API

Die API ist jetzt einsatzbereit. An diesem Punkt müssen Sie sie bereitstellen, um einen öffentlich zugänglichen Endpunkt zu erstellen.

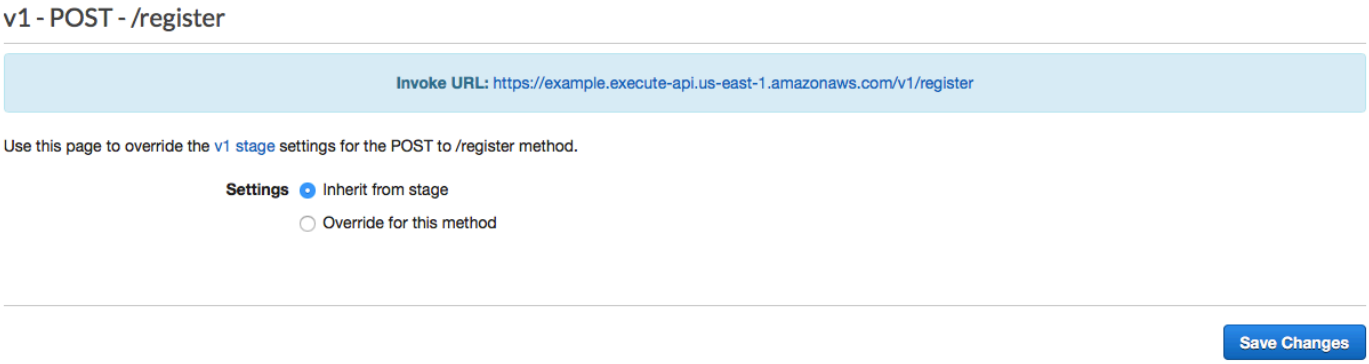
1. Wählen Sie im Menü Actions (Aktionen) die Option Deploy API (API bereitstellen) aus. Wählen Sie im Fenster Deploy API (API bereitstellen) folgende Optionen aus:

   - Wählen Sie für Deployment stage (Bereitstellungsstufe) [New Stage] ([Neue Stufe]) aus.
   - Geben Sie für Stage name (Stufenname) **v1** ein.
   - Wählen Sie Deploy (Bereitstellen) aus.

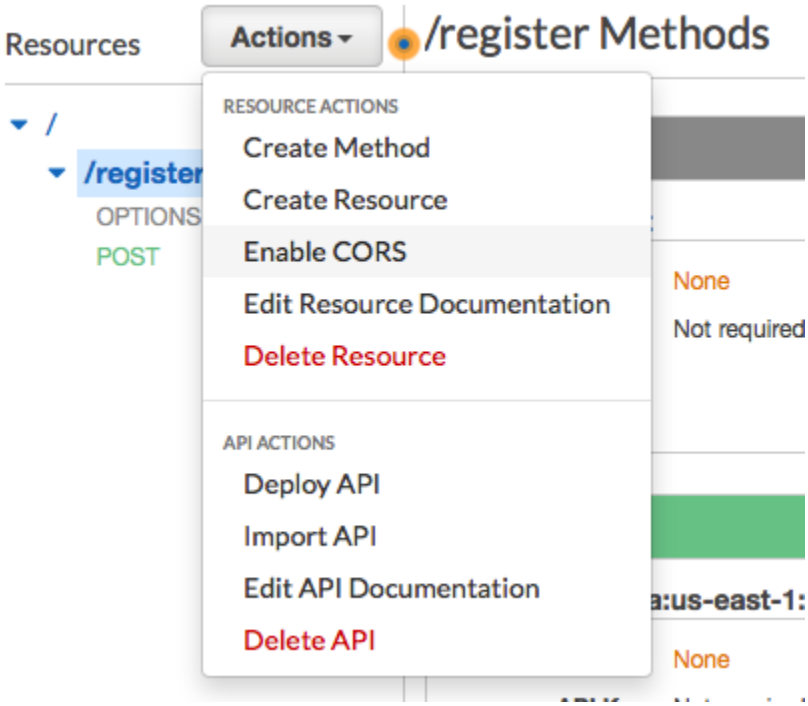
   Ein Beispiel für diese Auswahl ist in der folgenden Abbildung dargestellt.



2. Wählen Sie im Bereich v1 Stage Editor die Ressource /register aus und anschließend die Methode POST. Kopieren Sie die Adresse, die neben Invoke URL (URL aufrufen) angezeigt wird, wie in der folgenden Abbildung dargestellt.



3. Wählen Sie im Navigationsbereich Resources aus. Klicken Sie in der Liste der Ressourcen auf die Ressource /register. Wählen Sie schließlich im Menü Actions (Aktionen) die Option Enable CORS (CORS aktivieren) aus, wie in der folgenden Abbildung dargestellt.



4. Wählen Sie im Bereich Enable CORS (CORS aktivieren) die Option Enable CORS and replace existing CORS headers (CORS aktivieren und bestehende CORS-Header ersetzen) aus.

Nächster Schritt: Erstellen und Bereitstellen des Webformulars

Schritt 5: Erstellen und Bereitstellen des Webformulars

Alle Komponenten dieser Lösung, die AWS Dienste nutzen, sind jetzt vorhanden. Der letzte Schritt besteht in der Erstellung und Bereitstellung des Webformulars, das die Kundendaten erfasst.

Erstellen Sie den JavaScript Formular-Handler

In diesem Abschnitt erstellen Sie eine JavaScript Funktion, die den Inhalt des Webformulars analysiert, das Sie im nächsten Abschnitt erstellen. Nach dem Parsen des Inhalts sendet diese Funktion die Daten an die API, die Sie unter Amazon API Gateway einrichten erstellt haben.

So erstellen Sie den Form Handler

1. Erstellen Sie in einem Texteditor eine neue Datei.
2. Fügen Sie folgenden Code in den Editor ein:

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

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

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

    e.preventDefault();

    if (firstName == "") {
      $('#form-response').html('<div class="mt-3 alert alert-info"
role="alert">Please enter your first name.</div>');
    } else if (lastName == "") {
      $('#form-response').html('<div class="mt-3 alert alert-info"
role="alert">Please enter your last name.</div>');
    } else if (phone.match(/[^0-9]/gi)) {
      $('#form-response').html('<div class="mt-3 alert alert-info"
role="alert">Your phone number contains invalid characters. Please check the phone
number that you supplied.</div>');
    } else if (phone.length < 10) {
      $('#form-response').html('<div class="mt-3 alert alert-info"
role="alert">Please enter your phone number.</div>');
    } else if (phone.length > 10) {
```

```
     $('#form-response').html('<div class="mt-3 alert alert-info"
role="alert">Your phone number contains too many digits. Please check the phone
number that you supplied.</div>');
   } else {
     $('#submit').prop('disabled', true);
     $('#submit').html('<span class="spinner-border spinner-border-sm"
role="status" aria-hidden="true"></span>  Saving your preferences</button>');

     timestamp.setUTCSeconds(utcSeconds);

     var data = JSON.stringify({
       'destinationNumber': phone,
       'firstName': firstName,
       'lastName': lastName,
       'source': source,
       'optTimestamp': timestamp.toString()
     });

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

```
});
```

3. Ersetzen Sie im vorherigen Beispiel *https://example.execute-api.us-east-1.amazonaws.com/v1/register* durch die Aufruf-URL, die Sie in Deploy the API abgerufen haben.
4. Speichern Sie die Datei.

Erstellen Sie die Formulardatei

In diesem Abschnitt erstellen Sie eine HTML-Datei, die das Formular enthält, das Kunden zur Registrierung für Ihr SMS-Programm verwenden. Diese Datei verwendet den JavaScript Formular-Handler, den Sie im vorherigen Abschnitt erstellt haben, um die Formulardaten an Ihre Lambda-Funktion zu übertragen.

> **Important**
>
> Wenn ein Benutzer dieses Formular übermittelt, löst es eine Lambda-Funktion aus, die mehrere Amazon-Pinpoint-API-Operationen aufruft. Böswillige Benutzer könnten einen Angriff auf Ihr Formular starten, der dazu führen kann, dass sehr viele Anfragen gestellt werden. Wenn Sie diese Lösung für einen Anwendungsfall in einer Produktionsumgebung verwenden möchten, sollten Sie sie mithilfe eines Systems wie z. B. Google reCAPTCHA schützen.

So erstellen Sie das Formular

1. Erstellen Sie in einem Texteditor eine neue Datei.
2. Fügen Sie folgenden Code in den Editor ein:

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

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

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/
css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

```
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
 crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/
popper.js/1.14.7/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
 crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/
bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/
nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>

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

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

```
          <div class="input-group">
            <span class="h3">( </span>
            <input type="tel" class="form-control" id="areaCode" placeholder="Area
 code" required>
            <span class="h3"> ) </span>
            <input type="tel" class="form-control" id="phone1" placeholder="555"
 required>
            <span class="h3"> - </span>
            <input type="tel" class="form-control" id="phone2" placeholder="0199"
 required>
          </div>
          <div id="form-response"></div>
          <button id="submit" type="submit" class="btn btn-primary btn-block
 mt-3">Submit</button>
        </form>
      </div>
    </div>
    <div class="row mt-3">
      <div class="col-md-12 text-center">
        <small class="text-muted">Copyright © 2019, ExampleCorp or its
 affiliates.</small>
      </div>
    </div>
  </div>
</body>

</html>
```

3. Ersetzen Sie im vorherigen Beispiel *FormHandlerSMS-.js* durch den vollständigen Pfad zur JavaScript Formular-Handler-Datei, die Sie im vorherigen Abschnitt erstellt haben.
4. Speichern Sie die Datei.

Laden Sie die Formulardateien hoch

Nachdem Sie das HTML-Formular und den JavaScript Formular-Handler erstellt haben, besteht der letzte Schritt darin, diese Dateien im Internet zu veröffentlichen. Dieser Abschnitt geht davon aus, dass Sie einen bestehenden Webhosting-Anbieter haben. Wenn Sie noch keinen Hosting-Anbieter haben, können Sie mithilfe von Amazon Route 53, Amazon Simple Storage Service (Amazon S3) und Amazon eine Website starten CloudFront. Weitere Informationen finden Sie unter Hosten einer statischen Website.

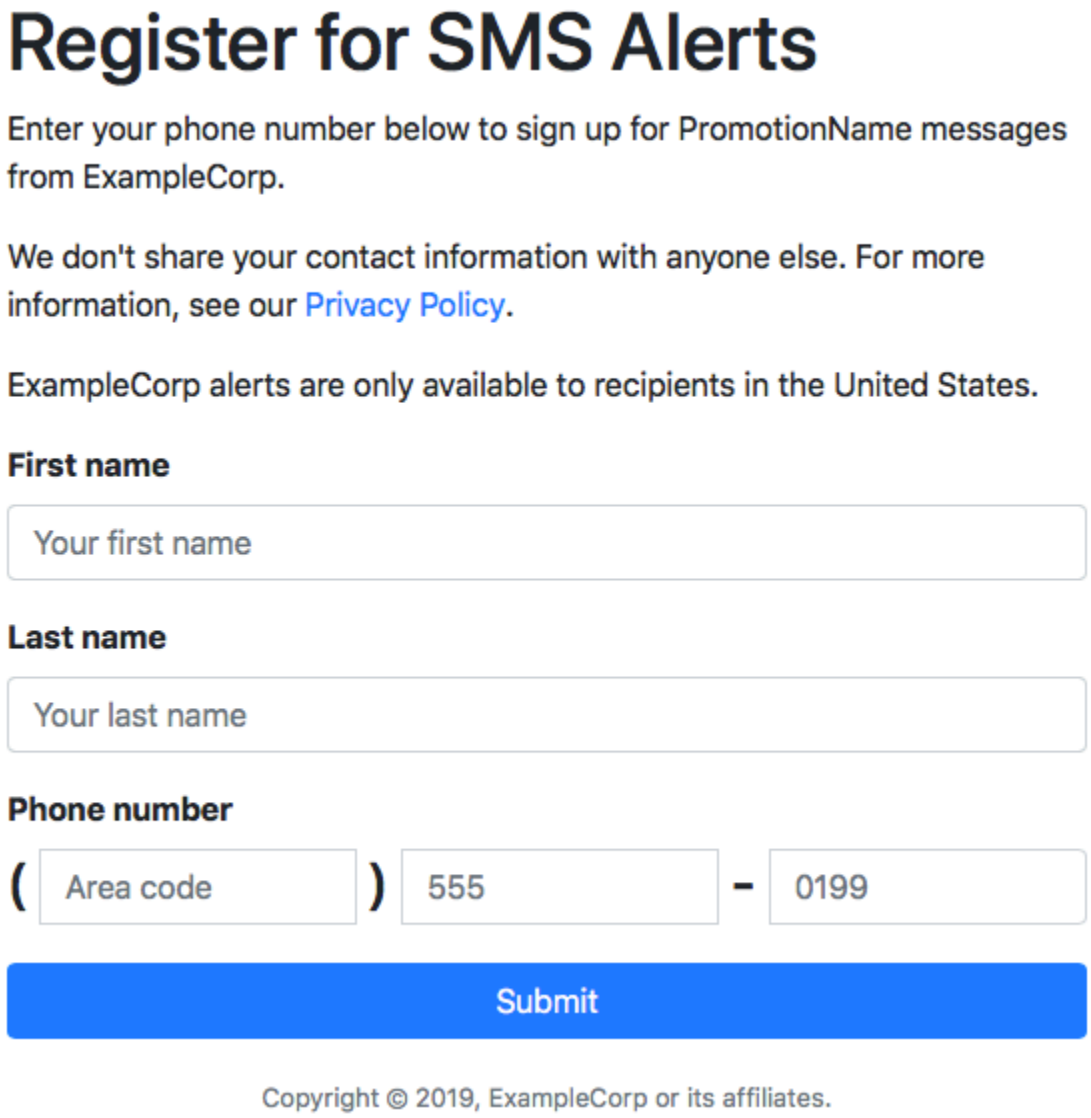
Wenn Sie einen anderen Webhosting-Anbieter verwenden, lesen Sie die Dokumentation des Anbieters, um weitere Informationen zur Veröffentlichung von Webseiten zu erhalten.

Testen des Formulars

Nachdem Sie das Formular veröffentlicht haben, sollten Sie einige Testereignisse übermitteln, um sicherzustellen, dass es wie erwartet funktioniert.

So testen Sie das Registrierungsformular

1. Navigieren Sie in einem Webbrowser zu dem Ort, an dem Sie das Anmeldeformular hochgeladen haben. Wenn Sie das Codebeispiel aus Create the JavaScript form handler verwendet haben, sehen Sie ein Formular, das dem Beispiel in der folgenden Abbildung ähnelt.

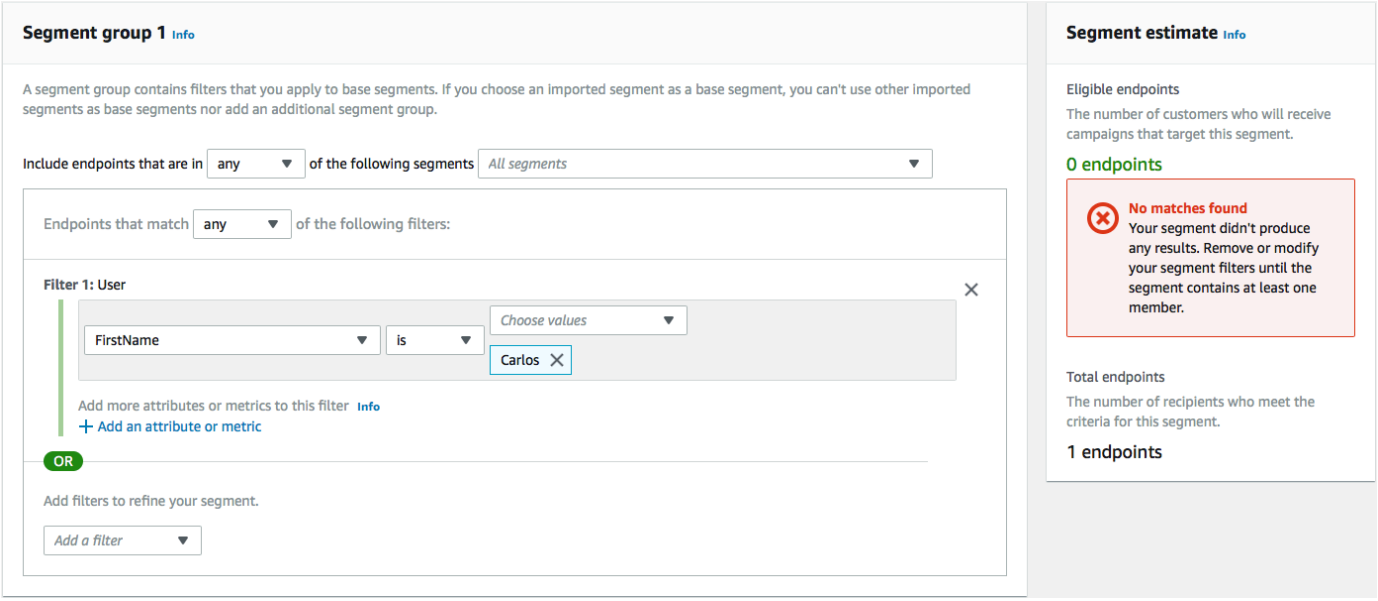


2. Geben Sie Ihre Kontaktinformationen in die Felder First name (Vorname), Last name (Nachname) und Phone number (Telefonnummer) ein.

> **Note**
>
> Wenn Sie das Formular absenden, versucht Amazon Pinpoint, eine Nachricht an die von Ihnen angegebene Telefonnummer zu senden. Aufgrund dieser Funktionalität sollten Sie eine echte Telefonnummer verwenden, um die Lösung von Anfang bis Ende testen zu können.
> Wenn Sie die Lambda-Funktion unter Lambda-Funktionen erstellen getestet haben, enthält Ihr Amazon Pinpoint Pinpoint-Projekt bereits mindestens einen Endpunkt. Wenn Sie dieses Formular testen, sollten Sie entweder eine andere Telefonnummer auf dem Formular angeben oder den vorhandenen Endpunkt mithilfe der DeleteEndpointAPI-Operation löschen.

3. Überprüfen Sie das Gerät, das der von Ihnen angegebenen Telefonnummer zugeordnet ist, um sicherzustellen, dass es die Nachricht empfangen hat.
4. Öffnen Sie die Amazon Pinpoint-Konsole unter https://console.aws.amazon.com/pinpoint/.
5. Wählen Sie auf der Seite Alle Projekte das Projekt aus, das Sie unter Amazon Pinpoint Pinpoint-Projekt erstellen erstellt haben.
6. Wählen Sie im Navigationsbereich die Option Segments (Segmente) aus. Wählen Sie auf der Seite Segments (Segmente) die Option Create a segment (Ein Segment erstellen) aus.
7. Wählen Sie in Segment group 1 (Segmentgruppe 1) unter Add filters to refine your segment (Filter zur Optimierung des Segments hinzufügen die Option Filter by user (Nach Benutzer filtern) aus.
8. Wählen Sie unter Benutzerattribut auswählen die Option FirstName. Wählen Sie dann für Choose values (Werte auswählen) den ersten Namen aus, den Sie angegeben haben, als Sie das Formular übermittelt haben.

   Im Abschnitt Segment estimate (Segmentschätzung) sollte angezeigt werden, dass es keine qualifizierten Endpunkte und insgesamt einen Endpunkt (unter „Total endpoints (Endpunkte gesamt)“) gibt, wie in der folgenden Abbildung gezeigt. Dieses Ergebnis wird erwartet. Wenn die Lambda-Funktion einen neuen Endpunkt erstellt, wird der Endpunkt standardmäßig deaktiviert.



9. Beantworten Sie die Nachricht auf dem Gerät, das die Nachricht erhalten hat, mit dem bidirektionalen SMS-Schlüsselwort, das Sie unter Bidirektional-SMS aktivieren angegeben haben. Amazon Pinpoint sendet sofort eine Antwortnachricht.
10. Wiederholen Sie in der Amazon-Pinpoint-Konsole die Schritte 4 bis 8. Dieses Mal sehen Sie, wenn Sie das Segment erstellen, einen qualifizierten Endpunkt und einen Endpunkt insgesamt. Dieses Ergebnis wird erwartet, da der Endpunkt jetzt qualifiziert ist.

Nächste Schritte

In dem abgeschlossenen Tutorial haben Sie Folgendes getan:

- Ein Amazon-Pinpoint-Projekt erstellt, den SMS-Kanal konfiguriert und eine dedizierte Langwahlnummer abgerufen.
- Eine IAM-Richtlinie erstellt, die das Prinzip der geringsten Berechtigung verwendet, um Zugriffsrechte zu erteilen, und diese Richtlinie mit einer Rolle verknüpft.
- Zwei Lambda-Funktionen erstellt, die die Operationen PhoneNumberValidate, UpdateEndpoint und SendMessages in der Amazon-Pinpoint-API verwenden.
- Hat eine REST-API mit API Gateway erstellt.
- Ein webbasiertes Formular erstellt und bereitgestellt, das Kontaktinformationen von Kunden erfasst.
- Die Lösung getestet, um sicherzustellen, dass sie funktioniert.

In diesem Abschnitt werden einige Möglichkeiten erläutert, wie Sie die Kundendaten nutzen können, die Sie mit dieser Lösung erfassen. Er enthält auch einige Vorschläge, wie Sie diese Lösung für Ihren individuellen Anwendungsfall anpassen können.

Erstellen von Kundensegmenten

Alle Kundendaten, die Sie über dieses Formular erfassen, werden als Endpunkte gespeichert. Diese Lösung erstellt Endpunkte, die mehrere Attribute enthalten, die Sie für die Segmentierung verwenden können.

Diese Lösung erfasst beispielsweise ein Endpunktattribut namens `Source`. Dieses Attribut enthält den vollständigen Pfad zu dem Ort, an dem das Formular gehostet wurde. Wenn Sie ein Segment erstellen, können Sie das Segment nach Endpunkt filtern und dann den Filter weiter verfeinern, indem Sie ein `Source`-Attribut auswählen.

Das Erstellen von Segmenten basierend auf dem Attribut `Source` kann auf mehrere Weisen nützlich sein. Erstens können Sie damit schnell ein Segment von Kunden erstellen, die sich angemeldet haben, um von Ihnen SMS-Nachrichten zu erhalten. Darüber hinaus schließt das Segmentierungstool in Amazon Pinpoint automatisch Endpunkte aus, die nicht akzeptiert haben, Nachrichten zu erhalten.

Das Attribut `Source` ist auch nützlich, wenn Sie sich entscheiden, das Registrierungsformular an mehreren verschiedenen Orten zu hosten. Beispielsweise könnte sich Ihr Marketingmaterial auf ein Formular beziehen, das an einem Ort gehostet ist, während Kunden, die beim Surfen auf Ihrer Website auf das Formular stoßen, eine Version sehen können, die woanders gehostet ist. Wenn Sie dies tun, unterscheidet sich das Quellattribut für Kunden, die das Formular ausfüllen, nachdem sie Ihre Marketingmaterialien gesehen haben, von Kunden, die das Formular ausfüllen, nachdem sie es auf Ihrer Website gefunden haben. Sie können diesen Unterschied nutzen, um verschiedene Segmente zu erstellen und dann maßgeschneiderte Mitteilungen an jede dieser Zielgruppen zu senden.

Senden personalisierter Kampagnennachrichten

Nachdem Sie Segmente angelegt haben, können Sie Kampagnen an diese Segmente senden. Wenn Sie Kampagnennachrichten erstellen, können Sie diese personalisieren, indem Sie angeben, welche Endpunktattribute Sie in die Nachricht aufnehmen möchten. Zum Beispiel verlangt das in dieser Lösung verwendete Webformular, dass der Kunde seinen Vor- und Nachnamen eingibt. Diese Werte werden in dem Benutzerdatensatz gespeichert, der dem Endpunkt zugeordnet ist.

Wenn Sie beispielsweise die API-Operation `GetEndpoint` verwenden, um Informationen über einen Endpunkt abzurufen, der mit dieser Lösung erstellt wurde, sehen Sie einen Abschnitt, der dem folgenden Beispiel ähnelt:

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

Wenn Sie die Werte dieser Attribute in Ihre Kampagnennachricht aufnehmen möchten, können Sie mit der Punktnotation auf das Attribut verweisen. Schließen Sie dann die gesamte Referenz in doppelte geschweifte Klammern ein. Um beispielsweise den Vornamen jedes Empfängers in eine Kampagnennachricht aufzunehmen, fügen Sie die folgende Zeichenfolge in die Nachricht ein: `{{User.UserAttributes.FirstName}}` Wenn Amazon Pinpoint die Nachricht sendet, ersetzt es die Zeichenkette durch den Wert des `FirstName`-Attributs.

Verwenden des Formulars zum Sammeln zusätzlicher Informationen

Sie können diese Lösung ändern, um zusätzliche Informationen auf dem Anmeldeformular zu sammeln. Sie könnten den Kunden beispielsweise bitten, seine Adresse anzugeben, und dann die Felder `Location.City`, `Location.Country`, `Location.Region` und `Location.PostalCode` in der Ressource `Endpoint` mit den Adressdaten ausfüllen. Das Sammeln von Adressinformationen auf dem Registrierungsformular kann dazu führen, dass der Endpunkt genauere Informationen enthält. Um diese Änderung vorzunehmen, müssen Sie dem Webformular die entsprechenden Felder hinzufügen. Sie müssen auch den JavaScript-Code für das Formular ändern, um die neuen Werte zu übergeben. Schließlich müssen Sie die Lambda-Funktion, die den Endpunkt erzeugt, ändern, um die neuen eingehenden Informationen zu verarbeiten.

Sie können das Formular auch so ändern, dass es Kontaktinformationen in anderen Kanälen sammelt. Beispielsweise können Sie mit dem Formular neben der Telefonnummer auch die E-Mail-Adresse des Kunden erfassen. Um diese Änderung vorzunehmen, müssen Sie HTML und JavaScript für das Webformular anpassen. Sie müssen auch die Lambda-Funktion, die den Endpunkt erstellt, so

ändern, dass sie zwei getrennte Endpunkte erzeugt (einen für den E-Mail-Endpunkt und einen für den SMS-Endpunkt). Sie sollten außerdem die Lambda-Funktion so ändern, dass sie einen eindeutigen Wert für das Attribut `User.UserId` erzeugt und diesen Wert dann beiden Endpunkten zuordnet.

Aufzeichnen zusätzlicher Attribute für Prüfungszwecke

Diese Lösung zeichnet zwei wertvolle Attribute auf, wenn sie Endpunkte erstellt und aktualisiert. Erstens, wenn die erste Lambda-Funktion anfänglich den Endpunkt erzeugt, zeichnet sie die URL des Formulars selbst im Attribut `Attributes.Source` auf. Wenn der Kunde auf die Nachricht antwortet, erzeugt die zweite Lambda-Funktion ein Attribut `Attributes.OptInTimestamp`. Dieses Attribut enthält das genaue Datum und die genaue Uhrzeit, wann der Kunde seine Zustimmung gegeben hat, Nachrichten von Ihnen zu empfangen.

Beide Felder können nützlich sein, wenn Sie jemals von einem Mobilfunkbetreiber oder einer Regulierungsbehörde aufgefordert werden, die Zustimmung eines Kunden nachzuweisen. Sie können diese Informationen jederzeit über die API-Operation GetEndpoint abrufen.

Sie können die Lambda-Funktionen auch ändern, um zusätzliche Daten zu erfassen, die für Prüfungszwecke nützlich sein können, wie z. B. die IP-Adresse, von der aus die Registrierungsanfrage gesendet wurde.

Integrieren von Amazon Pinpoint in Ihre Anwendung

Integrieren Sie Amazon Pinpoint in Ihren Client-Code, um Ihre Benutzer zu verstehen und mit ihnen zu interagieren.

Nach der Integration verbindet sich die App beim Start mit dem Amazon-Pinpoint-Service, um Endpunkte hinzuzufügen oder zu aktualisieren. Endpunkte stellen die Ziele dar, an die Sie Nachrichten senden können, z. B. Benutzergeräte, E-Mail-Adressen oder Telefonnummern.

Außerdem stellt Ihre Anwendung auch Nutzungsdaten oder Ereignisse bereit. Zeigen Sie Ereignisdaten in der Amazon-Pinpoint-Konsole an, um zu erfahren, wie viele Benutzer Sie haben, wie oft diese Ihre Anwendung verwenden, wann sie diese verwenden usw.

Sie können die von Ihrer Anwendung bereitgestellten Endpunkte und Ereignisse verwenden, um Messaging-Kampagnen für bestimmte Zielgruppen oder Segmente zu erstellen. (Sie können auch einfachen Empfängerlisten ohne Kampagnen direkt Nachrichten senden.)

Nutzen Sie die Themen in diesem Abschnitt, um Amazon Pinpoint in einen mobilen oder Web-Client zu integrieren. Zu diesen Themen gehören Codebeispiele und Verfahren zur Integration in eine Android- JavaScript, Swift- oder Flutter-Anwendung. Informationen zur Integration Ihrer Apps finden Sie unter the section called “Verbinden Ihrer Frontend-Anwendung mit AWS Amplify”.

Außerhalb Ihres Clients können Sie unterstützte AWS SDKs oder Amazon-Pinpoint-API verwenden, um Endpunkte zu importieren, Ereignisdaten zu exportieren, Kundensegmente zu definieren, Kampagnen zu erstellen und auszuführen usw.

Themen

Amazon Pinpoint mit einem AWS SDK verwenden

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK for C++	AWS SDK for C++ Code-Beispiele
AWS CLI	AWS CLI Code-Beispiele
AWS SDK for Go	AWS SDK for Go Code-Beispiele
AWS SDK for Java	AWS SDK for Java Code-Beispiele
AWS SDK for JavaScript	AWS SDK for JavaScript Code-Beispiele
AWS SDK for Kotlin	AWS SDK for Kotlin Code-Beispiele
AWS SDK for .NET	AWS SDK for .NET Code-Beispiele
AWS SDK for PHP	AWS SDK for PHP Code-Beispiele
AWS Tools for PowerShell	Tools für PowerShell Codebeispiele
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) Code-Beispiele
AWS SDK for Ruby	AWS SDK for Ruby Code-Beispiele
AWS SDK for Rust	AWS SDK for Rust Code-Beispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Code-Beispiele
AWS SDK for Swift	AWS SDK for Swift Code-Beispiele

Für Amazon Pinpoint spezifische Beispiele finden Sie unter Codebeispiele für Amazon Pinpoint unter Verwendung von AWS-SDKs.

> **Beispiel für die Verfügbarkeit**
>
> Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Provide feedback (Feedback geben) auswählen.

Verbinden Ihrer Frontend-Anwendung mit Amazon Pinpoint mit AWS Amplify

Verwenden Sie AWS Amplify, um Ihre App mit AWS zu integrieren. Informationen zu Swift-Apps finden Sie unter Erste Schritte in der Dokumentation zu Amplify für Swift. Informationen zu Android-Apps finden Sie unter Erste Schritte in der Dokumentation zu Amplify für Android SDK. Informationen zur React-Native-App finden Sie unter Erste Schritte in der Dokumentation zu Amplify JavaScript. Informationen zu Flutter-Apps finden Sie unter Erste Schritte in der Dokumentation zum Flutter SDK. Diese Themen sind hilfreich:

- Einrichten Ihrer Backend-Ressourcen.
- Verbinden Sie Ihre App mithilfe der Amplify-Bibliotheken mit den Backend-Ressourcen.

Weitere Informationen zur Verbindung Ihrer Frontend-App mit Amazon Pinpoint für Analytics, In-App-Messaging und Push-Benachrichtigungen finden Sie unter AWS Amplify.

Nächster Schritt

Sie haben AWS Amplify in Ihre Anwendung integriert. Aktualisieren Sie nun Ihren Code, um die Geräte Ihrer Benutzer als Endpunkte registrieren zu können. Siehe Registrieren von Endpunkten in Ihrer Anwendung.

Registrieren von Endpunkten in Ihrer Anwendung

Wenn ein Benutzer eine Sitzung startet (beispielsweise durch Starten Ihrer mobilen App), kann Ihre mobile oder Webanwendung automatisch einen Endpunkt bei Amazon Pinpoint registrieren (oder aktualisieren). Der Endpunkt repräsentiert das Gerät, mit dem der Benutzer die Sitzung startet. Er beinhaltet Attribute, die das Gerät beschreiben und kann auch benutzerdefinierte Attribute enthalten, die Sie definieren. Endpunkte können auch andere Methoden zur Kommunikation mit Kunden darstellen, wie E-Mail-Adressen oder Mobiltelefonnummern.

Nachdem die Anwendung Endpunkte registriert, können Sie Ihre Zielgruppe nach Endpunktattributen unterteilen. Sie können diese Segmente mit angepassten Messaging-Kampagnen ansprechen. Auf der Seite Analytics in der Amazon-Pinpoint-Konsole sehen Sie Diagramme über die Registrierung und die Aktivitäten von Endpunkten, wie beispielsweise Neue Endpunkte und Aktive Endpunkte pro Tag.

Sie können mehreren Endpunkten eine einzelne Benutzer-ID zuweisen. Eine Benutzer-ID repräsentiert einen einzelnen Benutzer. Ein der Benutzer-ID zugewiesener Endpunkt repräsentiert dagegen eines der Geräte des Benutzers. Nachdem Sie Ihren Endpunkten Benutzer-IDs zugewiesen haben, können Sie Diagramme zur Benutzeraktivität in der Konsole anzeigen, z. B. Daily active users und Monthly active users.

Bevor Sie beginnen

Wenn Sie dies noch nicht getan haben, integrieren Sie das AWS SDK für Mobilgeräte für Android oder iOS oder die AWS Amplify JavaScript-Bibliothek in Ihre Anwendung. Siehe Verbinden Ihrer Frontend-Anwendung mit Amazon Pinpoint mit AWS Amplify.

Registrieren von Endpunkten bei den AWS Mobile SDKs für Android oder iOS

Mit den AWS SDKs für Mobilgeräte für Android oder iOS können Sie Endpunkte registrieren und anpassen. Weitere Informationen sowie Codebeispiele finden Sie in den folgenden Dokumenten:

- Registrieren von Endpunkten in Ihrer Anwendung in der Dokumentation zum Android SDK.
- Registrieren von Endpunkten in Ihrer Anwendung in der Dokumentation zum iOS SDK.

Registrieren von Endpunkten mit der AWS Amplify JavaScript-Bibliothek

Sie können die AWS Amplify JavaScript-Bibliothek verwenden, um Endpunkte in Ihren Apps zu registrieren und zu aktualisieren. Weitere Informationen und Codebeispiele finden Sie unter Update endpoint in der AWS Amplify JavaScript-Dokumentation.

Nächste Schritte

Sie haben Ihre App aktualisiert, um Endpunkte registrieren zu können. Ab sofort werden von Amazon Pinpoint Geräteinformationen und benutzerdefinierte Attribute bereitgestellt, wenn ein Benutzer

Ihre App startet. Mithilfe dieser Informationen können Sie Zielgruppensegmente erstellen. In der Konsole finden Sie Metriken zu Endpunkten sowie gegebenenfalls Benutzer, denen eine Benutzer-ID zugewiesen ist.

Als Nächstes führen Sie die Schritte unter Melden von Ereignissen in Ihrer Anwendung , um Ihre App zu aktualisieren, um Nutzungsdaten zu melden.

Melden von Ereignissen in Ihrer Anwendung

In Ihrer mobilen oder Webanwendung können Sie mithilfe der AWS SDKs für Mobilgeräte oder der Amazon-Pinpoint-Ereignis-API Nutzungsdaten oder Ereignisse an Amazon Pinpoint melden. Sie können Ereignisse melden, um Informationen wie etwa Sitzungsdauern, das Kaufverhalten der Benutzer, Anmeldeversuche oder benutzerdefinierte Ereignistypen für Ihre Anforderungen zu erfassen.

Nachdem Ihre Anwendung Ereignisse gemeldet hat, können Sie die Analyse in der Amazon-Pinpoint-Konsole anzeigen. Die Diagramme auf der Seite Analytics bieten Metriken für viele Aspekte des Benutzerverhaltens. Weitere Informationen finden Sie unter Chart-Referenz für Amazon Pinpoint Analytics im Amazon-Pinpoint-Benutzerhandbuch.

Um Ihre Ereignisdaten außerhalb von Amazon Pinpoint zu analysieren oder zu speichern, können Sie Amazon Pinpoint so konfigurieren, dass die Daten zu Amazon Kinesis gestreamt werden. Weitere Informationen finden Sie unter Streamen von Amazon-Pinpoint-Ereignissen zu Kinesis.

Mithilfe der AWS SDKs für Mobilgeräte und der AWS-Amplify-JavaScript-Bibliotheken können Sie die Amazon-Pinpoint-API aufrufen, um die folgenden Ereignistypen zu melden:

Sitzungsereignisse

Geben Sie an, wann und wie oft Benutzer Ihre App öffnen und schließen.

Nachdem Ihre Anwendung Sitzungsereignisse gemeldet hat, zeigen Sie auf der Seite Analysen in der Amazon-Pinpoint-Konsole Diagramme für Sitzungen, Täglich aktive Endpunkte, 7-tägige Aufbewahrungsfrist und anderes an.

Benutzerdefinierte Ereignisse

Sind nicht standardmäßige Ereignisse, die Sie durch Zuweisen eines benutzerdefinierten Ereignistyps definieren. Sie können benutzerdefinierte Attribute und Metriken zu benutzerdefinierten Ereignissen hinzufügen.

Auf der Seite Analytics (Analysen) in der Konsole zeigt die Registerkarte Events (Ereignisse) Metriken für alle benutzerdefinierten Ereignisse an, die Ihre App meldet.

Monetarisierungsereignisse

Melden den Umsatz, der von Ihrer Anwendung erzeugt wird, sowie die Anzahl an Artikeln, die von Benutzern gekauft werden.

Auf der Seite Analytics (Analyse) zeigt die Registerkarte Revenue (Umsatz) Diagramme für Revenue (Umsatz), Paying users (Zahlende Benutzer), Units sold (Verkaufte Einheiten) und anderes an.

Authentifizierungsereignisse

Gibt an, wie oft Benutzer sich in Ihrer Anwendung authentifizieren.

Auf der Seite Analytics (Analysen) zeigt die Registerkarte Users (Benutzer) Diagramme für Sign-ins (Anmeldungen), Sign-ups (Registrierungen) und Authentication failures (Authentifizierungsfehler) an.

Bevor Sie beginnen

Sofern Sie das noch nicht getan haben, führen Sie die folgenden Schritte aus:

- Integrieren Sie Ihre App mit AWS Amplify. Siehe Verbinden Ihrer Frontend-Anwendung mit Amazon Pinpoint mit AWS Amplify.
- Aktualisieren Sie Ihre Anwendung, um Endpunkte registrieren zu können. Siehe Registrieren von Endpunkten in Ihrer Anwendung.

Melden von Ereignissen mit den AWS Mobile SDKs für Android oder iOS

Sie können eine mobile App so konfigurieren, dass sie mithilfe der AWS SDKs für Mobilgeräte für iOS und Android Ereignisse an Amazon Pinpoint meldet.

Weitere Informationen zum Aktualisieren Ihrer App zum Aufzeichnen und Übermitteln von Ereignissen an Amazon Pinpoint finden Sie auf den folgenden Seiten in der Dokumentation zu AWS Amplify:

- Analytics in der Dokumentation zum iOS SDK
- Analytics in der Dokumentation zum Android SDK

Melden von Ereignissen mit der AWS Amplify JavaScript-Bibliothek

Sie können JavaScript und React Native-Apps aktivieren, um Anwendungsnutzungsereignisse an Amazon Pinpoint zu melden, indem Sie die AWS-Amplify-JavaScript-Bibliothek verwenden. Weitere Informationen zum Aktualisieren Ihrer App zur Übermittlung von Ereignissen an Amazon Pinpoint finden Sie unter Analytics in der Dokumentation zu AWS Amplify JavaScript.

Melden von Ereignissen unter Verwendung der Amazon-Pinpoint-API

Sie können mit der Amazon-Pinpoint-API oder einem AWS-SDK Ereignisse gesammelt an Amazon Pinpoint übertragen. Weitere Informationen finden Sie unter Ereignisse in der Amazon-Pinpoint-API-Referenz.

Nächster Schritt

Sie haben Ihre App aktualisiert, um Ereignisse melden zu können. Wenn Benutzer nun mit Ihrer App interagieren, werden Nutzungsdaten an Amazon Pinpoint gesendet. Sie können diese Daten in der Konsole anzeigen oder sie nach Amazon Kinesis streamen.

Aktualisieren Sie nun Ihre App, um Push-Benachrichtigungen, die Sie mit Amazon Pinpoint senden, verarbeiten zu können. Siehe Umgang mit Push-Benachrichtigungen.

Umgang mit Push-Benachrichtigungen

In den folgenden Themen wird beschrieben, wie Sie Ihre Swift-, Android-, React Native- oder Flutter-App ändern, sodass sie von Amazon Pinpoint gesendete Push-Benachrichtigungen erhält.

Themen

Einrichten von Push-Benachrichtigungen für Amazon Pinpoint

Um Amazon Pinpoint so einzurichten, dass Push-Benachrichtigungen an Ihre Apps gesendet werden können, müssen Sie zunächst die Anmeldeinformationen angeben, die Amazon Pinpoint zum Senden von Nachrichten an Ihre App berechtigen. Welche Anmeldeinformationen Sie angeben, hängt davon ab, welches Push-Benachrichtigungssystem Sie verwenden:

- Stellen Sie für iOS-Apps ein SSL-Zertifikat bereit, das Sie vom Apple Developer-Portal beziehen. Das Zertifikat autorisiert Amazon Pinpoint zum Versenden von Nachrichten an Ihre App über den Apple Push Notification Service (APNs).
- Für Android-Apps stellen Sie einen Web-API-Schlüssel zur Verfügung, den Sie über die Firebase-Konsole erhalten. Diese Anmeldeinformationen autorisieren Amazon Pinpoint zum Versenden von Nachrichten an Ihre App über Firebase Cloud Messaging.

Nachdem Sie die Anmeldeinformationen für einen Push-Benachrichtigungskanal abgerufen haben, müssen Sie ein Projekt in Amazon Pinpoint erstellen und ihm die Anmeldeinformationen für den Push-Benachrichtigungsdienst bereitstellen.

Themen

Einrichten von Swift-Push-Benachrichtigungen

Push-Benachrichtigungen für iOS-Apps werden mit dem Apple-Push-Notification-Service (APNs) gesendet. Bevor Sie Push-Benachrichtigungen an iOS-Geräte senden können, müssen Sie eine App-ID im Apple Developer-Portal anlegen und die erforderlichen Zertifikate erstellen. Weitere Informationen zum Ausführen dieser Schritte finden Sie unter Einrichten von Push-Benachrichtigungs-Services in der Dokumentation zu AWS Amplify.

Arbeiten mit APNs-Token

Eine bewährte Methode ist das Entwickeln der App in der Art und Weise, dass die Geräte-Token Ihrer Kunden bei der Neuinstallation der App neu generiert werden.

Wenn ein Empfänger sein Gerät auf eine neue Hauptversion von iOS aktualisiert (z. B. von iOS 12 auf iOS 13) und später Ihre App neu installiert, generiert die App ein neues Token. Wenn Ihre App das Token nicht aktualisiert, wird zum Senden der Benachrichtigung das ältere Token verwendet. Infolgedessen lehnt der Apple Push Notification-Service (APNs) die Benachrichtigung ab, da das Token nun ungültig ist. Wenn Sie versuchen, die Benachrichtigung zu senden, erhalten Sie eine Fehlerbenachrichtigung in Form einer Nachricht von APNs.

Einrichten von Android-Push-Benachrichtigungen

Push-Benachrichtigungen für Android-Apps werden über Firebase Cloud Messaging (FCM) gesendet, das Google Cloud Messaging (GCM) ersetzt Bevor Sie Push-Benachrichtigungen an Android-Geräte senden können, müssen Sie FCM-Anmeldeinformationen erhalten. Mit diesen Anmeldeinformationen können Sie dann ein Android-Projekt erstellen und eine Beispielanwendung starten, die Push-Benachrichtigungen empfangen kann. Weitere Informationen zum Ausführen dieser Schritte finden Sie im Abschnitt Push-Benachrichtigungen in der Dokumentation zu AWS Amplify.

Einrichten von Flutter-Push-Benachrichtigungen

Push-Benachrichtigungen für Flutter-Apps werden mit Firebase Cloud Messaging (FCM) für Android und APNs für iOS gesendet. Weitere Informationen zum Ausführen dieser Schritte finden Sie im Abschnitt Push-Benachrichtigungen in der Dokumentation zu AWS Amplify Flutter.

Einrichten von React-Native-Push-Benachrichtigungen

Push-Benachrichtigungen für React-Native-Apps werden mit Firebase Cloud Messaging (FCM) für Android und APNs für iOS gesendet. Weitere Informationen zum Ausführen dieser Schritte finden Sie im Abschnitt Push-Benachrichtigungen in der Dokumentation zu AWS Amplify JavaScript.

Ein Projekt in Amazon Pinpoint erstellen

In Amazon Pinpoint ist ein Projekt eine Sammlung von Einstellungen, Daten, Kampagnen und Segmenten, die alle einen gemeinsamen Zweck haben. In der Amazon-Pinpoint-API werden Projekte auch als Anwendungen bezeichnet. Dieser Abschnitt verwendet das Wort "Projekt" ausschließlich in Bezug auf dieses Konzept.

Um mit dem Senden von Push-Benachrichtigungen in Amazon Pinpoint zu beginnen, müssen Sie ein Projekt erstellen. Als Nächstes müssen Sie die Push-Benachrichtigungskanäle aktivieren, die Sie verwenden möchten, indem Sie die entsprechenden Anmeldeinformationen angeben.

Mit der Amazon-Pinpoint-Konsole können Sie neue Projekte erstellen und Push-Benachrichtigungskanäle einrichten. Weitere Informationen finden Sie unter Einrichten von Push-Benachrichtigungskanälen in Amazon Pinpoint im Amazon-Pinpoint-Benutzerhandbuch.

Sie können Projekte auch mithilfe der Amazon-Pinpoint-API, eines AWS SDKs oder der AWS Command Line Interface (AWS CLI) erstellen und einrichten. Verwenden Sie die `Apps`-Ressource zum Erstellen eines Projekts. Verwenden Sie zum Konfigurieren von Push-Benachrichtigungskanälen die folgenden Ressourcen:

- APNs-Kanal zum Senden von Nachrichten an Benutzer von iOS-Geräten mithilfe des Apple Push Notification Service.
- ADM-Kanal zum Senden von Nachrichten an Benutzer von Amazon Kindle Fire-Geräten.
- Baidu-Kanal zum Senden von Nachrichten an Baidu-Benutzer.
- GCM-Channel zum Senden von Nachrichten an Android-Geräte mithilfe von Firebase Cloud Messaging (FCM), das Google Cloud Messaging (GCM) ersetzt.

Umgang mit Push-Benachrichtigungen

Nachdem Sie die zum Senden von Push-Benachrichtigungen erforderlichen Anmeldeinformationen abgerufen haben, können Sie Ihre Apps aktualisieren, damit sie Push-Benachrichtigungen erhalten können. Weitere Informationen finden Sie unter Push-Benachrichtigungen – Erste Schritte in der AWS Amplify-Dokumentation.

Definieren Ihrer Zielgruppe für Amazon Pinpoint

In Amazon Pinpoint wird jedes Element Ihrer Zielgruppe durch einen oder mehrere Endpunkte dargestellt. Wenn Sie mit Amazon Pinpoint eine Nachricht senden, können Sie die Nachricht an die Endpunkte richten, die die Mitglieder Ihrer Zielgruppe darstellen. Jede Endpunktdefinition enthält ein Nachrichtenziel, z. B. ein Geräte-Token, eine E-Mail-Adresse oder eine Telefonnummer. Außerdem enthält sie Daten über Ihre Benutzer und deren Geräte. Bevor Sie Ihre Zielgruppe analysieren, segmentieren oder ansprechen, ist der erste Schritt, Ihrem Amazon-Pinpoint-Projekt Endpunkte hinzuzufügen.

Um Endpunkte hinzuzufügen, können Sie:

- Amazon Pinpoint in Ihren Android-, iOS- oder JavaScript-Client integrieren, sodass Endpunkte automatisch hinzugefügt werden, wenn Benutzer Ihre Anwendung besuchen.
- Die Amazon-Pinpoint-API verwenden, um Endpunkte einzeln oder in Batches hinzuzufügen.
- Endpunktdefinitionen importieren, die außerhalb von Amazon Pinpoint gespeichert sind.

Nachdem Sie Endpunkte hinzugefügt haben, können Sie:

- Analysen über Ihre Zielgruppe in der Amazon-Pinpoint-Konsole anzeigen.
- Mehr über Ihre Zielgruppe erfahren, indem Sie Endpunktdaten abfragen oder exportieren.
- Zielgruppensegmente basierend auf Endpunktattributen definieren, wie z. B. demografische Daten oder Benutzerinteressen.
- Ihre Zielgruppen mit maßgeschneiderten Messaging-Kampagnen ansprechen.
- Nachrichten direkt an Listen von Endpunkten senden.

Verwenden Sie die Themen in diesem Abschnitt, um Endpunkte unter Verwendung der Amazon-Pinpoint-API hinzuzufügen, zu aktualisieren und zu löschen. Wenn Sie Endpunkte automatisch von Ihrem Android-, iOS- oder JavaScript-Client hinzufügen möchten, lesen Sie stattdessen unter Registrieren von Endpunkten in Ihrer Anwendung nach.

Themen

Hinzufügen von Endpunkten zu Amazon Pinpoint

Ein Endpunkt stellt ein Ziel dar, an das Sie Nachrichten senden können, wie zum Beispiel ein mobiles Gerät, eine Telefonnummer oder eine E-Mail-Adresse. Bevor Sie Nachrichten an ein Mitglied Ihrer Zielgruppe senden können, müssen Sie einen oder mehrere Endpunkte definieren.

Wenn Sie einen Endpunkt definieren, geben Sie den Kanal und die Adresse an. Der Kanal ist der Typ der Plattform, die Sie verwenden, um Nachrichten an den Endpunkt zu senden. Beispiele für Kanäle sind ein Push-Benachrichtigungsservice, SMS oder E-Mail. Die Adresse gibt an, wohin die Nachricht an den Endpunkt gesendet werden soll, wie z. B. ein Geräte-Token, eine Telefonnummer oder E-Mail-Adresse.

Um weitere Details über Ihre Zielgruppe hinzuzufügen, können Sie Ihre Endpunkte um benutzerdefinierte und Standard-Attribute erweitern. Diese Attribute können beispielsweise Daten über Ihre Benutzer enthalten, ihre Präferenzen, ihre Geräte, die Versionen des Clients, die sie verwenden, oder ihre Standorte. Wenn Sie Ihren Endpunkten diese Art Daten hinzufügen, können Sie:

- Zeigen Sie Diagramme über Ihre Zielgruppe in der Amazon-Pinpoint-Konsole an.
- Ihre Zielgruppe basierend auf Endpunktattributen unterteilen, damit Sie Ihre Nachrichten genau an den richtigen Teil Ihrer Zielgruppe senden können.
- Ihre Nachrichten personalisieren, indem Sie Nachrichtenvariablen verwenden, die durch Endpunktattributwerte ersetzt werden.

Eine mobile oder JavaScript-Client-Anwendung registriert Endpunkte automatisch, wenn Sie Amazon Pinpoint unter Verwendung der mobilen AWS-SDKs oder der AWS-Amplify-JavaScript-Bibliothek integrieren. Der Client registriert einen Endpunkt für jeden neuen Benutzer und aktualisiert Endpunkte für zurückkehrende Benutzer. Weitere Informationen, wie Sie Endpunkte von einem mobilen oder JavaScript-Client registrieren, finden Sie unter Registrieren von Endpunkten in Ihrer Anwendung.

Beispiele

Die folgenden Beispiele zeigen Ihnen, wie Sie einem Amazon-Pinpoint-Projekt einen Endpunkt hinzufügen. Der Endpunkt stellt ein Zielgruppenmitglied dar, das in Seattle lebt und ein iPhone verwendet. Dieser Person können Nachrichten über den Apple Push Notification-Service (APNs) gesendet werden. Die Adresse des Endpunkts ist das Geräte-Token, das von APNs bereitgestellt wird.

AWS CLI

Sie können Amazon Pinpoint verwenden, indem Sie Befehle mit der AWS CLI ausführen.

Example Befehl zum Aktualisieren eines Endpunkts

Um einen Endpunkt hinzuzufügen oder zu aktualisieren, verwenden Sie den Befehl update-endpoint:

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

Wobei gilt:

- application-id ist die ID des Amazon-Pinpoint-Projekts, in dem Sie einen Endpunkt hinzufügen oder aktualisieren.
- example-endpoint ist die ID, die Sie einem neuen Endpunkt zuweisen, oder die ID eines bestehenden Endpunkts, den Sie aktualisieren.
- endpoint-request-file.json ist der Dateipfad zu einer lokalen JSON-Datei, die die Eingabe für den Parameter `--endpoint-request` enthält.

Example Endpunkt-Anforderungsdatei

Der Beispielbefehl `update-endpoint` verwendet eine JSON-Datei als Argument für den Parameter `--endpoint-request`. Diese Datei enthält eine Endpunktdefinition wie die folgende:

```
{
  "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
  }
}
```

Weitere Informationen zu den Attributen, die Sie verwenden können, um einen Endpunkt zu definieren, finden Sie im EndpointRequest-Schema in der Amazon-Pinpoint-API-Referenz.

AWS SDK for Java

Sie können die Amazon-Pinpoint-API in Ihren Java-Anwendungen verwenden, indem Sie den vom AWS SDK for Java bereitgestellten Client verwenden.

Example Code

Um einen Endpunkt hinzuzufügen, initialisieren Sie ein `EndpointRequest`-Objekt und übergeben es der Methode `updateEndpoint` des `AmazonPinpoint`-Clients:

```
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

Sie können Amazon Pinpoint verwenden, indem Sie HTTP-Anforderungen direkt an die REST-API stellen.

Example PUT-Anforderung zum Hinzufügen eines Endpunkts

Um einen Endpunkt hinzuzufügen, stellen Sie eine PUT-Anforderung an die Endpunkt-Ressource unter der folgenden URI:

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

Wobei gilt:

- application-id ist die ID des Amazon-Pinpoint-Projekts, in dem Sie einen Endpunkt hinzufügen oder aktualisieren.
- endpoint-id ist die ID, die Sie einem neuen Endpunkt zuweisen, oder die ID eines bestehenden Endpunkts, den Sie aktualisieren.

Fügen Sie Ihrer Anforderung die erforderlichen Header hinzu und geben Sie die EndpointRequest-JSON als Text an.

```
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
  }
}
```

Wenn Ihre Anforderung erfolgreich ist, erhalten Sie eine Antwort wie die folgende:

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

Ähnliche Informationen

Weitere Informationen über die Endpunkt-Ressource in der Amazon-Pinpoint-API, einschließlich der unterstützten HTTP-Methoden und Anforderungsparameter, finden Sie unter Endpunkt in der Amazon-Pinpoint-API-Referenz.

Weitere Informationen zum Personalisieren von Nachrichten mit Variablen finden Sie unter Nachrichtenvariablen im Amazon-Pinpoint-Benutzerhandbuch.

Informationen zu den Kontingenten für Endpunkte, z. B. die Anzahl der Attribute, die Sie zuweisen können, finden Sie unter the section called “Endpunktkontingente”.

Verknüpfen von Benutzern mit Endpunkten von Amazon Pinpoint

Ein Endpunkt kann Attribute enthalten, die einen Benutzer definieren, der eine Person in Ihrer Zielgruppe darstellt. Ein Benutzer könnte beispielsweise eine Person darstellen, die Ihre mobile App installiert hat, oder eine Person, die ein Konto auf Ihrer Website hat.

Sie definieren einen Benutzer, indem Sie eine eindeutige Benutzer-ID und optional benutzerdefinierte Benutzerattribute angeben. Wenn jemand Ihre App auf mehreren Geräten verwendet, oder wenn diese Person über mehrere Adressen Nachrichten erhalten kann, können Sie mehreren Endpunkten die gleiche Benutzer-ID zuordnen. In diesem Fall synchronisiert Amazon Pinpoint Benutzerattribute über die Endpunkte. Wenn Sie also einem Endpunkt ein Benutzerattribut hinzufügen, fügt Amazon Pinpoint dieses Attribut jedem Endpunkt hinzu, der dieselbe Benutzer-ID enthält.

Sie können Benutzerattribute hinzufügen, um Daten nachzuverfolgen, die sich auf eine einzelne Person beziehen, und die nicht abhängig von dem Gerät, das die Person verwendet, variieren. Sie können beispielsweise Attribute für den Namen einer Person ihr Alter oder ihren Kontostatus hinzufügen.

> **Tip**
>
> Wenn Ihre Anwendung Amazon-Cognito-Benutzerpools für die Benutzerauthentifizierung verwendet, kann Amazon Cognito Ihren Endpunkten automatisch Benutzer-IDs und Attribute hinzufügen. Für den Benutzer-ID-Wert des Endpunkts weist Amazon Cognito den `sub`-Wert zu, der dem Benutzer im Benutzerpool zugeordnet ist. Weitere Informationen zum Hinzufügen von Benutzern mit Amazon Cognito finden Sie unter Verwenden von Amazon Pinpoint Analytics mit Amazon-Cognito-Benutzerpools im Amazon Cognito-Entwicklerhandbuch.

Nachdem Sie Ihren Endpunkten Benutzerdefinitionen hinzugefügt haben, haben Sie mehrere Möglichkeiten, wie Sie Ihre Zielgruppe segmentieren. Sie können ein Segment basierend auf Benutzerattributen definieren, oder Sie können ein Segment durch Importieren einer Liste mit Benutzer-IDs definieren. Wenn Sie eine Nachricht an ein Segment senden, das auf Benutzern basiert, umfassen die potenziellen Ziele alle Endpunkte, die den einzelnen Benutzer-IDs in dem Segment zugeordnet sind.

Außerdem haben Sie mehrere Möglichkeiten, wie Sie die Nachrichten an Ihre Zielgruppe senden. Sie können eine Kampagne verwenden, um Nachrichten an ein Segment von Benutzern zu senden, oder Sie können eine Nachricht direkt an eine Liste mit Benutzer-IDs senden. Um Ihre Nachricht zu

personalisieren, können Sie Nachrichtenvariablen aufnehmen, die durch Benutzerattributwerte ersetzt werden.

Beispiele

Die folgenden Beispiele zeigen Ihnen, wie Sie einem Endpunkt eine Benutzerdefinition hinzufügen.

AWS CLI

Sie können Amazon Pinpoint verwenden, indem Sie Befehle mit der AWS CLI ausführen.

Example Befehl zum Aktualisieren eines Endpunkts

Um einem Endpunkt einen Benutzer hinzuzufügen, verwenden Sie den Befehl update-endpoint. Für den `--endpoint-request`-Parameter können Sie einen neuen Endpunkt definieren, der einen Benutzer enthalten kann. Um einen bestehenden Endpunkt zu aktualisieren, können Sie einfach auch die Attribute angeben, die Sie ändern möchten. Das folgende Beispiel fügt einen Benutzer zu einem vorhandenen Endpunkt hinzu, indem nur die Benutzerattribute angegeben werden:

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

Wobei gilt:

- *`application-id`* ist die ID des Amazon-Pinpoint-Projekts, in dem Sie einen Endpunkt hinzufügen oder aktualisieren.
- *`endpoint-id`* ist die ID, die Sie einem neuen Endpunkt zuweisen, oder die ID eines bestehenden Endpunkts, den Sie aktualisieren.
- *`endpoint-request-file.json`* ist der Dateipfad zu einer lokalen JSON-Datei, die die Eingabe für den Parameter `--endpoint-request` enthält.

Example Endpunkt-Anforderungsdatei

Der Beispielbefehl `update-endpoint` verwendet eine JSON-Datei als Argument für den Parameter `--endpoint-request`. Diese Datei enthält eine Benutzerdefinition wie die folgende:

```
{
```

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

Weitere Informationen zu den Attributen, die Sie verwenden können, um einen Benutzer zu definieren, finden Sie im `User`-Objekt im EndpointRequest-Schema in der Amazon-Pinpoint-API-Referenz.

AWS SDK for Java

Sie können die Amazon-Pinpoint-API in Ihren Java-Anwendungen verwenden, indem Sie den vom AWS SDK for Java bereitgestellten Client verwenden.

Example Code

Um einem Endpunkt einen Benutzer hinzuzufügen, initialisieren Sie ein EndpointRequest-Objekt und übergeben es der Methode `updateEndpoint` des `AmazonPinpoint`-Clients. Sie können dieses Objekt verwenden, um einen neuen Endpunkt zu definieren, der einen Benutzer enthalten kann. Um einen bestehenden Endpunkt zu aktualisieren, können Sie einfach auch die Eigenschaften aktualisieren, die Sie ändern möchten. Das folgende Beispiel fügt einem bestehenden Endpunkt einen Benutzer hinzu, indem ein EndpointUser-Objekt, dem EndpointRequest-Objekt hinzugefügt wird:

```
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);
        }
    }
```

Das vollständige SDK-Beispiel finden Sie unter AddExampleUser.java auf GitHub.

HTTP

Sie können Amazon Pinpoint verwenden, indem Sie HTTP-Anforderungen direkt an die REST-API stellen.

Example Anforderung zum Hinzufügen eines Endpunkts mit Benutzerdefinition

Um einem Endpunkt einen Benutzer hinzuzufügen, stellen Sie eine PUT-Anforderung an die Endpunkt-Ressource unter der folgenden URI:

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

Wobei gilt:

- *application-id* ist die ID des Amazon-Pinpoint-Projekts, in dem Sie einen Endpunkt hinzufügen oder aktualisieren.
- *endpoint-id* ist die ID, die Sie einem neuen Endpunkt zuweisen, oder die ID eines bestehenden Endpunkts, den Sie aktualisieren.

Fügen Sie Ihrer Anforderung die erforderlichen Header hinzu und geben Sie die EndpointRequest-JSON als Text an. Für den Anforderungstext können Sie einen neuen Endpunkt definieren, der einen Benutzer enthalten kann. Um einen bestehenden Endpunkt zu aktualisieren, können Sie einfach auch die Attribute angeben, die Sie ändern möchten. Das folgende Beispiel fügt einen Benutzer zu einem vorhandenen Endpunkt hinzu, indem nur die Benutzerattribute angegeben werden:

```
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"
        }
    }
}
```

Wenn die Anforderung erfolgreich ist, erhalten Sie eine Antwort wie die folgende:

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

Ähnliche Informationen

Weitere Informationen über die Endpunkt-Ressource in der Amazon-Pinpoint-API, einschließlich der unterstützten HTTP-Methoden und Anforderungsparameter, finden Sie unter Endpunkt in der Amazon-Pinpoint-API-Referenz.

Weitere Informationen zum Personalisieren von Nachrichten mit Variablen finden Sie unter Nachrichtenvariablen im Amazon-Pinpoint-Benutzerhandbuch.

Weitere Informationen, wie Sie ein Segment durch Importieren einer Liste mit Benutzer-IDs definieren, finden Sie unter Importieren von Segmenten im Amazon-Pinpoint-Benutzerhandbuch.

Informationen zum Senden einer Direktnachricht an bis zu 100 Benutzer-IDs finden Sie unter Benutzernachrichten in der Amazon-Pinpoint-API-Referenz.

Informationen zu den Kontingenten, die für Endpunkte gelten, einschließlich der Anzahl der Benutzerattribute, die Sie zuweisen können, finden Sie unter the section called “Endpunktkontingente”.

Hinzufügen eines Stapels Endpunkte zu Amazon Pinpoint

Sie können mehrere Endpunkte in einer einzigen Operation hinzufügen oder aktualisieren, indem Sie die Endpunkte in Stapeln übergeben. Jede Stapelanforderung kann bis zu 100 Endpunktdefinitionen enthalten.

Wenn Sie mehr als 100 Endpunkte in einer einzigen Operation hinzufügen oder aktualisieren möchten, lesen Sie stattdessen unter Importieren von Endpunkten in Amazon Pinpoint nach.

Beispiele

Die folgenden Beispiele zeigen Ihnen, wie Sie zwei Endpunkte auf einmal hinzufügen, indem Sie die Endpunkte in eine Stapelanforderung aufnehmen.

AWS CLI

Sie können Amazon Pinpoint verwenden, indem Sie Befehle mit der AWS CLI ausführen.

Example Befehl zum Aktualisieren von Endpunktstapeln

Um eine Endpunkt-Stapelanforderung zu senden, verwenden Sie den Befehl `update-endpoints-batch`:

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

Wobei gilt:

- *`application-id`* ist die ID des Amazon-Pinpoint-Projekts, in dem Sie die Endpunkte hinzufügen oder aktualisieren.
- *`endpoint_batch_request_file.json`* ist der Dateipfad zu einer lokalen JSON-Datei, die die Eingabe für den Parameter `--endpoint-batch-request` enthält.

Example Endpunktstapel-Anforderungsdatei

Der Beispielbefehl `update-endpoints-batch` verwendet eine JSON-Datei als Argument für den Parameter `--endpoint-request`. Diese Datei enthält einen Stapel von Endpunktdefinitionen wie die folgenden:

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

Weitere Informationen zu den Attributen, die Sie verwenden können, um einen Stapel Endpunkte zu definieren, finden Sie im EndpointBatchRequest-Schema in der Amazon-Pinpoint-API-Referenz.

AWS SDK for Java

Sie können die Amazon-Pinpoint-API in Ihren Java-Anwendungen verwenden, indem Sie den vom AWS SDK for Java bereitgestellten Client verwenden.

Example Code

Um eine Endpunkt-Stapelanforderung zu senden, initialisieren Sie ein `EndpointBatchRequest`-Objekt und übergeben es der Methode `updateEndpointsBatch` des `AmazonPinpoint`-Clients. Das folgende Beispiel füllt ein `EndpointBatchRequest`-Objekt mit zwei `EndpointBatchItem`-Objekten:

```
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);
                }
        }
```

```
}
```

Das vollständige SDK-Beispiel finden Sie unter AddExampleEndpoints.java auf GitHub.

HTTP

Sie können Amazon Pinpoint verwenden, indem Sie HTTP-Anforderungen direkt an die REST-API stellen.

Example Anforderung zum Ablegen von Endpunkten

Um eine Endpunkt-Stapelanforderung zu senden, stellen Sie eine PUT-Anforderung an die Endpunkte-Ressource unter der folgenden URI:

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

Wobei *application-id* die ID des Amazon-Pinpoint-Projekts ist, in dem Sie die Endpunkte hinzufügen oder aktualisieren.

Fügen Sie Ihrer Anforderung die erforderlichen Header hinzu und geben Sie die EndpointBatchRequest-JSON als Text an:

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

Wenn Ihre Anforderung erfolgreich ist, erhalten Sie eine Antwort wie die folgende:

```
{
```

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

Ähnliche Informationen

Weitere Informationen über die Endpunkt-Ressource in der Amazon-Pinpoint-API, einschließlich der unterstützten HTTP-Methoden und Anforderungsparameter, finden Sie unter Endpunkt in der Amazon-Pinpoint-API-Referenz.

Importieren von Endpunkten in Amazon Pinpoint

Sie können sehr viele Endpunkte hinzufügen oder aktualisieren, indem Sie sie aus einem Amazon-S3-Bucket importieren. Das Importieren von Endpunkten ist nützlich, wenn Sie die Aufzeichnungen zu Ihrer Zielgruppe außerhalb von Amazon Pinpoint vorliegen haben und diese Informationen einem Amazon-Pinpoint-Projekt hinzufügen möchten. In diesem Fall müssen Sie wie folgt verfahren:

1. Erstellen Sie Endpunktdefinitionen, die auf Ihren eigenen Zielgruppendaten basieren.
2. Speichern Sie diese Endpunktdefinitionen in einer oder mehreren Dateien, und laden Sie die Dateien in einen Amazon-S3-Bucket.
3. Fügen Sie Ihrem Amazon-Pinpoint-Projekt die Endpunkte hinzu, indem Sie sie aus dem Bucket importieren.

Jede Importauftrag kann bis zu 1 GB Daten übertragen. In einem typischen Auftrag, wobei jeder Endpunkt 4 KB oder kleiner ist, können Sie ca. 250.000 Endpunkte importieren. Sie können bis zu zwei gleichzeitige Importaufträge pro AWS-Konto ausführen. Wenn Sie mehr Bandbreite für Ihre Importaufträge benötigen, können Sie eine Erhöhung des Servicekontingents bei AWS Support anfordern. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung.

Bevor Sie beginnen

Bevor Sie Endpunkte importieren können, benötigen Sie die folgenden Ressourcen in Ihrem AWS-Konto:

- Ein Amazon-S3-Bucket Informationen zum Erstellen eines Buckets finden Sie unter Erstellen eines Buckets im Benutzerhandbuch für Amazon Simple Storage Service.

- Eine AWS Identity and Access Management (IAM)-Rolle, die Amazon Pinpoint Leseberechtigungen für Ihren Amazon-S3-Bucket erteilt. Weitere Informationen zum Erstellen der Rolle finden Sie unter IAM-Rolle für das Importieren von Endpunkten oder Segmenten.

Beispiele

Die folgenden Beispiele zeigen, wie Sie Ihrem Amazon-S3-Bucket Endpunktdefinitionen hinzufügen, und diese Endpunkte dann in ein Amazon-Pinpoint-Projekt importieren.

Dateien mit Endpunktdefinitionen

Die Dateien, die Sie Ihrem Amazon-S3-Bucket hinzufügen, können Endpunktdefinitionen im CSV- oder JSON-Format mit Trennung durch neue Zeilen enthalten. Weitere Informationen zu den Attributen, die Sie verwenden können, um Ihre Endpunkte zu definieren, finden Sie im EndpointRequest-JSON-Schema in der Amazon-Pinpoint-API-Referenz.

CSV

Sie können Endpunkte importieren, die in einer CSV-Datei definiert sind, wie im folgenden Beispiel dargestellt:

```
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
```

Bei der ersten Zeile handelt es sich um die Kopfzeile, die die Endpunkt-Attribute enthält. Geben Sie verschachtelte Attribute unter Verwendung der Punktnotation an, wie in `Location.Country` gezeigt.

Die nachfolgenden Zeilen definieren die Endpunkte anhand von Werten für jedes Attribut in der Kopfzeile.

Wenn Sie ein Komma oder doppelte Anführungszeichen in einen Wert einschließen möchten, verwenden Sie doppelte Anführungszeichen für den Wert, z. B. `"aaa,bbb"`.

Zeilenumbrüche werden innerhalb eines Werts in der CSV nicht unterstützt.

JSON

Sie können Endpunkte importieren, die in einer JSON-Datei mit Trennung durch neue Zeilen definiert sind, wie im folgenden Beispiel dargestellt:

```
{"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 diesem Format ist jede Zeile ein vollständiges JSON-Objekt mit einer einzelnen Endpunktdefinition.

Importauftrags-Anforderungen

Die folgenden Beispiele zeigen Ihnen, wie Sie Endpunktdefinitionen zu Amazon S3 hinzufügen, indem Sie eine lokale Datei in einen Bucket hochladen. Anschließend importieren die Beispiele die Endpunkt-Definitionen in ein Amazon-Pinpoint-Projekt.

AWS CLI

Sie können Amazon Pinpoint verwenden, indem Sie Befehle mit der AWS CLI ausführen.

Example S3 CP-Befehl

Um eine lokale Datei in einen Amazon-S3-Bucket hochzuladen, verwenden Sie den Amazon-S3-Befehl `cp`:

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

Wobei gilt:

- /endpoints-file ist der Dateipfad zu einer lokalen Datei, die die Endpunkt-Definitionen enthält.
- bucket-name/prefix/ ist der Name Ihres Amazon-S3-Buckets und optional ein Präfix, das Ihnen hilft, die Objekte in Ihrem Bucket hierarchisch zu organisieren. Ein praktisches Präfix könnte beispielsweise `pinpoint/imports/endpoints/` sein.

Example Befehl zum Erstellen eines Importauftrags

Um Endpunktdefinitionen aus einem Amazon-S3-Bucket zu importieren, verwenden Sie den folgenden `create-import-job`-Befehl:

```
$ 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
```

Wobei gilt:

- application-id ist die ID des Amazon-Pinpoint-Projekts, für das Sie Endpunkte importieren
- bucket-name/prefix/key ist der Speicherort in Amazon S3, der ein oder mehrere Objekte zum Importieren enthält. Der Speicherort kann mit dem Schlüssel für ein einzelnes Objekt enden, oder mit einem Präfix, das mehrere Objekte qualifiziert.
- iam-import-role-arn ist der Amazon-Ressourcenname (ARN) einer IAM-Rolle, die Amazon Pinpoint Lesezugriff auf den Bucket erteilt.
- Format kann `JSON` oder `CSV` sein, abhängig davon, mit welchem Format Sie Ihre Endpunkte definiert haben. Wenn der Amazon-S3-Speicherort mehrere Objekte mit gemischten Formaten enthält, importiert Amazon Pinpoint nur die Objekte, die mit dem angegebenen Format übereinstimmen.
- RegisterEndpoints kann entweder `true` oder `false` sein. Bei der Einstellung „true“ registriert der Importauftrag die Endpunkte mit Amazon Pinpoint, wenn die Endpunktdefinitionen importiert werden.

  Kombinationen aus RegisterEndpoints und DefineSegments

RegisterEndpoints	DefineSegments	Beschreibung
true	true	Amazon Pinpoint importiert die Endpunkte und erstellt ein Segment, das die Endpunkte enthält.

RegisterEndpoints	DefineSegments	Beschreibung
true	false	Amazon Pinpoint importiert die Endpunkte und erstellt kein Segment.
false	true	Amazon Pinpoint importiert die Endpunkte und erstellt ein Segment, das die Endpunkte enthält. Die Endpunkte werden nicht gespeichert und überschreiben keine vorhandenen Endpunkte.
false	false	Amazon Pinpoint lehnt diese Anfrage ab.

Die Antwort enthält Details über den Importauftrag:

```
{
    "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"
    }
}
```

Die Ausgabe stellt die Auftrags-ID mit dem `Id`-Attribut bereit. Diese ID können Sie verwenden, um den aktuellen Status des Importauftrags zu überprüfen.

Example Befehl zum Abrufen eines Importauftrags

Um den aktuellen Status eines Importauftrags zu überprüfen, verwenden Sie den Befehl `get-import-job`:

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

Wobei gilt:

- application-id ist die ID des Amazon-Pinpoint-Projekts, für das der Importauftrag initiiert wurde.
- job-id ist die ID des Importauftrags, den Sie überprüfen.

Die Ausgabe dieses Befehls stellt Informationen über den aktuellen Status des Importauftrags bereit:

```
{
    "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"
    }
}
```

Die Ausgabe stellt den Auftragsstatus mit dem `JobStatus`-Attribut bereit.

AWS SDK for Java

Sie können die Amazon-Pinpoint-API in Ihren Java-Anwendungen verwenden, indem Sie den vom AWS SDK for Java bereitgestellten Client verwenden.

Example Code

Um eine Datei mit Endpunktdefinitionen in Amazon S3 hochzuladen, verwenden Sie die Methode `putObject` des `AmazonS3`-Clients.

Um Endpunkte in ein Amazon-Pinpoint-Projekt zu importieren, initialisieren Sie ein `CreateImportJobRequest`-Objekt. Anschließend übergeben Sie dieses Objekt der Methode `createImportJob` des `AmazonPinpoint`-Clients.

```
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

Sie können Amazon Pinpoint verwenden, indem Sie HTTP-Anforderungen direkt an die REST-API stellen.

Example PUT-Anforderung für ein Objekt in S3

Um Ihre Endpunktdefinitionen einem Bucket hinzuzufügen, verwenden Sie die Amazon-S3-Operation PUT-Objekt und geben die Endpunktdefinitionen als Text an:

```
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"}}
```

Wobei gilt:

- /prefix/key ist das Präfix und den Schlüsselname für das Objekt, das die Endpunktdefinitionen nach dem Hochladen enthält. Sie können das Präfix verwenden, um Ihre Objekte hierarchisch zu organisieren. Ein praktisches Präfix könnte beispielsweise `pinpoint/imports/endpoints/` sein.
- bucket-name ist der Name des Amazon-S3-Buckets, dem Sie die Endpunktdefinitionen hinzufügen.

Example POST-Anforderung für den Importauftrag

Um Endpunktdefinitionen aus einem Amazon-S3-Bucket zu importieren, senden Sie eine POST-Anforderung an die Importaufträge-Ressource. Fügen Sie Ihrer Anforderung die erforderlichen Header hinzu und geben Sie die ImportJobRequest-JSON als Text an:

```
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
}
```

Wobei gilt:

- application-id ist die ID des Amazon-Pinpoint-Projekts, für das Sie Endpunkte importieren
- bucket-name/prefix/key ist der Speicherort in Amazon S3, der ein oder mehrere Objekte zum Importieren enthält. Der Speicherort kann mit dem Schlüssel für ein einzelnes Objekt enden, oder mit einem Präfix, das mehrere Objekte qualifiziert.
- iam-import-role-arn ist der Amazon-Ressourcenname (ARN) einer IAM-Rolle, die Amazon Pinpoint Lesezugriff auf den Bucket erteilt.
- Format kann `JSON` oder `CSV` sein, abhängig davon, mit welchem Format Sie Ihre Endpunkte definiert haben. Wenn der Amazon-S3-Speicherort mehrere Dateien mit gemischten Formaten enthält, importiert Amazon Pinpoint nur die Dateien, die mit dem angegebenen Format übereinstimmen.

Wenn Ihre Anforderung erfolgreich ist, erhalten Sie eine Antwort wie die folgende:

```
{
    "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
    }
}
```

Die Ausgabe stellt die Auftrags-ID mit dem `Id`-Attribut bereit. Diese ID können Sie verwenden, um den aktuellen Status des Importauftrags zu überprüfen.

Example GET-Anforderung für den Importauftrag

Um den aktuellen Status eines Importauftrags zu überprüfen, führen Sie eine `GET`-Anforderung für die Import Job-Ressource durch:

```
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
```

Wobei gilt:

- application_id ist die ID des Amazon-Pinpoint-Projekts, für das der Importauftrag initiiert wurde.
- job_id ist die ID des Importauftrags, den Sie überprüfen.

Wenn Ihre Anforderung erfolgreich ist, erhalten Sie eine Antwort wie die folgende:

```
{
    "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
    }
}
```

Die Ausgabe stellt den Auftragsstatus mit dem `JobStatus`-Attribut bereit.

Ähnliche Informationen

Weitere Informationen über die „Importaufträge“-Ressource in der Amazon-Pinpoint-API, einschließlich der unterstützten HTTP-Methoden und Anforderungsparameter, finden Sie unter Importaufträge in der Amazon-Pinpoint-API-Referenz.

Löschen von Endpunkten aus Amazon Pinpoint

Sie können Endpunkte löschen, wenn Sie keine Nachrichten mehr an ein bestimmtes Ziel senden wollen, z. B. wenn das Ziel nicht mehr erreichbar ist, oder wenn ein Kunde ein Konto schließt.

Beispiele

Die folgenden Beispiele zeigen Ihnen, wie Sie einen Endpunkt löschen.

AWS CLI

Sie können Amazon Pinpoint verwenden, indem Sie Befehle mit der AWS CLI ausführen.

Example Befehl zum Löschen eines Endpunkts

Zum Löschen eines Endpunkts verwenden Sie den Befehl `delete-endpoint`:

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

Wobei gilt:

- application-id ist die ID des Amazon-Pinpoint-Projekts, das den Endpunkt enthält.
- endpoint-id ist die ID des Endpunkts, den Sie löschen.

Die Ausgabe dieses Befehls ist die JSON-Definition des Endpunkts, den Sie gelöscht haben.

AWS SDK for Java

Sie können die Amazon-Pinpoint-API in Ihren Java-Anwendungen verwenden, indem Sie den vom AWS SDK for Java bereitgestellten Client verwenden.

Example Code

Um einen Endpunkt zu löschen, verwenden Sie die Methode `deleteEndpoint` des `AmazonPinpoint`-Clients. Geben Sie ein `DeleteEndpointRequest`-Objekt als Methodenargument an:

```
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");
    }
}
```

Das vollständige SDK-Beispiel finden Sie unter DeleteEndpoint.java auf GitHub.

HTTP

Sie können Amazon Pinpoint verwenden, indem Sie HTTP-Anforderungen direkt an die REST-API stellen.

Example DELETE-Anforderung zum Löschen eines Endpunkts

Um einen Endpunkt zu löschen, führen Sie eine DELETE-Anforderung an die Ressource Endpunkt aus:

```
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
```

Wobei gilt:

- application-id ist die ID des Amazon-Pinpoint-Projekts, das den Endpunkt enthält.
- endpoint-id ist die ID des Endpunkts, den Sie löschen.

Die Ausgabe dieser Anforderung ist die JSON-Definition des Endpunkts, den Sie gelöscht haben.

Verwaltung der maximalen Anzahl von Endpunkten eines Zielgruppenmitglieds

Jedes Mitglied Ihrer Zielgruppe kann maximal 15 Endpunkte mit seiner UserId verknüpft haben, siehe Endpunktkontingente. Wenn Sie versuchen, einen 16. Endpunkt hinzuzufügen, erhalten Sie je nach ChannelType entweder BadRequestException oder es gelingt, indem der Endpunkt mit dem ältesten EffectiveDate entfernt wird.

Hinzufügen eines 16. Endpunkts

- Wenn der neue Kanaltyp für den Endpunkt SMS, PUSH, VOICE, EMAIL, CUSTOM oder IN_APP ist, wird BadRequestException zurückgegeben, weil das Zielgruppenmitglied die maximale Anzahl von Endpunkten erreicht hat. Sie müssen einen Endpunkt entfernen, der dem Zielgruppenmitglied zugeordnet ist, und es erneut versuchen, siehe Löschen von Endpunkten aus Amazon Pinpoint.
- Wenn der neue Kanaltyp für den Endpunkt ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX oder BAIDU lautet:
  - Vergewissern Sie sich, dass mindestens ein Endpunkt, der derzeit mit dem Zielgruppenmitglied verknüpft ist, den ChannelType ADM, GCM, APNS, APNS_VOICE, APNS_VOIP_SANDBOX oder BAIDU hat. Ist dies nicht der Fall, wird BadRequestException zurückgegeben und ein Endpunkt muss entfernt werden, bevor Sie es erneut versuchen, siehe Löschen von Endpunkten aus Amazon Pinpoint.
  - Andernfalls wird der Endpunkt mit dem ältesten EffectiveDate auf `INACTIVE` gesetzt, wobei der ChannelType ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX oder BAIDU lautet.
    - Die UserId des alten Endpunkts wird entfernt.

- Der neue Endpunkt ist dem Zielgruppenmitglied zugeordnet und es hat immer noch die maximale Anzahl an Endpunkten.

Der Endpunkt kann erneut aktiviert werden, indem der Status auf `ACTIVE` gesetzt und die UserId wieder zum Endpunkt hinzugefügt wird.

Zugreifen auf Zielgruppendaten in Amazon Pinpoint

Wenn Sie Endpunkte zu Amazon Pinpoint hinzufügen, wird daraus ein zunehmendes Repository mit Zielgruppendaten. Diese Daten bestehen aus:

- Den Endpunkten, die Sie unter Verwendung der Amazon-Pinpoint-API hinzufügen oder aktualisieren.
- Die Endpunkte, die Ihr Client-Code hinzufügt oder aktualisiert, wenn Benutzer Ihre Anwendung besuchen.

So wie Ihre Zielgruppe wachsen und ändern sich auch Ihre Endpunktdaten. Um die neuesten Informationen anzuzeigen, die Amazon Pinpoint über Ihre Zielgruppe besitzt, können Sie Endpunkte einzeln abrufen oder alle Endpunkte für ein Amazon-Pinpoint-Projekt exportieren. Durch die Anzeige Ihrer Endpunktdaten erfahren Sie mehr über die Eigenschaften Ihrer Zielgruppe, die Sie in Ihren Endpunkten aufzeichnen, wie z. B.:

- Die Geräte und Plattformen Ihrer Benutzer.
- Die Zeitzonen Ihrer Benutzer.
- Welche Versionen Ihrer App auf den Geräten der Benutzer installiert sind.
- Die Standorte Ihrer Benutzer, wie z. B. ihre Städte oder Länder.
- Alle benutzerdefinierten Attribute oder Metriken, die Sie aufzeichnen.

Die Amazon-Pinpoint-Konsole bietet auch Analysen für demografische Informationen und benutzerdefinierte Attribute, die in Ihren Endpunkte erfasst werden.

Bevor Sie Endpunkte abfragen können, müssen Sie sie Ihrem Amazon-Pinpoint-Projekt hinzufügen. Weitere Informationen zum Hinzufügen von Endpunkten finden Sie unter Definieren Ihrer Zielgruppe für Amazon Pinpoint.

Verwenden Sie die Themen in diesem Abschnitt, um Endpunkte unter Verwendung der Amazon-Pinpoint-API abzurufen oder zu exportieren.

Themen

- Auflisten von Endpunkt-IDs mit Amazon Pinpoint

Suchen nach Endpunkten mit Amazon Pinpoint

Sie können die Details für einen einzelnen Endpunkt abrufen, der einem Amazon-Pinpoint-Projekt hinzugefügt wurde. Bei diesen Details kann es sich beispielsweise um die Zieladresse für Ihre Nachrichten, den Messaging-Kanal, Daten über das Gerät des Benutzers, Daten über den Standort des Benutzers und alle benutzerdefinierten Attribute, die Sie in Ihren Endpunkten aufzeichnen, handeln.

Um einen Endpunkt abzufragen, benötigen Sie die Endpunkt-ID. Wenn Sie die ID nicht kennen, können Sie die Endpunktdaten auch abfragen, indem Sie sie stattdessen exportieren. Weitere Informationen zum Exportieren von Endpunkten finden Sie unter the section called “Exportieren von Endpunkten”.

Beispiele

Die folgenden Beispiele zeigen Ihnen, wie Sie einen einzelnen Endpunkt abfragen, indem Sie seine ID angeben.

AWS CLI

Sie können Amazon Pinpoint verwenden, indem Sie Befehle mit der AWS CLI ausführen.

Example Befehl zum Anfordern eines Endpunkts

Zum Abrufen eines Endpunkts verwenden Sie den Befehl `get-endpoint`:

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

Wobei gilt:

- *`application-id`* ist die ID des Amazon-Pinpoint-Projekts, das den Endpunkt enthält.
- *`endpoint-id`* ist die ID des Endpunkts, den Sie abrufen.

Die Ausgabe dieses Befehls ist die JSON-Definition des Endpunkts, wie im folgenden Beispiel gezeigt:

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

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

AWS SDK for Java

Sie können die Amazon-Pinpoint-API in Ihren Java-Anwendungen verwenden, indem Sie den vom AWS SDK for Java bereitgestellten Client verwenden.

Example Code

Um einen Endpunkt abzurufen, initialisieren Sie ein `GetEndpointRequest`-Objekt. Anschließend übergeben Sie dieses Objekt der Methode `getEndpoint` des `AmazonPinpoint`-Clients:

```
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");
    }
}
```

Um die Endpunktdaten in einem lesbaren Format auszudrucken, verwendet dieses Beispiel die Google GSON-Bibliothek, um das `EndpointResponse`-Objekt in eine JSON-Zeichenfolge umzuwandeln.

HTTP

Sie können Amazon Pinpoint verwenden, indem Sie HTTP-Anforderungen direkt an die REST-API stellen.

Example GET-Endpunktanforderung

Um einen Endpunkt abzurufen, führen Sie eine `GET`-Anforderung an die Ressource Endpunkt aus:

```
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
```

Wobei gilt:

- *`application-id`* ist die ID des Amazon-Pinpoint-Projekts, das den Endpunkt enthält.
- *`endpoint-id`* ist die ID des Endpunkts, den Sie abrufen.

Die Ausgabe dieser Anforderung ist die JSON-Definition des Endpunkts, wie im folgenden Beispiel gezeigt:


```
{
    "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"
```

```
}
```

Ähnliche Informationen

Weitere Informationen zur Endpunkt-Ressource in der Amazon-Pinpoint-API finden Sie unter Endpunkt in der Amazon-Pinpoint-API-Referenz.

Exportieren von Endpunkten aus Amazon Pinpoint

Um alle Informationen zu erhalten, die Amazon Pinpoint über Ihre Zielgruppe besitzt, können Sie die zu einem Projekt gehörenden Endpunktdefinitionen exportieren. Beim Export legt Amazon Pinpoint die Endpunktdefinitionen in einen von Ihnen angegebenen Amazon-S3-Bucket ab. Das Exportieren von Endpunkten ist nützlich, wenn Sie Folgendes erledigen möchten:

- Anzeige der neuesten Daten über neue und bestehende Endpunkte, die Ihre Client-Anwendung bei Amazon Pinpoint registriert hat.
- Synchronisieren der Endpunktdaten in Amazon Pinpoint mit Ihrem eigenen CRM (Customer Relationship Management)-System.
- Berichte über Ihre Kundendaten erstellen oder diese analysieren.

Bevor Sie beginnen

Bevor Sie Endpunkte exportieren können, benötigen Sie die folgenden Ressourcen in Ihrem AWS-Konto:

- Ein Amazon-S3-Bucket Informationen zum Erstellen eines Buckets finden Sie unter Erstellen eines Buckets im Benutzerhandbuch für Amazon Simple Storage Service.
- Eine AWS Identity and Access Management (IAM)-Rolle, die Amazon Pinpoint Schreibberechtigungen für Ihren Amazon-S3-Bucket erteilt. Weitere Informationen zum Erstellen der Rolle finden Sie unter IAM-Rolle für das Exportieren von Endpunkten oder Segmenten.

Beispiele

Die folgenden Beispiele zeigen, wie Sie Endpunkte aus einem Amazon-Pinpoint-Projekt exportieren und diese Endpunkte anschließend aus Ihrem Amazon-S3-Bucket herunterladen.

AWS CLI

Sie können Amazon Pinpoint verwenden, indem Sie Befehle mit der AWS CLI ausführen.

Example Befehl zum Erstellen eines Exportauftrags

Zum Exportieren der Endpunkte in Ihrem Amazon-Pinpoint-Projekt verwenden Sie den Befehl `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
```

Wobei gilt:

- *`application-id`* ist die ID des Amazon-Pinpoint-Projekts, das die Endpunkte enthält.
- *`bucket-name/prefix/`* ist der Name Ihres Amazon-S3-Buckets und optional ein Präfix, das Ihnen hilft, die Objekte in Ihrem Bucket hierarchisch zu organisieren. Ein praktisches Präfix könnte beispielsweise `pinpoint/exports/endpoints/` sein.
- *`iam-export-role-arn`* ist der Amazon-Ressourcenname (ARN) einer IAM-Rolle, die Amazon Pinpoint Schreibzugriff auf den Bucket erteilt.

Die Ausgabe dieses Befehls enthält Details über den Exportauftrag:

```
{
    "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"
    }
}
```

Die Ausgabe stellt die Auftrags-ID mit dem `Id`-Attribut bereit. Diese ID können Sie verwenden, um den aktuellen Status des Exportauftrags zu überprüfen.

Example Befehl zum Abrufen eines Exportauftrags

Um den aktuellen Status eines Exportauftrags zu überprüfen, verwenden Sie den Befehl `get-export-job`:

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

Wobei gilt:

- *`application-id`* ist die ID des Amazon-Pinpoint-Projekts, aus dem Sie die Endpunkte exportiert haben.
- *`job-id`* ist die ID des Auftrags, den Sie überprüfen.

Die Ausgabe dieses Befehls stellt Informationen über den aktuellen Status des Exportauftrags bereit:

```
{
    "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"
    }
}
```

Die Ausgabe stellt den Auftragsstatus mit dem `JobStatus`-Attribut bereit. Wenn der Wert des Auftragsstatus gleich `COMPLETED` ist, können Sie ihre exportierten Endpunkte aus Ihrem Amazon-S3-Bucket abrufen.

Example S3 CP-Befehl

Um die exportierten Endpunkte herunterzuladen, verwenden Sie den Amazon-S3-Befehl cp:

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

Wobei gilt:

- *bucket-name/prefix/key* ist der Speicherort der .gz-Datei, die Amazon Pinpoint Ihrem Bucket hinzugefügt hat, als Sie Ihre Endpunkte exportiert haben. Diese Datei enthält die exportierten Endpunktdefinitionen. So ist in der URL `https://PINPOINT-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/Exports/example.csv` `PINPOINT-EXAMPLE-BUCKET` der Name des Buckets und `Exports/example.csv` der Schlüssel. Weitere Informationen zu Schlüsseln finden Sie unter Schlüssel im Amazon-S3-Benutzerhandbuch.
- */local/directory/* ist der Dateipfad zu dem lokalen Verzeichnis, in das Sie die Endpunkte herunterladen möchten.

AWS SDK for Java

Sie können die Amazon-Pinpoint-API in Ihren Java-Anwendungen verwenden, indem Sie den vom AWS SDK for Java bereitgestellten Client verwenden.

Example Code

Um Endpunkte aus einem Amazon-Pinpoint-Projekt zu exportieren, initialisieren Sie ein `CreateExportJobRequest`-Objekt. Anschließend übergeben Sie dieses Objekt der Methode `createExportJob` des `AmazonPinpoint`-Clients.

Um die exportierten Endpunkte von Amazon Pinpoint herunterzuladen, verwenden Sie die Methode `getObject` des `AmazonS3`-Clients.

```
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);
        }
    }
}
```

Das vollständige SDK-Beispiel finden Sie unter ExportEndpoints.java auf GitHub.

HTTP

Sie können Amazon Pinpoint verwenden, indem Sie HTTP-Anforderungen direkt an die REST-API stellen.

Example POST-Anforderung für einen Exportauftrag

Um die Endpunkte in Ihrem Amazon-Pinpoint-Projekt zu exportieren, führen Sie eine `POST`-Anforderung für die Ressource Exportaufträge aus:

```
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"
}
```

Wobei gilt:

- *`application-id`* ist die ID des Amazon-Pinpoint-Projekts, das die Endpunkte enthält.
- *`bucket-name/prefix`* ist der Name Ihres Amazon-S3-Buckets und optional ein Präfix, das Ihnen hilft, die Objekte in Ihrem Bucket hierarchisch zu organisieren. Ein praktisches Präfix könnte beispielsweise `pinpoint/exports/endpoints/` sein.
- *`iam-export-role-arn`* ist der Amazon-Ressourcenname (ARN) einer IAM-Rolle, die Amazon Pinpoint Schreibzugriff auf den Bucket erteilt.

Die Ausgabe dieser Anforderung enthält Details über den Exportauftrag:

```
{
    "Id": "611bdc54c75244bfa51fe7001ddb2e36",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-06T00:12:43.271Z",
```

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

Die Ausgabe stellt die Auftrags-ID mit dem `Id`-Attribut bereit. Diese ID können Sie verwenden, um den aktuellen Status des Exportauftrags zu überprüfen.

Example GET-Anforderung für einen Exportauftrag

Um den aktuellen Status eines Exportauftrags zu überprüfen, führen Sie eine `GET`-Anforderung für die Ressource Export job durch:

```
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
```

Wobei gilt:

- *`application-id`* ist die ID des Amazon-Pinpoint-Projekts, aus dem Sie die Endpunkte exportiert haben.
- *`job-id`* ist die ID des Auftrags, den Sie überprüfen.

Die Ausgabe dieser Anforderung stellt Informationen über den aktuellen Status des Exportauftrags bereit:

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

Die Ausgabe stellt den Auftragsstatus mit dem `JobStatus`-Attribut bereit. Wenn der Wert des Auftragsstatus gleich `COMPLETED` ist, können Sie ihre exportierten Endpunkte aus Ihrem Amazon-S3-Bucket abrufen.

Ähnliche Informationen

Weitere Informationen über die „Exportaufträge“-Ressource in der Amazon-Pinpoint-API, einschließlich der unterstützten HTTP-Methoden und Anforderungsparameter, finden Sie unter Exportaufträge in der Amazon-Pinpoint-API-Referenz.

Auflisten von Endpunkt-IDs mit Amazon Pinpoint

Um einen Endpunkt zu aktualisieren oder zu löschen, benötigen Sie die Endpunkt-ID. Wenn Sie also diese Operationen für alle Endpunkte eines Amazon-Pinpoint-Projekts durchführen möchten, ist der erste Schritt die Auflistung aller Endpunkt-IDs, die zu diesem Projekt gehören. Anschließend können Sie diese IDs durchlaufen, um beispielsweise Attribut global hinzuzufügen oder alle Endpunkte in Ihrem Projekt zu löschen.

Das folgende Beispiel verwendet AWS SDK for Java und führt die folgenden Schritte aus:

1. Ruft die Beispielmethode `exportEndpointsToS3` aus dem Beispielcode in Exportieren von Endpunkten aus Amazon Pinpoint auf. Diese Methode exportiert die Endpunktdefinitionen aus einem Amazon-Pinpoint-Projekt. Die Endpunktdefinitionen werden als gzip-Dateien einem Amazon-S3-Bucket hinzugefügt.
2. Lädt die exportierten gzip-Dateien herunter.
3. Liest die gzip-Dateien und erhält die Endpunkt-ID aus der JSON-Definition der einzelnen Endpunkte.
4. Gibt die Endpunkt-IDs auf der Konsole aus.
5. Bereinigt die Daten durch Löschen der Dateien, die Amazon Pinpoint Amazon S3 hinzugefügt hat.

```
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);
        }
    }
}
```

Das vollständige SDK-Beispiel finden Sie unter ListEndpointIs.java auf GitHub.

Erstellen von Segmenten

Ein Benutzersegment stellt einen Teil Ihrer Benutzer basierend auf gemeinsamen Merkmalen dar, z. B. Zeitpunkt der letzten App-Nutzung oder verwendete Geräteplattform. Ein Segment legt fest, welche Benutzer die Nachrichten einer Kampagne erhalten. Definieren Sie Segmente so, dass Sie die richtige Zielgruppe erreichen, wenn Sie Benutzer zur Nutzung Ihrer App motivieren, Sonderangebote unterbreiten oder anderweitig die Nutzereinbindung und das Kaufverhalten fördern möchten.

Nachdem Sie ein Segment erstellt haben, können Sie es in einer oder mehreren Kampagnen verwenden. Eine Kampagne stellt den Benutzern im Segment maßgeschneiderte Nachrichten zu.

Weitere Informationen hierzu finden Sie unter Segmente.

Themen

Erstellen von Segmenten

Um die gewünschte Zielgruppe für eine Kampagne zu erreichen, erstellen Sie ein Segment auf Grundlage der Daten, die von Ihrer App gemeldet wurden. Um beispielsweise Benutzer zu erreichen, die Ihre App in letzter Zeit nicht verwendet haben, können Sie ein Segment für Benutzer definieren, die Ihre App in den letzten 30 Tagen nicht genutzt haben.

Erstellen von Segmenten mit dem AWS SDK for Java

Das folgende Beispiel zeigt, wie ein Segment mit dem AWS SDK for Java erstellt wird.

```
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;
        }
}
```

Wenn Sie dieses Beispiel ausführen, wird Folgendes im Konsolenfenster Ihrer IDE ausgegeben:

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

Das vollständige SDK-Beispiel finden Sie unter CreateSegment.java auf GitHub.

Importieren von Segmenten

Sie können in Amazon Pinpoint ein Benutzersegment definieren, indem Sie Informationen zu den Endpunkten importieren, die zum Segment gehören. Ein Endpunkt ist ein einzelnes Messaging-Ziel, wie ein Geräte-Token für mobile Push-Benachrichtigungen, eine Mobiltelefonnummer oder eine E-Mail-Adresse.

Das Importieren von Segmenten ist nützlich, wenn Sie Ihre Benutzer bereits außerhalb von Amazon Pinpoint segmentiert haben, Ihre Benutzer jedoch in Amazon-Pinpoint-Kampagnen einbinden möchten.

Wenn Sie ein Segment importieren, ruft Amazon Pinpoint die Endpunkte des Segments von Amazon Simple Storage Service (Amazon S3) ab. Vor dem Importieren fügen Sie die Endpunkte zu Amazon S3 hinzu und erstellen eine IAM-Rolle, die Amazon Pinpoint Zugriff auf Amazon S3 gewährt. Anschließend teilen Sie Amazon Pinpoint den Amazon-S3-Standort mit, an dem die Endpunkte gespeichert sind, und Amazon Pinpoint fügt jeden Endpunkt dem Segment hinzu.

Informationen zum Erstellen der IAM-Rolle finden Sie unter IAM-Rolle für das Importieren von Endpunkten oder Segmenten. Weitere Informationen zum Importieren von Segmenten über die Amazon-Pinpoint-Konsole finden Sie unter Importieren von Segmenten im Amazon-Pinpoint-Benutzerhandbuch.

Importieren von Segmenten

Das folgende Beispiel zeigt, wie Sie ein Segment über AWS SDK for Java importieren.

```
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;
    }
}
```

Das vollständige SDK-Beispiel finden Sie unter ImportingSegments.java auf GitHub.

Anpassung von Segmenten mit AWS Lambda

Dies ist die Vorabdokumentation eines Features, das als öffentliche Beta-Version vorliegt. Änderungen sind vorbehalten.

Sie können AWS Lambda verwenden, um anzupassen, wie eine Amazon-Pinpoint-Kampagne Ihrer Zielgruppe anspricht. Mit AWS Lambda können Sie das Kampagnensegment an dem Zeitpunkt ändern, an dem Amazon Pinpoint die Kampagnennachricht sendet.

AWS Lambda ist ein Datenverarbeitungsservice, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Sie packen Ihren Code und laden ihn als Lambda-Funktionen in Lambda hoch. Lambda führt eine Funktion aus, wenn die Funktion ausgerufen wird, was manuell durch Sie oder automatisch als Reaktion auf Ereignisse passieren kann. Weitere Informationen finden Sie im AWS Lambda-Entwicklerhandbuch.

Um einer Kampagne eine Lambda-Funktion zuzuweisen, definieren Sie die `CampaignHook`-Einstellungen der Kampagne unter Verwendung der Ressource Kampagne in der Amazon-Pinpoint-API. Diese Einstellungen schließen den Namen der Lambda-Funktion ein. Sie schließen auch den `CampaignHook`-Modus ein, der festlegt, ob Amazon Pinpoint einen Rückgabewert aus der Funktion erhält.

Eine Lambda-Funktion, die Sie einer Kampagne zuweisen, wird als Amazon-Pinpoint-Erweiterung bezeichnet.

Mit den definierten `CampaignHook`-Einstellungen ruft Amazon Pinpoint automatisch die Lambda-Funktion auf, wenn die Kampagne ausgeführt wird, bevor die Nachricht der Kampagne gesendet wird. Wenn Amazon Pinpoint die Funktion aufruft, stellt es Ereignisdaten über die Nachrichtenzustellung bereit. Diese Daten schließen das Kampagnensegment ein. Dies ist die Liste der Endpunkte, an die Amazon Pinpoint die Nachricht sendet.

Wenn der `CampaignHook`-Modus auf `FILTER` gesetzt ist, gestattet Amazon Pinpoint der Funktion, das Segment zu ändern und zurückzugeben, bevor die Nachricht gesendet wird. Beispielsweise könnte die Funktion die Endpunktdefinitionen mit Attributen aktualisieren, die Daten aus einer Quelle außerhalb von Amazon Pinpoint enthalten. Die Funktion könnte das Segment auch filtern, indem bestimmte Endpunkte abhängig von den Bedingungen in Ihrem Funktionscode entfernt werden. Nachdem Amazon Pinpoint das geänderte Segment aus Ihrer Funktion erhalten hat, sendet es

die Nachricht an alle Endpunkte des Segments unter Verwendung des Bereitstellungskanals der Kampagne.

Durch die Verarbeitung Ihrer Segmente mit AWS Lambda haben Sie mehr Kontrolle darüber, an wen Sie Nachrichten senden, und was diese Nachrichten enthalten. Sie können Ihre Kampagnen in Echtzeit anpassen, wenn die Nachrichten der Kampagne Nachrichten gesendet werden. Das Filtern von Segmenten ermöglicht Ihnen, präziser definierte Untergruppen Ihrer Segmente anzusprechen. Das Hinzufügen oder Aktualisieren von Endpunkt-Attributen ermöglicht Ihnen, neue Daten für Nachrichtenvariablen bereitzustellen.

> **Note**
>
> Sie können auch die `CampaignHook`-Einstellungen verwenden, um eine Lambda-Funktion zuzuweisen, die die Nachrichtenzustellung verarbeitet. Diese Art Funktion ist nützlich, wenn Nachrichten über benutzerdefinierte Kanäle zugestellt werden sollen, die Amazon Pinpoint nicht unterstützt, z. B. Social Media-Plattformen. Weitere Informationen finden Sie unter Erstellen von benutzerdefinierten Kanälen in Amazon Pinpoint.
> Wenn Sie einen Lambda-Hook mit Amazon Pinpoint aufrufen, muss sich die Lambda-Funktion auch in derselben Region wie das Amazon-Pinpoint-Projekt befinden.

Um Kampagnensegmente mit AWS Lambda zu ändern, erstellen Sie zuerst eine Funktion, die von Amazon Pinpoint gesendete Ereignisdaten verarbeitet und ein geändertes Segment zurückgibt. Anschließend autorisieren Sie Amazon Pinpoint, die Funktion aufzurufen, indem Sie eine Lambda-Funktionsrichtlinie zuweisen. Schließlich weisen Sie der Funktion eine oder mehrere Kampagnen zu, indem Sie die `CampaignHook`-Einstellungen definieren.

Ereignisdaten

Wenn Amazon Pinpoint Ihre Lambda-Funktion aufruft, stellt es die folgende Nutzlast als Ereignisdaten bereit:

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

```
  "Endpoints": {
    EndpointId: {Endpoint definition}
    . . .
  }
}
```

AWS Lambda übergibt die Ereignisdaten an Ihren Funktionscode. Die Ereignisdaten stellen die folgenden Attribute bereit:

- `MessageConfiguration`: Hat dieselbe Struktur wie das `DirectMessageConfiguration`-Objekt der Nachrichten-Ressource in der Amazon-Pinpoint-API.
- `ApplicationId`: Die ID des Amazon-Pinpoint-Projekts, zu dem die Kampagne gehört.
- `CampaignId`: Die ID der Amazon-Pinpoint-Kampagne, für die die Funktion aufgerufen wurde.
- `TreatmentId`: Die ID einer Kampagnenvariante, die für A/B-Tests verwendet wird.
- `ActivityId`: Die ID der Aktivität, die von der Kampagne ausgeführt wird.
- `ScheduledTime`: Datum/Uhrzeit, an dem/zu der die Kampagnennachrichten im Format ISO 8601 zugestellt werden.
- `Endpoints`: Eine Karte, die Endpunkt-IDs Endpunktdefinitionen zuordnet. Jede Ereignisdaten-Nutzlast enthält bis zu 50 Endpunkte. Wenn das Kampagnensegment mehr als 50 Endpunkte enthält, ruft Amazon Pinpoint die Funktion wiederholt mit bis zu 50 Endpunkten auf, bis alle Endpunkte verarbeitet wurden.

Erstellen einer Lambda-Funktion

Informationen zum Erstellen einer Lambda-Funktion finden Sie unter Erste Schritte im AWS Lambda-Entwicklerhandbuch. Denken Sie bei der Erstellung der Funktion daran, dass die Nachrichtenzustellung unter folgenden Bedingungen fehlschlägt:

- Die Lambda-Funktion benötigt mehr als 15 Sekunden, um das modifizierte Segment zurückzugeben.
- Amazon Pinpoint kann den Rückgabewert der Funktion nicht decodieren.
- Die Funktion benötigt mehr als 3 Versuche von Amazon Pinpoint für einen erfolgreichen Aufruf.

Amazon Pinpoint akzeptiert nur Endpunktdefinitionen im Rückgabewert der Funktion. Die Funktion kann keine anderen Elemente in den Ereignisdaten ändern.

Beispiel-Lambda-Funktion

Ihre Lambda-Funktion verarbeitet die von Amazon Pinpoint gesendeten Ereignisdaten und gibt die geänderten Endpunkte zurück, wie in der folgenden Beispielprozedur gezeigt, die in Node.js geschrieben ist:

```
'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 übergibt die Ereignisdaten der Prozedur als `event`-Parameter.

In diesem Beispiel durchläuft die Prozedur jeden Endpunkt im `event.Endpoints`-Objekt und fügt dem Endpunkt ein neues Attribut hinzu, `CreditScore`. Der Wert des `CreditScore`-Attributs ist einfach eine Zufallszahl.

Die `console.log()`-Anweisung protokolliert das Ereignis in CloudWatch Logs.

Die `callback()`-Anweisung gibt die geänderten Endpunkte an Amazon Pinpoint zurück. Normalerweise ist der `callback`-Parameter optional in Node.js Lambda-Funktionen, aber in diesem Kontext ist er erforderlich, da die Funktion die aktualisierten Endpunkte an Amazon Pinpoint zurückgeben muss.

Ihre Funktion muss Endpunkte im selben Format zurückgeben, wie von den Ereignisdaten bereitgestellt. Dabei handelt es sich um ein Schema, das Endpunkt-IDs Endpunktdefinitionen zuweist, wie im folgenden Beispiel:

```
{
```

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

Die Beispiel-Funktion ändert das `event.Endpoints`-Objekt, das sie in den Ereignisdaten erhalten hat, und gibt es zurück.

Optional können Sie die Attribute `TitleOverride` und `BodyOverride` in die Endpunktdefinitionen aufnehmen, die Sie zurückgeben.

> **Note**
>
> Wenn Sie mit dieser Lösung Nachrichten senden, berücksichtigt Amazon Pinpoint die Attribute `TitleOverride` und `BodyOverride` nur für Endpunkte, bei denen der Wert des `ChannelType`-Attributs einen der folgenden Werte hat: `ADM`, `APNS`, `APNS_SANDBOX`, `APNS_VOIP`, `APNS_VOIP_SANDBOX`, `BAIDU`, `GCM` oder `SMS`.
> Amazon Pinpoint berücksichtigt nicht diese Attribute für Endpunkte, bei denen der Wert des `ChannelType`-Attributs `EMAIL` ist.

Zuweisen einer Lambda-Funktionsrichtlinie

Bevor Sie Ihre Lambda-Funktion verwenden können, um Ihre Endpunkte zu verarbeiten, müssen Sie Amazon Pinpoint autorisieren, Ihre Lambda-Funktion aufzurufen. Um eine Aufrufberechtigung zu erteilen, weisen Sie der Funktion eine Lambda-Funktionsrichtlinie zu. Eine Lambda-Funktionsrichtlinie ist eine ressourcenbasierte Richtlinie, die bestimmt, welche Entitäten Ihre Funktion verwenden dürfen, und welche Aktionen diese Entitäten durchführen können.

Weitere Informationen finden Sie unter Verwenden von ressourcenbasierten Richtlinien für AWS Lambda im AWS Lambda-Entwicklerhandbuch.

Beispiel für eine Funktionsrichtlinie

Mit der folgenden Richtlinie wird dem Amazon-Pinpoint-Service-Prinzipal die Berechtigung erteilt, die `lambda:InvokeFunction`-Aktion für eine bestimmte Kampagne (*`campaign-id`*) zu verwenden:

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

Ihre Funktionsrichtlinie benötigt einen `Condition`-Block, der einen `AWS:SourceArn`-Schlüssel beinhaltet. Dieser Code gibt an, welche Amazon-Pinpoint-Kampagne die Funktion aufrufen darf. In diesem Beispiel erteilt die Richtlinie eine Berechtigung für nur eine einzelne Kampagne. Der `Condition`-Block muss auch einen `AWS:SourceAccount`-Schlüssel enthalten, der steuert, welches AWS-Konto die Aktion aufrufen kann.

Um eine allgemeinere Richtlinie zu schreiben, verwenden Sie Wildcards (*), die mehrere Zeichen darstellen. Beispielsweise können Sie den folgenden `Condition`-Block verwenden, um einer beliebigen Kampagne in einem bestimmten Amazon-Pinpoint-Projekt (*`application-id`*) zu gestatten, die Funktion aufzurufen:

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

Wenn Sie die Lambda-Funktion als Standardfunktion verwenden möchten, die von allen Kampagnen für ein Projekt verwendet wird, empfehlen wir, den `Condition`-Block für die Richtlinie auf die zuvor beschriebene Weise zu konfigurieren. Informationen zum Festlegen einer Lambda-Funktion als Standard für alle Kampagnen in einem Projekt finden Sie unter Zuweisen einer Lambda-Funktion zu einer Kampagne.

Erteilen der Amazon-Pinpoint-Aufrufberechtigung

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Berechtigungen für die Lambda-Funktionsrichtlinie hinzuzufügen, die Ihrer Lambda-Funktion zugewiesen ist. Um Amazon Pinpoint zu erlauben, eine Funktion für eine bestimmte Kampagne aufzurufen, verwenden Sie den Lambda-Befehl `add-permission`, wie im folgenden Beispiel gezeigt:

```
$ 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
```

Sie können Ihre Kampagnen-IDs mit dem Befehl get-campaigns in der AWS CLI nachschlagen. Sie können Ihre Anwendungs-ID auch mit dem Befehl get-apps nachschlagen.

Wenn Sie den Lambda-Befehl `add-permission` ausführen, gibt Lambda die folgende Ausgabe zurück:

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

Der `Statement`-Wert ist eine JSON-Zeichenfolgenversion der Anweisung, die der Lambda-Funktionsrichtlinie hinzugefügt wurde.

Zuweisen einer Lambda-Funktion zu einer Kampagne

Sie können eine Lambda-Funktion einer einzelnen Amazon-Pinpoint-Kampagne zuweisen. Sie können auch die Lambda-Funktion als Standard festlegen, der von allen Kampagnen für ein Projekt verwendet wird, außer für Kampagnen, für die Sie separat eine Funktion zuweisen.

Um eine Lambda-Funktion einer individuellen Kampagne zuzuweisen, verwenden Sie die Amazon-Pinpoint-API, um ein `Campaign`-Objekt zu erstellen oder zu aktualisieren und sein `CampaignHook`-Attribut zu definieren. Um eine Lambda-Funktion als Standard für alle Kampagnen in einem Projekt festzulege, erstellen Sie die `Settings`-Ressource für dieses Projekt und definieren ihr `CampaignHook`-Objekt.

In beiden Fällen legen Sie die folgenden `CampaignHook`-Attribute fest:

- `LambdaFunctionName`: Der Name oder ARN der Lambda-Funktion, die Amazon Pinpoint aufruft, bevor es Nachrichten für die Kampagne sendet.
- `Mode` – Eingestellt auf `FILTER`. Mit diesem Modus ruft Amazon Pinpoint die Funktion auf und wartet, bis sie die geänderten Endpunkte zurückgibt. Nachdem es sie erhalten hat, sendet Amazon

Pinpoint die Nachricht. Amazon Pinpoint wartet bis zu 15 Sekunden, bevor die Zustellung von Nachrichten als fehlgeschlagen betrachtet wird.

Mit für eine Kampagne definierten `CampaignHook`-Einstellungen ruft Amazon Pinpoint die angegebene Lambda-Funktion auf, bevor die Nachrichten der Kampagne gesendet werden. Amazon Pinpoint wartet, bis es die geänderten Endpunkte von der Funktion erhalten hat. Wenn Amazon Pinpoint die aktualisierten Endpunkte empfängt, setzt es die Nachrichtenzustellung unter Verwendung der aktualisierten Endpunktdaten fort.

Erstellen von Kampagnen

Um die Interaktion zwischen Ihrer App und deren Benutzer zu steigern, erstellen und verwalten Sie mit Amazon Pinpoint Push-Benachrichtigungskampagnen, die bestimmte Benutzersegmente erreichen.

Beispiel: Ihre Kampagne kann Benutzer, die die App seit längerem nicht ausgeführt haben, dazu anregen, zur App zurückzukehren, oder Benutzern, die seit einiger Zeit keinen Kauf mehr getätigt haben, Sonderangebote unterbreiten.

Eine Kampagne sendet eine maßgeschneiderte Nachricht an ein von Ihnen definiertes Benutzersegment. Die Kampagne kann die Nachricht an alle Benutzer im Segment senden oder Sie können unter den Benutzern einen Prozentsatz bestimmen, der keine Nachrichten erhält.

Sie können die Kampagne so planen, dass die Nachricht einmalig oder regelmäßig (z. B. einmal pro Woche) gesendet wird. Um zu verhindern, dass Benutzer die Meldung zu einem ungünstigen Zeitpunkt erhalten, kann der Zeitplan eine Leerlaufzeit umfassen, während der keine Nachrichten gesendet werden.

Um mit alternativen Kampagnenstrategien zu experimentieren, konfigurieren Sie Ihre Kampagne als A/B-Test. Dieser beinhaltet zwei oder mehr Behandlungen der Nachricht oder des Zeitplans. Behandlungen sind Varianten der Nachricht oder des Zeitplans. Wenn die Benutzer auf die Kampagne reagieren, können Sie anhand der Kampagnen-Analysen die Effektivität jeder Behandlung vergleichen.

Weitere Informationen finden Sie unter Kampagnen.

Erstellen von Standardkampagnen

Eine Standardkampagne sendet eine benutzerdefinierte Push-Benachrichtigung an ein bestimmtes Segment entsprechend einem von Ihnen festgelegten Zeitplan.

Kampagnen erstellen mit dem AWS SDK for Java

Das folgende Beispiel veranschaulicht das Erstellen einer Kampagne mit dem AWS SDK for Java.

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

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

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

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

        final String usage = """

                Usage:   <appId> <segmentId>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

        return null;
    }
}
```

Wenn Sie dieses Beispiel ausführen, wird Folgendes im Konsolenfenster Ihrer IDE ausgegeben:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Das vollständige SDK-Beispiel finden Sie unter CreateCampaign.java on GitHub.

Erstellen von A/B-Test-Kampagnen

Ein A/B-Test verhält sich wie eine Standardkampagne, ermöglicht Ihnen jedoch, unterschiedliche Behandlungen für die Nachricht oder den Zeitplan der Kampagne zu definieren.

Erstellen von A/B-Testkampagnen mit dem AWS SDK for Java

Das folgende Beispiel veranschaulicht das Erstellen einer A/B-Test-Kampagne mit dem AWS SDK for Java.

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

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

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

public class PinpointCampaignSample {

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

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

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

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

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

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

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

        CreateCampaignResult result = client.createCampaign(createCampaignRequest);

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

        return result.getCampaignResponse();
    }

}
```

Wenn Sie dieses Beispiel ausführen, wird Folgendes im Konsolenfenster Ihrer IDE ausgegeben:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Verwenden der Amazon-Pinpoint-SMS- und -Sprachnachrichten-API, Version 2

> **Note**
>
> Amazon Pinpoint hat die Dokumentation seines Benutzerhandbuchs aktualisiert. Aktuelle Informationen zur Erstellung, Konfiguration und Verwaltung Ihrer Amazon-Pinpoint-SMS- und -Sprachressourcen finden Sie im neuen Amazon-Pinpoint-SMS-Benutzerhandbuch.
> Das folgende Thema wurde in das neue Amazon Pinpoint SMS-Benutzerhandbuch verschoben.
>
> - Verwaltung von Telefonnummern
> - Verwaltung von Absender-IDs
> - Pools verwalten
> - Verwaltung von Opt-Out-Listen
> - Konfigurationssätze verwalten
> - Verwaltung von Schlüsselwörtern
> - Verwaltung von Veranstaltungszielen
> - Senden von Nachrichten

Amazon Pinpoint enthält eine API (als SMS- und Sprachnachrichten-API, Version 2, bezeichnet), die für das Senden von SMS- und Sprachnachrichten konzipiert wurde. Während sich die Amazon-Pinpoint-API auf das Senden von Nachrichten im Rahmen von geplanten und ereignisgesteuerten Kampagnen und Journeys konzentriert, bietet die SMS- und Sprachnachrichten-API neue Features und Möglichkeiten für den direkten Versand von SMS- und Sprachnachrichten an einzelne Empfänger. Sie können die SMS- und Sprachnachrichten-API unabhängig von den Kampagnen- und Journey-Features von Amazon Pinpoint verwenden oder Sie können beide gleichzeitig verwenden, um unterschiedlichen Anwendungsfällen gerecht zu werden. Wenn Sie Amazon Pinpoint bereits zum Senden von SMS oder Sprachnachrichten verwenden, ist Ihr Konto bereits für die Verwendung dieser API konfiguriert.

Diese API ist eine gute Lösung für Benutzer mit einer Mehrmandantenarchitektur, wie z. B. unabhängige Softwareanbieter (ISVs). Mit dieser API kann einfacher sichergestellt werden, dass

Ereignisdaten, Ursprungstelefonnummern und Opt-Out-Listen für verschiedene Mandanten getrennt sind.

Wenn Sie die SMS- und Sprachnachrichten-API verwenden, empfehlen wir Ihnen, Konfigurationssätze und Ereignisziele einzurichten. Die SMS- und Sprachnachrichten-API gibt nicht automatisch Ereignisdaten für die von Ihnen gesendeten Nachrichten aus. Durch die Einrichtung von Ereigniszielen wird sichergestellt, dass Sie wichtige Ereignisdaten wie Nachrichtenzustellung und Fehlerereignisse erfassen.

Version 2 dieser API ging Version 1 voraus. Wenn Sie derzeit Version 1 dieser API verwenden, ist sie weiterhin verfügbar und Sie können sie weiterhin verwenden. Wenn Sie jedoch auf Version 2 migrieren, erhalten Sie zusätzliche Features, z. B. die Möglichkeit, Telefonnummernpools zu erstellen, neue Telefonnummern programmgesteuert anzufordern und bestimmte Funktionen von Telefonnummern zu aktivieren oder zu deaktivieren.

> **Note**
>
> Es gibt einige Aufgaben, die derzeit nur mit der Amazon-Pinpoint-Konsole erledigt werden können. Wenn Sie beispielsweise eine Telefonnummer verifizieren möchten, die Sie verwenden möchten, während sich Ihr Konto in der SMS-Sandbox befindet, oder wenn Sie sich für die Nutzung von 10DLC registrieren möchten, müssen Sie die Amazon-Pinpoint-Konsole verwenden.

Dieser Abschnitt enthält Informationen zu dieser API sowie Beispiele für deren Verwendung. Eine Referenzdokumentation finden Sie auch in der API-Referenz für SMS und Sprache, Version 2.

Senden und Überprüfen von Einmalpasswörtern (OTPs) mit Amazon Pinpoint

Amazon Pinpoint beinhaltet eine Funktion zur Verwaltung von Einmalpasswörtern (OTP). Sie können dieses Feature verwenden, um neue Einmalpasswörter zu generieren und diese als SMS-Nachrichten an Ihre Empfänger zu senden. Ihre Anwendungen können dann die Amazon-Pinpoint-API aufrufen, um diese Passwörter zu überprüfen.

Important

Um diese Funktion nutzen zu können, muss Ihr Konto über Produktionszugriff und eine aktive Originationsidentität verfügen. Weitere Informationen finden Sie unter Über die Amazon Pinpoint SMS-Sandbox und Telefonnummer anfordern im Amazon Pinpoint SMS-Benutzerhandbuch.

In einigen Ländern und Regionen müssen Sie sich eine spezielle Telefonnummer oder Ursprungs-ID besorgen, bevor Sie SMS-Nachrichten versenden können. Wenn Sie beispielsweise Nachrichten an Empfänger in den USA senden, benötigen Sie eine eigene gebührenfreie Nummer, eine 10DLC-Nummer oder eine Kurzwahlnummer. Wenn Sie Nachrichten an Empfänger in Indien senden, benötigen Sie eine registrierte Absender-ID, die eine Principal Entity ID (PEID) und eine Vorlagen-ID umfasst. Diese Anforderungen gelten weiterhin, wenn Sie das OTP-Feature verwenden.

Um dieses Feature verwenden zu können, benötigen Sie Berechtigungen zum Senden und Überprüfen von OTP-Nachrichten, siehe. Einmalpasswörter Wenn Sie Hilfe beim Ermitteln von Berechtigungen benötigen, finden Sie weitere Informationen unter Fehlerbehebung der Identitäts- und Zugriffsverwaltung für Amazon Pinpoint.

Senden einer OTP-Nachricht

Sie können die `SendOtpMessages`-Operation in der Amazon-Pinpoint-API verwenden, um einen OTP-Code an einen Benutzer Ihrer Anwendung zu senden. Wenn Sie diese API verwenden, generiert Amazon Pinpoint einen zufälligen Code und sendet ihn als SMS-Nachricht an Ihren Benutzer. Ihre Anfrage kann die folgenden Parameter enthalten:

- `Channel`: Der Kommunikationskanal, über den der OTP-Code gesendet wird. Derzeit werden nur SMS-Nachrichten unterstützt, sodass der einzig akzeptable Wert SMS ist.

- `BrandName`: Der Name der Marke, des Unternehmens oder des Produkts, das dem OTP-Code zugeordnet ist. Dieser Name kann bis zu 20 Zeichen umfassen.

  > **Note**
  >
  > Wenn Amazon Pinpoint die OTP-Nachricht sendet, wird der Markenname automatisch in die folgende Nachrichtenvorlage eingefügt:
  >
  > ```
  > This is your One Time Password: {{otp}} from {{brand}}
  > ```
  >
  > Wenn Sie also Ihren Markennamen angeben ExampleCorp und Amazon Pinpoint ein Einmalpasswort von 123456 generiert, sendet Amazon die folgende Nachricht an Ihren Benutzer:
  >
  > ```
  > This is your One Time Password: 123456 from ExampleCorp
  > ```

- `CodeLength`: Die Anzahl der Ziffern, die der OTP-Code enthalten soll, der an den Empfänger gesendet wird. OTP-Codes können zwischen 5 und einschließlich 8 Ziffern enthalten.
- `ValidityPeriod`: Die Dauer in Minuten, in der der OTP-Code gültig ist. Die Gültigkeitsdauer kann zwischen 5 und einschließlich 60 Minuten liegen.
- `AllowedAttempts`: Wie oft der Empfänger erfolglos versuchen kann, das Einmalpasswort zu verifizieren. Wenn die Anzahl der Versuche diesen Wert überschreitet, wird das OTP automatisch ungültig. Die maximale Anzahl der zulässigen Versuche beträgt 5.
- `Language`: Die Sprache im IETF BCP-47-Format, die beim Senden der Nachricht verwendet werden soll. Zulässige Werte sind:
  - `de-DE` – Deutsch
  - `en-GB` – Englisch (UK)
  - `en-US` – Englisch (USA)
  - `es-419` – Spanisch (lateinamerikanisch)
  - `es-ES` – Spanisch
  - `fr-CA` – Französisch (Kanada)
  - `fr-FR` – Französisch
  - `it-IT` – Italienisch
  - `ja-JP` – Japanisch

  - `ko-KR` – Koreanisch
  - `pt-BR` – Portugiesisch (Brasilien)
  - `zh-CN` – Chinesisch (vereinfacht)
  - `zh-TW` – Chinesisch (traditionell)
- `OriginationIdentity`: Die ursprüngliche Identität (z. B. ein Langcode, ein Kurzcode oder eine Absender-ID), die zum Senden des OTP-Codes verwendet wird. Wenn Sie zum Senden des OTP einen Langcode oder eine gebührenfreie Nummer verwenden, muss die Telefonnummer im E.164-Format vorliegen.
- `DestinationIdentity`: Die Telefonnummer im E.164-Format, an die der OTP-Code gesendet wurde.
- `ReferenceId`: Eine eindeutige Referenz-ID für die Anforderung. Die Referenz-ID entspricht genau der Referenz-ID, die Sie bei der Überprüfung des OTP angeben. Die Referenz-ID kann zwischen 1 und einschließlich 48 Zeichen enthalten.
- `EntityId`: Eine Entitäts-ID, die bei einer Aufsichtsbehörde registriert ist. Dieser Parameter wird derzeit nur beim Senden von Nachrichten an Empfänger in Indien verwendet. Wenn Sie nicht an Empfänger in Indien senden, können Sie diesen Parameter weglassen.
- `TemplateId`: Eine Vorlagen-ID, die bei einer Aufsichtsbehörde registriert ist. Dieser Parameter wird derzeit nur beim Senden von Nachrichten an Empfänger in Indien verwendet. Wenn Sie nicht an Empfänger in Indien senden, können Sie diesen Parameter weglassen.

> **Note**
>
> Weitere Informationen zu den Anforderungen für das Senden von Nachrichten an Empfänger in Indien finden Sie unter Besondere Anforderungen für das Senden von SMS-Nachrichten an Empfänger in Indien im Amazon-Pinpoint-Benutzerhandbuch.

Um sicherzustellen, dass Ihr Amazon Pinpoint Pinpoint-Konto ordnungsgemäß für das Senden von OTP-Nachrichten konfiguriert ist, können Sie das AWS CLI zum Senden einer Testnachricht verwenden. Weitere Informationen zu finden Sie im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Um eine OTP-Testnachricht mit dem zu senden AWS CLI, führen Sie den send-otp-messagefolgenden Befehl im Terminal aus:

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

Gehen Sie im vorhergehenden Befehl wie folgt vor:

- Ersetzen Sie *7353f53e6885409fa32d07cedexample* durch Ihre Anwendungs-ID.
- *ExampleCorp*Ersetzen Sie es durch den Namen Ihres Unternehmens.
- Ersetzen Sie *5* durch die Anzahl der Ziffern, die im OTP-Code enthalten sein werden, der an den Empfänger gesendet wird. `CodeLegth`
- Ersetzen Sie *20* durch den Zeitraum in Minuten, in dem der OTP-Code gültig sein wird. `ValidityPeriod`
- Ersetzen Sie *5* durch die Anzahl der Versuche, `AllowedAttempts` mit denen der Empfänger erfolglos versuchen kann, das Einmalpasswort zu verifizieren.
- Ersetzen Sie *+18555550142* durch die ursprüngliche Identität, die zum Senden des OTP-Codes verwendet wird. `OriginationIdentity`
- Ersetzen Sie *+12065550007* durch die Telefonnummer, an die der OTP-Code gesendet werden soll. `DestinationIdentity`
- Ersetzen Sie *SampleReferenceId*in durch eine eindeutige `ReferenceId` Referenz-ID für die Anfrage.

SendOtpMessage-Antwort

Wenn Sie eine OTP-Nachricht erfolgreich senden, erhalten Sie eine Antwort, die dem folgenden Beispiel ähnelt:

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

```
            }
        }
    }
}
```

Überprüfen einer OTP-Nachricht

Rufen Sie die `VerifyOtpMessages`-API auf, um einen OTP-Code zu verifizieren. Ihre Anforderung muss die folgenden Parameter enthalten:

- `DestinationIdentity`: Die Telefonnummer im E.164-Format, an die der OTP-Code gesendet wurde.
- `ReferenceId`: Die Referenz-ID, die Sie verwendet haben, als Sie den OTP-Code an den Empfänger gesendet haben. Die Referenz-ID muss eine exakte Übereinstimmung sein.
- `Otp`: Der OTP-Code, den Sie validieren.

Sie können das verwenden AWS CLI , um den Validierungsprozess zu testen. Weitere Informationen zur Installation und Konfiguration von finden Sie im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Um ein OTP mit dem zu überprüfen AWS CLI, führen Sie den verify-otp-messagefolgenden Befehl im Terminal aus:

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

Gehen Sie im vorhergehenden Befehl wie folgt vor:

- Ersetzen Sie *7353f53e6885409fa32d07cedexample* durch Ihre Anwendungs-ID.
- Ersetzen Sie *+12065550007 durch die Telefonnummer, an die der OTP-Code* `DestinationIdentity` gesendet wurde.
- Ersetzen Sie *SampleReferenceId*in durch eine eindeutige `ReferenceId` Referenz-ID für die Anfrage. Dieser Wert muss mit dem Wert übereinstimmen`ReferenceID`, der zum Senden der Anfrage verwendet wurde.
- Ersetzen Sie *01234* in `Otp` durch ein Otp, das an die gesendet wurde. `DestinationIdentity`

VerifyOtpMessage-Antwort

Wenn Sie eine Anforderung an die `VerifyOTPMessage`-API senden, wird ein `VerificationResponse`-Objekt zurückgegeben, das eine einzelne Eigenschaft, `Valid`, enthält. Wenn die Referenz-ID, die Telefonnummer und das Einmalpasswort alle den von Amazon Pinpoint erwarteten Werten entsprechen und das Einmalpasswort nicht abgelaufen ist, ist der Wert von `Valid` `true`, andernfalls ist er `false`. Nachfolgend finden Sie ein Beispiel für eine Antwort für eine erfolgreiche OTP-Verifizierung.

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

Codebeispiele

Dieser Abschnitt enthält Codebeispiele, die zeigen, wie das SDK für Python (Boto3) zum Senden und Überprüfen von OTP-Codes verwendet wird.

Generieren einer Referenz-ID

Die folgende Funktion generiert für jeden Empfänger eine eindeutige Referenz-ID, die auf der Telefonnummer des Empfängers, dem Produkt oder der Marke, für das der Empfänger ein OTP erhält, und der Quelle der Anforderung (dies kann beispielsweise der Name einer Seite auf einer Website oder App sein) basiert. Wenn Sie den OTP-Code verifizieren, müssen Sie eine identische Referenz-ID übergeben, damit die Validierung erfolgreich ist. Sowohl die Beispiele für den Sende- als auch für den Validierungscode verwenden diese Hilfsfunktion.

Diese Funktion ist nicht erforderlich, aber sie ist eine nützliche Methode, um den OTP-Sende- und Verifizierungsprozess auf eine bestimmte Transaktion zu beschränken, sodass diese während des Bestätigungsschritts problemlos erneut eingereicht werden kann. Sie können eine beliebige Referenz-ID verwenden – dies ist nur ein einfaches Beispiel. Die anderen Codebeispiele in diesem Abschnitt basieren jedoch auf dieser Funktion.

```
# 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()
```

Senden von OTP-Codes

Das folgende Codebeispiel zeigt Ihnen, wie Sie das SDK für Python (Boto3) verwenden, um einen OTP-Code zu senden.

```
# 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")
```

Überprüfen von OTP-Codes

Das folgende Codebeispiel zeigt Ihnen, wie Sie das SDK für Python (Boto3) verwenden, um einen OTP-Code zu verifizieren, den Sie bereits gesendet haben. Damit der Validierungsschritt erfolgreich ist, muss Ihre Anforderung eine Referenz-ID enthalten, die genau mit der Referenz-ID übereinstimmt, die zum Senden der Nachricht verwendet wurde.

```
# 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")
```

Senden und Abrufen von In-App-Nachrichten in Amazon Pinpoint

Sie können In-App-Nachrichten verwenden, um gezielt Nachrichten an Benutzer Ihrer Anwendungen zu senden. In-App-Nachrichten sind stark anpassbar. Sie können Schaltflächen enthalten, mit denen Websites geöffnet oder Benutzer zu bestimmten Teilen Ihrer Anwendung weitergeleitet werden. Sie können Hintergrund- und Textfarben konfigurieren, den Text positionieren und Benachrichtigungen Schaltflächen und Bilder hinzufügen. Sie können eine einzelne Nachricht senden oder ein Karussell erstellen, das bis zu fünf einzelne Nachrichten enthält. Eine Übersicht über In-App-Nachrichten, einschließlich Anweisungen zum Erstellen von In-App-Nachrichtenvorlagen, finden Sie unter In-App-Vorlagen erstellen im Amazon-Pinpoint-Benutzerhandbuch.

Sie können AWS Amplify für die nahtlose Integration der In-App-Nachrichtenfunktionen von Amazon Pinpoint in Ihre App nutzen. Amplify kann die Prozesse zum Abruf von Nachrichten, zur Ausgabe von Nachrichten und zum Senden von Analysedaten an Amazon Pinpoint automatisch abwickeln. Diese Integration wird derzeit für React-Native-Anwendungen unterstützt. Weitere Informationen finden Sie unter In-App Messaging (In-App-Nachrichten) in der Amplify-Framework-Dokumentation.

Dieser Abschnitt enthält Informationen zum Anfordern der In-App-Nachrichten für einen Endpunkt in Ihrer App und zur Interpretation des Ergebnisses dieser Anforderung.

Abrufen von In-App-Nachrichten für einen Endpunkt

Ihre Anwendungen können die GetInAppMessages-API aufrufen, um alle In-App-Nachrichten abzurufen, für die ein bestimmter Endpunkt berechtigt ist. Wenn Sie die `GetInAppMessages`-API aufrufen, geben Sie die folgenden Parameter an:

- `ApplicationId`: Die eindeutige ID des Amazon-Pinpoint-Projekts, dem die Kampagne zugeordnet ist.
- `EndpointId`: Die eindeutige ID des Endpunkts, für den Sie Nachrichten abrufen.

Wenn Sie die API mit diesen Werten aufrufen, gibt sie eine Liste von Nachrichten zurück. Weitere Informationen zur von dieser Operation produzierten Antwort finden Sie unter Verstehen von `GetInAppMessages`-API-Antworten.

Sie können die `GetInAppMessages`-Operation auch mit den AWS-SDKs aufrufen. Die folgenden Codebeispiele enthalten Funktionen, die In-App-Nachrichten abrufen.

JavaScript

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn:

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

In-App-Nachrichten für einen Endpunkt abrufen:

```
// 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()
```

Verstehen von **`GetInAppMessages`**-API-Antworten

Wenn Sie den API-Vorgang GetInAppMessages aufrufen, gibt er eine Liste von Nachrichten zurück, für die der angegebene Endpunkt berechtigt ist. Ihre App kann die Nachricht dann auf der Grundlage der Werte in der Antwort rendern.

Nachstehend finden Sie ein Beispiel eines JSON-Objekts, das zurückgegeben wird, wenn Sie die `GetInAppMessages`-API aufrufen:


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

In den folgenden Abschnitten erhalten Sie weitere Informationen zu den Komponenten dieser Antwort.

InAppMessageCampaigns-Objekt

Das `InAppMessageCampaigns`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`CampaignId`	Eine Zeichenfolge, die den Namen und die eindeutige Kampagnen-ID der Amazon-Pinpoint-Kampagne enthält, von der aus die Nachricht	Wird automatisch von Amazon Pinpoint erstellt, wenn Sie die Kampagne erstellen.

Attribut	Beschreibung	Anwendungsbereiche
	gesendet wurde. Der Name steht vor der Kampagnen-ID. Die beiden Werte werden durch einen Bindestrich (-) getrennt.	
`TreatmentId`	Eine Ganzzahl, die die ID der Kampagnenbehandlung für diese Nachricht darstellt. Wenn die Kampagne nur eine Behandlung hat, ist der Wert `0`.	
`Priority`	Die Priorität der In-App-Nachricht, ausgedrückt als Ganzzahl zwischen 1 und 5, wobei 1 die höchste Priorität und 5 die niedrigste ist.	Schritt 1 des Prozesses zur Kampagnenerstellung.
`InAppMessage`	Ein `InAppMessage` -Objekt, das Informationen darüber enthält, wie die Nachricht gerendert wird.	Basierend auf dem Inhalt der In-App-Nachrichten vorlage, die für die Kampagne angegeben wurde.
`Schedule`	Ein Schedule-Objekt, das Informationen darüber enthält, wann die Nachricht gesendet wurde.	Schritt 4 des Prozesses zur Kampagnenerstellung (wenn die Kampagne in der Konsole erstellt wurde) oder des `Schedule`-Objekts (wenn die Kampagne mithilfe der API oder eines SDK erstellt wurde).

Attribut	Beschreibung	Anwendungsbereiche
`DailyCap`	Die Häufigkeit, ausgedrückt als Ganzzahl, mit der eine In-App-Nachricht dem Benutzer innerhalb von 24 Stunden angezeigt werden kann.	Von Einstellungen auf Projektebene übernommen. Wenn die Kampagne Einstellungen enthält, die die Projekteinstellungen überschreiben, werden diese stattdessen verwendet.
`SessionCap`	Die Häufigkeit, ausgedrückt als Ganzzahl, dass dem Benutzer während einer Anwendungssitzung eine In-App-Nachricht angezeigt werden kann.	
`TotalCap`	Die Gesamtzahl der Male, ausgedrückt als Ganzzahl, pro Kampagne eine In-App-Nachricht einem Endpunkt angezeigt werden kann.	

`InAppMessage`-Objekt

Das `InAppMessage`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`Content`	Ein Array, das ein InAppMessageContent-Objekt enthält, das den Inhalt der Nachricht beschreibt.	Basierend auf dem Inhalt der In-App-Nachrichtenvorlage, die für die Kampagne angegeben wurde.
`Layout`	Eine Zeichenfolge, die beschreibt, wie die In-App-Nachricht auf dem Gerät des Empfängers angezeigt wird. Die möglichen Werte sind:	

Attribut	Beschreibung	Anwendungsbereiche
	• `BOTTOM_BANNER` – eine Nachricht, die als Banner unten auf der Seite angezeigt wird. • `TOP_BANNER` – eine Nachricht, die als Banner oben auf der Seite angezeigt wird. • `OVERLAYS` – eine Nachricht, die den gesamten Bildschirm abdeckt. • `MOBILE_FEED` – eine Nachricht, die in einem Fenster vor der Seite angezeigt wird. • `MIDDLE_BANNER` – eine Nachricht, die als Banner in der Mitte der Seite angezeigt wird. • `CAROUSEL` – ein scrollbares Layout mit bis zu fünf eindeutigen Nachrichten.	

`HeaderConfig`-Objekt

Das `HeaderConfig`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`Alignment`	Eine Zeichenfolge, die die Textausrichtung des Kopfzeilentexts angibt. Mögliche Werte	Basierend auf dem Inhalt der In-App-Nachrichtenvorlage, die für die Kampagne angegeben wurde.

Attribut	Beschreibung	Anwendungsbereiche
	sind `LEFT`, `CENTER` und `RIGHT`.	
`Header`	Der Text der Nachricht enkopfzeile.	
`TextColor`	Die Farbe des Kopfzeilentexts, ausgedrückt als Hex-Farbcode (z. B. #000000 für Schwarz).	

`BodyConfig`-Objekt

Das `BodyConfig`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`Alignment`	Eine Zeichenfolge, die die Textausrichtung des Nachricht entexts angibt. Mögliche Werte sind `LEFT`, `CENTER` und `RIGHT`.	Basierend auf dem Inhalt der In-App-Nachrichten vorlage, die für die Kampagne angegeben wurde.
`Body`	Der Haupttext Text der Nachricht.	
`TextColor`	Die Farbe des Textkörpe rs, ausgedrückt als eine Zeichenfolge für einen Hex-Farbcode (z. B. #000000 für Schwarz).	

`InAppMessageContent`-Objekt

Das `InAppMessageContent`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`BackgroundColor`	Die Hintergrundfarbe für ein In-App-Nachrichtenbanner, ausgedrückt als Zeichenfolge mit Hex-Farbcode (z. B. #000000 für Schwarz).	Basierend auf dem Inhalt der In-App-Nachrichtenvorlage, die für die Kampagne angegeben wurde.
`BodyConfig`	Ein BodyConfig-Objekt, das Informationen zum Hauptinhalt der Nachricht enthält.	
`HeaderConfig`	Ein HeaderConfig-Objekt, das Informationen zur Kopfzeile oder zum Titel der Nachricht enthält.	
`ImageUrl`	Die URL des Bildes, das in der Nachricht angezeigt wird.	
`PrimaryBtn`	Ein InAppMessageButton-Objekt, das Informationen über die Hauptschaltfläche in der Nachricht enthält.	
`SecondaryBtn`	Ein InAppMessageButton-Objekt, das Informationen über die sekundäre Schaltfläche in der Nachricht enthält. Nicht vorhanden, wenn die In-App-Nachrichtenvorlage keine sekundäre Schaltfläche angibt.	

`Schedule`-Objekt

Das `Schedule`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`EndDate`	Die geplante Zeit im ISO 8601-Format, zu der die Kampagne beendet wird.	Schritt 4 des Prozesses zur Kampagnenerstellung (wenn die Kampagne in der Konsole erstellt wurde) oder des `Schedule`-Objekts (wenn die Kampagne mithilfe der API oder eines SDK erstellt wurde).
`EventFilter`	Informationen über das Ereignis, das dazu führt, dass die In-App-Nachricht angezeigt wird. Wenn Sie ein Ereignis generieren, das mit einer Amazon-Pinpoint-In-App-Kampagne übereinst immt, wird die Nachricht angezeigt.	

`InAppMessageButton`-Objekt

Ein `InAppMessageButton`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`DefaultConfig`	Ein DefaultButtonConfig-Objekt, das Informationen zu den Standardeinstellungen für eine Schaltfläche in einer In-App-Nachricht enthält.	Basierend auf dem Inhalt der In-App-Nachrichten vorlage, die für die Kampagne angegeben wurde.
`Android`	Ein OverrideButtonConfig-Objekt, das angibt, wie sich die Schaltfläche auf Android-G eräten verhält. Dies überschre ibt die Standard-Schaltflä chenkonfiguration, die im `DefaultConfig` -Objekt detailliert beschrieben ist.	

Attribut	Beschreibung	Anwendungsbereiche
`IOS`	Ein OverrideButtonConfig-Objekt, das angibt, wie sich die Schaltfläche auf iOS-Gerät en verhält. Dies überschre ibt die Standard-Schaltflä chenkonfiguration, die im `DefaultConfig` -Objekt detailliert beschrieben ist.	
`Web`	Ein OverrideButtonConfig-Objekt, das angibt, wie sich die Schaltfläche in Web-Apps verhält. Dies überschreibt die Standard-Schaltflächenkonfi guration, die im `DefaultCo nfig` -Objekt detailliert beschrieben ist.	

`DefaultButtonConfig`-Objekt

Ein `DefaultButtonConfig`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`BackgroundColor`	Die Hintergrundfarbe der Schaltfläche, ausgedrückt als Zeichenfolge mit Hex-Farbcode (z. B. #000000 für Schwarz).	Basierend auf dem Inhalt der In-App-Nachrichten vorlage, die für die Kampagne angegeben wurde.
`BorderRadius`	Der Radius des Rahmens der Schaltfläche in Pixeln, ausgedrückt als Ganzzahl.	

Attribut	Beschreibung	Anwendungsbereiche
	Eine größere Zahl führt zu mehr abgerundeten Ecken.	
`ButtonAction`	Eine Zeichenfolge, die die Aktion beschreibt, die auftritt, wenn ein Empfänger eine Schaltfläche in der In-App-Nachricht auswählt. Die möglichen Werte sind: • `LINK` – Ein Link zu einem Webziel. • `DEEP_LINK` – Ein Link zu einer bestimmten Seite in einer Anwendung. • `CLOSE` – Lehnt die Nachricht ab.	
`Link`	Die Ziel-URL für eine Schaltfläche. Nicht vorhanden für Schaltflächen, bei denen die ButtonAction `CLOSE` lautet.	
`Text`	Der Text, der auf der Schaltfläche angezeigt wird.	
`TextColor`	Die Farbe des Texts auf der Schaltfläche, ausgedrückt als Zeichenfolge mit Hex-Farbcode (z. B. #000000 für Schwarz).	

`OverrideButtonConfig`-Objekt

Das `OverrideButtonConfig`-Objekt ist nur vorhanden, wenn die In-App-Nachrichtenvorlage Schaltflächen zum Überschreiben verwendet. Eine Override-Schaltfläche ist eine Schaltfläche, die eine spezifische Konfiguration für einen bestimmten Gerätetyp hat, z. B. ein iOS-Gerät, ein Android-Gerät oder einen Webbrowser.

Ein `OverrideButtonConfig`-Objekt enthält die folgenden Attribute:

Attribut	Beschreibung	Anwendungsbereiche
`ButtonAction`	Die Aktion, die auftritt, wenn ein Empfänger eine Schaltfläche in der In-App-Nachricht auswählt. Die möglichen Werte sind: • `LINK` – Ein Link zu einem Webziel. • `DEEP_LINK` – Ein Link zu einer bestimmten Seite in einer Anwendung. • `CLOSE` – Lehnt die Nachricht ab.	Basierend auf dem Inhalt der In-App-Nachrichten vorlage, die für die Kampagne angegeben wurde.
`Link`	Die Ziel-URL für eine Schaltfläche. Nicht vorhanden für Schaltflächen, bei denen die `ButtonAction` `CLOSE` lautet.	
`Text`	Der Text, der auf der Schaltfläche angezeigt wird.	
`TextColor`	Die Farbe des Texts auf der Schaltfläche, ausgedrückt als Zeichenfolge mit Hex-	

Attribut	Beschreibung	Anwendungsbereiche
	Farbcode (z. B. #000000 für Schwarz).	

Überprüfen von Telefonnummern in Amazon Pinpoint

Amazon Pinpoint enthält einen Service zur Telefonnummernüberprüfung, mit dem Sie ermitteln können, ob eine Telefonnummer gültig ist. Außerdem können Sie weitere Informationen über die Telefonnummer abrufen. Wenn Sie den Service zur Telefonnummernüberprüfung verwenden, gibt er folgende Informationen zurück:

- Telefonnummer im E.164-Format.
- Telefonnummerntyp (z. B. mobil, Festnetz oder VoIP).
- Stadt und Land , aus der die Telefonnummer stammt.
- Der für die Telefonnummer zuständige Dienstanbieter.

Für die Nutzung des Services zur Telefonnummernüberprüfung fällt eine zusätzliche Gebühr an. Weitere Informationen finden Sie unter Amazon-Pinpoint – Preise.

> **Important**
>
> Bei Telefonnummern mit Ursprung in den Vereinigten Staaten und Kanada gibt die API zur Überprüfung von Telefonnummern keine Daten mehr für `City`, `County`, `Timezone` und `ZipCode` zurück.

Anwendungsfälle der Telefonnummernüberprüfung

Sie können den Service zur Telefonnummernüberprüfung für unterschiedliche Anwendungsfälle nutzen, z. B.:

- Überprüfung der in einem Webformular angegebenen Telefonnummern: Wenn Sie webbasierte Formulare verwenden, um Kontaktinformationen für Ihre Kunden zu sammeln, überprüfen Sie die von den Kunden angegebenen Telefonnummern, bevor Sie das Formular absenden. Verwenden Sie das Backend der Website, um die Telefonnummern mithilfe der Amazon Pinpoint-API zu überprüfen. In der API-Antwort wird angegeben, ob die Nummer ungültig ist, z. B. wenn die Telefonnummer falsch formatiert ist. Wenn Sie ermitteln, dass die vom Kunden angegebene Telefonnummer ungültig ist, kann das Web-Formular den Kunden auffordern, die Telefonnummer zu korrigieren.

- Bereinigen Ihrer vorhandenen Kontaktdatenbank: Wenn Sie über eine Datenbank mit Kunden-Telefonnummern verfügen, können Sie jede Telefonnummer überprüfen und dann Ihre Datenbank auf der Grundlage Ihrer Ergebnisse aktualisieren. Wenn Sie beispielsweise Endpunkte mit Telefonnummern finden, die keine SMS-Nachrichten empfangen können, können Sie die Eigenschaft `ChannelType` des Endpunkts von `SMS` in `VOICE` ändern. Sie können zuerst die Telefonnummer überprüfen und dann die `ChannelType`-Eigenschaft für neue oder bestehende Endpunkte aktualisieren, indem Sie den Anweisungen in Hinzufügen von Endpunkten zu Amazon Pinpoint für einen einzelnen Endpunkt oder in Hinzufügen eines Stapels Endpunkte zu Amazon Pinpoint für mehrere Endpunkte folgen.
- Auswählen des richtigen Kanals, bevor Sie eine Nachricht senden: Wenn Sie eine SMS-Nachricht senden möchten, aber feststellen, dass die Zieltelefonnummer ungültig ist, können Sie die Nachricht über einen anderen Channel an den Empfänger senden. Wenn der Endpunkt beispielsweise keine SMS-Nachrichten empfangen kann, können Sie stattdessen eine Sprachnachricht senden.

Verwenden des Services zur Telefonnummernüberprüfung

Das folgende Beispiel zeigt, wie eine Telefonnummer mit dem validiert wird AWS CLI. Weitere Informationen finden Sie phone-number-validatein der AWS CLI Befehlsreferenz. Antworten auf die Validierung finden Sie beispielsweise unterAntworten der Telefonnummernüberprüfung. Weitere Informationen zur Konfiguration von finden Sie unter Configure the AWS CLI im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Um den Dienst zur Überprüfung von Telefonnummern zu nutzen, verwenden Sie den AWS CLI

- Geben Sie in der Befehlszeile folgenden Befehl ein:

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

  Ersetzen Sie im vorherigen Befehl *+442079460881* durch die Telefonnummer, die Sie überprüfen möchten, und *GB durch den zweistelligen ISO-Landes* - oder Regionalcode.

> **Note**
>
> Wenn Sie eine Telefonnummer an den Service zur Telefonnummernüberprüfung übergeben, sollten Sie immer den Ländercode einschließen. Wenn Sie den Ländercode nicht einschließen, gibt der Service möglicherweise Informationen zu einer Telefonnummer in einem anderen Land zurück. *Die Telefonnummer kann Bindestriche enthalten, z. B. +44-207-946-0881.*

Antworten der Telefonnummernüberprüfung

Die Informationen, die der Service zur Telefonnummernüberprüfung zurückgibt, variieren geringfügig in Abhängigkeit von den Daten, die zu der von Ihnen angegebenen Telefonnummer verfügbar sind. Dieser Abschnitt enthält Beispiele für die Antworten, die der Service zur Telefonnummernüberprüfung zurückgibt.

> **Note**
>
> Die Daten, die der Service zur Telefonnummernüberprüfung bereitstellt, basieren auf den Daten, die von Telekommunikationsanbietern und anderen Organisationen weltweit übermittelt werden. Anbieter aktualisieren diese Daten in einigen Ländern möglicherweise seltener als Anbieter in anderen Ländern. Wenn Sie beispielsweise eine Anforderung zum Validieren einer Mobiltelefonnummer senden und die angegebene Telefonnummer von einer Telefongesellschaft zu einer anderen portiert wurde, enthält die Antwort des Services zur Telefonnummernüberprüfung möglicherweise den Namen der ursprünglichen Telefongesellschaft und nicht den des jetzt zuständigen Unternehmens.

Gültige Mobiltelefonnummern

Wenn Sie eine Anforderung an den Service zur Telefonnummernüberprüfung senden und es sich bei der Telefonnummer um eine gültige Mobiltelefonnummer handelt, werden Daten wie im folgenden Beispiel zurückgegeben:

```
{
    "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"
    }
}
```

Gültige Festnetztelefonnummern

Wenn die Anforderung eine gültige Festnetztelefonnummer enthält, gibt der Service zur Telefonnummernüberprüfung Daten wie im folgenden Beispiel zurück:

```
{
    "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"
}
```

Gültige VoIP-Telefonnummern

Wenn die Anforderung eine gültige VoIP (Voice-over-Internet-Protocol)-Telefonnummer enthält, gibt der Service zur Telefonnummernüberprüfung Daten wie im folgenden Beispiel zurück:

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
```

```
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+441514960001",
        "CleansedPhoneNumberNational": "1514960001",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+441514960001",
        "PhoneType": "VOIP",
        "PhoneTypeCode": 2
    }
}
```

Ungültige Telefonnummern

Wenn die Anforderung eine ungültige Telefonnummer enthält, gibt der Service zur Telefonnummernüberprüfung Daten wie im folgenden Beispiel zurück:

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

Beachten Sie, dass die Eigenschaft `PhoneType` in dieser Antwort angibt, dass diese Telefonnummer `INVALID` ist und dass keine Informationen zur Telefongesellschaft oder zum Ort der Telefonnummer enthalten sind. Senden Sie keine SMS- oder Sprachnachrichten an Telefonnummern, deren `PhoneType` den Wert `INVALID` hat, weil sie wahrscheinlich keinen tatsächlichen Empfängern gehören.

Andere Telefonnummern

Gelegentlich enthält die Antwort des Services zur Telefonnummernüberprüfung einen `PhoneType`-Wert von `OTHER`. Der Service gibt Antworten dieser Art möglicherweise in den folgenden Situationen zurück:

- Die Telefonnummer ist eine gebührenfreie Nummer.
- Die Telefonnummer ist für Fernsehsendungen und Filme reserviert. In Nordamerika sind dies z. B. Telefonnummern, die mit 555 beginnen.
- Die Telefonnummer enthält eine Ortsnetzkennzahl, die derzeit nicht genutzt wird (z. B. 999 in Nordamerika).
- Die Telefonnummer ist für andere Zwecke reserviert.

Das folgende Beispiel zeigt die Antwort, die der Service zur Telefonnummernüberprüfung zurückgibt, wenn die Anforderung eine fiktive nordamerikanische Telefonnummer enthält:

```
{
    "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"
    }
}
```

Prepaid-Telefonnummern

Wenn die Anforderung eine gültige Prepaid-Telefonnummer enthält, gibt der Service zur Telefonnummernüberprüfung Daten wie im folgenden Beispiel zurück:

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

```
        "PhoneType": "PREPAID",
        "PhoneTypeCode": 5
    }
}
```

Weitere Informationen über die in diesen Antworten enthaltenen Daten finden Sie unter Telefonnummernüberprüfung in der Amazon-Pinpoint-API-Referenz.

Senden von Transaktionsnachrichten von Ihren Apps

Sie können die Amazon-Pinpoint-API und die AWS-SDKs zum Senden von Transaktionsnachrichten direkt aus Ihren Apps verwenden. Transaktionsnachrichten sind Nachrichten, die Sie an bestimmte Empfänger senden – im Gegensatz zu Nachrichten, die Sie an Segmente senden. Es gibt mehrere Gründe, warum Sie möglicherweise Transaktionsnachrichten senden möchten anstatt kampagnenbasierte Nachrichten. Sie könnten z. B. eine Auftragsbestätigung per E-Mail senden, wenn ein Kunde eine Bestellung aufgibt. Sie könnten per SMS- oder Sprachnachricht auch ein einmaliges Passwort senden, mit dem ein Kunde die Erstellung seines Kontos für Ihren Service abschließen kann.

Dieser Abschnitt enthält Codebeispiele in mehreren Programmiersprachen, die Sie verwenden können, um Transaktionsnachrichten, SMS-Nachrichten und Sprachnachrichten zu versenden.

Themen in diesem Abschnitt:

Versenden von E-Mail-Transaktionsnachrichten

Dieser Abschnitt enthält umfassende Codebeispiele, die Sie verwenden können, um E-Mail-Transaktionsnachrichten über Amazon Pinpoint zu versenden.

- Mithilfe des SendMessages Vorgangs in der Amazon Pinpoint-API: Sie können den `SendMessages` Vorgang in der Amazon Pinpoint Pinpoint-API verwenden, um Nachrichten in allen Kanälen zu senden, die Amazon Pinpoint unterstützt, einschließlich Push-Benachrichtigungen, SMS, Sprach- und E-Mail-Kanälen.

  Der Vorteil dieser Operation besteht darin, dass die Anforderungssyntax zum Senden von Nachrichten in allen Kanälen sehr ähnlich ist. So ist es einfacher, vorhandenen Code wiederzuverwenden. Mit der Operation `SendMessages` können Sie auch Inhalte in Ihren E-Mail-Nachrichten ersetzen und E-Mails an die Amazon-Pinpoint-Endpunkt-IDs senden statt an bestimmte E-Mail-Adressen.

Dieser Abschnitt enthält Codebeispiele in mehreren Programmiersprachen, die Sie verwenden können, um E-Mail-Transaktionsnachrichten zu versenden.

Themen in diesem Abschnitt:

Auswählen einer Methode für den E-Mail-Versand

Die beste Methode für den Versand von E-Mail-Transaktionsnachrichten hängt vom jeweiligen Anwendungsszenario ab. Wenn Sie beispielsweise E-Mails mithilfe einer Drittanbieteranwendung senden müssen oder wenn kein AWS SDK für Ihre Programmiersprache verfügbar ist, müssen Sie möglicherweise die SMTP-Schnittstelle verwenden. Wenn Sie Nachrichten in anderen Kanälen senden möchten, die von Amazon Pinpoint unterstützt werden, und für diese Anforderungen einen konsistenten Code anwenden wollen, sollten Sie die Operation `SendMessages` in der Amazon-Pinpoint-API verwenden.

Auswahl zwischen Amazon Pinpoint und Amazon Simple Email Service (SES)

Wenn Sie eine große Anzahl von Transaktions-E-Mails senden, z. B. Kaufbestätigungen oder Nachrichten zum Zurücksetzen des Passworts, sollten Sie Amazon SES verwenden. Amazon SES verfügt über eine API und eine SMTP-Schnittstelle, die beide gut geeignet sind, um E-Mails von Ihren Anwendungen oder Services zu senden. Es bietet auch zusätzliche E-Mail-Funktionen, einschließlich E-Mail-Empfangsfunktionen, Konfigurationssets und Sendeautorisierungsfunktionen.

Amazon SES enthält auch eine SMTP-Schnittstelle, die Sie in Ihre vorhandenen Drittanbieteranwendungen integrieren können, einschließlich CRM-Services (Customer Relationship Management) wie Salesforce. Weitere Informationen zum Senden von E-Mails mit Amazon SES finden Sie im Amazon Simple Email Service Entwicklerhandbuch.

Senden von E-Mails mithilfe der Amazon-Pinpoint-API

Dieser Abschnitt enthält vollständige Codebeispiele, die Sie verwenden können, um E-Mails mithilfe eines AWS SDK über die Amazon Pinpoint Pinpoint-API zu versenden.

C#

Verwenden Sie dieses Beispiel, um eine E-Mail mithilfe des AWS SDK for .NET zu versenden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das AWS SDK for .NET bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Erste Schritte mit AWS SDK for .NET im AWS SDK for .NET -Entwicklerhandbuch.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen Benutzer anzugeben. Weitere Informationen finden Sie unter Konfigurieren von AWS -Anmeldeinformationen) im AWS SDK for .NET -Entwicklerhandbuch.

Dieses Codebeispiel wurde mit der AWS SDK for .NET Version 3.3.29.13 und.NET Core Runtime Version 2.1.2 getestet.

```
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

Verwenden Sie dieses Beispiel, um eine E-Mail mithilfe des AWS SDK for Java zu versenden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das AWS SDK for Java 2.x bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Erste Schritte im AWS SDK for Java 2.x -Entwicklerhandbuch.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen Benutzer anzugeben. Weitere Informationen finden Sie unter Einrichten der Standard-Anmeldeinformationen und -Region im AWS SDK for Java -Entwicklerhandbuch.

Dieses Codebeispiel wurde mit der AWS SDK for Java Version 2.3.1 und der OpenJDK-Version 11.0.1 getestet.

```
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);
        }
    }
```

```
}
```

Das vollständige SDK-Beispiel finden Sie unter .java on. SendEmailMessage GitHub

JavaScript (Node.js)

Verwenden Sie dieses Beispiel, um E-Mails mithilfe des AWS SDK für JavaScript in Node.js zu senden. In diesem Beispiel wird vorausgesetzt, dass Sie das SDK für JavaScript in Node.js bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Erste Schritte im AWS SDK für JavaScript im Node.js Developer Guide.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen Benutzer anzugeben. Weitere Informationen finden Sie unter Einrichten von Anmeldeinformationen im AWS SDK für JavaScript im Node.js Developer Guide.

Dieses Codebeispiel wurde mit dem SDK für JavaScript in Node.js Version 2.388.0 und Node.js Version 11.7.0 getestet.

```
"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

Verwenden Sie dieses Beispiel, um eine E-Mail mithilfe des AWS SDK for Python (Boto3) zu versenden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das SDK für Python (Boto3) bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Quickstart in der API-Referenz zum AWS -SDK für 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()
```

Sie können Nachrichtenvorlagen auch zum Senden von E-Mail-Nachrichten verwenden, wie im folgenden Beispiel gezeigt:

```
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()
```

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen Benutzer anzugeben. Weitere Informationen finden Sie unter Anmeldeinformation in der API-Referenz zum AWS -SDK für Python (Boto3).

Senden einer E-Mail mit Headern zum Abbestellen

> **Note**
>
> Bevor Sie E-Mail-Header verwenden können, müssen Sie eine E-Mail-Orchestration-Senderolle einrichten, wenn Sie E-Mails aus einer Kampagne oder einer Journey versenden. Für das direkte Senden von E-Mails benötigen Sie Berechtigungen für `ses:SendEmail` und. `ses:SendRawEmail` Weitere Informationen finden Sie unter Erstellen einer E-Mail-Orchestration-Senderolle im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

Es hat sich bewährt, einen Link zum Abbestellen in Ihre E-Mail aufzunehmen. In einigen Ländern ist dies sogar gesetzlich vorgeschrieben. Um einen Link zum Abbestellen mit einem Klick hinzuzufügen, fügen Sie die folgenden Überschriften hinzu:

1. Setzen Sie den Header-Namen auf `List-Unsubscribe` und legen Sie Wert auf Ihren Abmeldelink fest. Der Link muss HTTP-POST-Anfragen unterstützen, um die Abmeldeanfrage des Empfängers bearbeiten zu können.
2. Setzen Sie den Header-Namen auf `List-Unsubscribe-Post` und legen Sie den Wert auf fest`List-Unsubscribe=One-Click`.

Sie können einer E-Mail-Nachricht bis zu 15 Kopfzeilen hinzufügen. Eine Liste der unterstützten Header finden Sie unter Amazon SES SES-Header-Felder im Amazon Simple Email Service Developer Guide.

Das folgende Beispiel zeigt, wie Sie eine E-Mail-Nachricht mit Abmelde-Headern mithilfe von versenden. AWS Command Line Interface Weitere Informationen zur Konfiguration von finden Sie unter Configure the AWS CLI im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Gehen Sie im folgenden Befehl wie folgt vor:

- *AppId*Ersetzen Sie es durch Ihre Anwendungs-ID.
- Ersetzen Sie *richard_roe@example.com* durch die E-Mail-Adresse des Empfängers.
- Ersetzen Sie *https://example.com/unsub* durch Ihren Abmeldelink.
- Ersetzen Sie *example123456* durch eine eindeutige Kennung für den Empfänger.

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

Senden von SMS-Nachrichten

Sie können die Amazon-Pinpoint-API zum Senden von SMS-Nachrichten (Textnachrichten) an bestimmte Telefonnummern oder Endpunkt-IDs verwenden. Dieser Abschnitt enthält vollständige Codebeispiele, die Sie verwenden können, um SMS-Nachrichten mithilfe eines AWS SDK über die Amazon Pinpoint Pinpoint-API zu versenden.

C#

Verwenden Sie dieses Beispiel, um eine SMS-Nachricht mithilfe des AWS SDK for .NET zu versenden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das AWS SDK for .NET bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Erste Schritte im AWS SDK for .NET -Entwicklerhandbuch.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen IAM-Benutzer anzugeben. Weitere Informationen finden Sie unter Konfigurieren von AWS -Anmeldeinformationen) im AWS SDK for .NET -Entwicklerhandbuch.

```
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

Verwenden Sie dieses Beispiel, um eine SMS-Nachricht mithilfe des AWS SDK for Java zu versenden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das SDK für Java bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Erste Schritte im AWS SDK for Java -Entwicklerhandbuch.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen IAM-Benutzer anzugeben. Weitere Informationen finden Sie unter Einrichten der Standard-Anmeldeinformationen und -Region im AWS SDK for Java -Entwicklerhandbuch.

```
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);
                }
        }
}
```

Das vollständige SDK-Beispiel finden Sie unter SendMessage.java on. GitHub

JavaScript (Node.js)

Verwenden Sie dieses Beispiel, um mithilfe des AWS SDK für JavaScript in Node.js eine SMS-Nachricht zu senden. In diesem Beispiel wird vorausgesetzt, dass Sie das SDK für JavaScript in Node.js bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Erste Schritte im AWS SDK für JavaScript im Node.js Developer Guide.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen IAM-Benutzer anzugeben. Weitere Informationen finden Sie unter Einrichten von Anmeldeinformationen im AWS SDK für JavaScript im 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

Verwenden Sie dieses Beispiel, um eine SMS-Nachricht mithilfe des AWS SDK for Python (Boto3) zu versenden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das SDK für Python bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Schnellstart 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()
```

Sie können Nachrichtenvorlagen auch zum Senden von SMS-Nachrichten verwenden, wie im folgenden Beispiel gezeigt:

```
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()
```

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen IAM-Benutzer anzugeben. Weitere Informationen finden Sie unter Anmeldeinformation in der API-Referenz zum AWS -SDK für Python (Boto3).

Senden von Sprachnachrichten

Sie können die Amazon-Pinpoint-API verwenden, um Sprachnachrichten an bestimmte Telefonnummern zu senden. Dieser Abschnitt enthält umfassende Codebeispiele, die Sie verwenden können, um Sprachnachrichten über die Amazon-Pinpoint-SMS- und Sprachnachrichten-API mithilfe eines AWS-SDK zu senden.

Java

Verwenden Sie dieses Beispiel, um eine Sprachnachricht mithilfe des AWS SDK for Java zu versenden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das SDK für Java bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Erste Schritte im AWS SDK for Java-Entwicklerhandbuch.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel

für einen vorhandenen Benutzer anzugeben. Weitere Informationen finden Sie unter Einrichten der AWS-Anmeldeinformationen und -Region für die Entwicklung im AWS SDK for Java-Entwicklerhandbuch.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

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

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

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

        // The content of the message. This example uses SSML to customize and
control
        // certain aspects of the message, such as by adding pauses and changing
        // phonation. The message can't contain any line breaks.
        static final String ssmlMessage = "<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);
                }
        }
}
```

Das vollständige SDK-Beispiel finden Sie unter SendVoiceMessage.java auf GitHub.

JavaScript (Node.js)

Verwenden Sie dieses Beispiel, um eine Sprachnachricht mithilfe der AWS-SDK für JavaScript in Node.js zu senden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das SDK für JavaScript in Node.js bereits installiert und konfiguriert haben.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen Benutzer anzugeben. Weitere Informationen finden Sie unter Festlegen von Anmeldeinformationen im Entwicklerhandbuch für AWS-SDK für JavaScript in Node.js.

```
"use strict";

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

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

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best results,
 you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
```

```
var languageCode = "en-US";

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

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

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

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

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

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
```

```
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

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

Python

Verwenden Sie dieses Beispiel, um eine Sprachnachricht mithilfe des AWS SDK for Python (Boto3) zu versenden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das SDK für Python (Boto3) bereits installiert und konfiguriert haben.

In diesem Beispiel wird davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen Benutzer anzugeben. Weitere Informationen finden Sie unter Anmeldeinformation in der API-Referenz zum AWS-SDK für 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()
```

Senden von Push-Benachrichtigungen

Die Amazon-Pinpoint-API kann transaktionale Push-Benachrichtigungen an bestimmte Geräte-IDs senden. Dieser Abschnitt enthält umfassende Codebeispiele, die Sie verwenden können, um Push-Benachrichtigungen über die Amazon-Pinpoint-API mithilfe eines AWS-SDKs zu senden.

Sie können diese Beispiele verwenden, um Push-Benachrichtigungen über jeden Push-Benachrichtigungsservice zu senden, den Amazon Pinpoint unterstützt. Derzeit unterstützt Amazon

Pinpoint die folgenden Kanäle: Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNs), Baidu Cloud Push und Amazon Device Messaging (ADM).

> **Note**
>
> Wenn Sie Push-Benachrichtigungen über den Firebase Cloud Messaging (FCM)-Service senden, verwenden Sie den Servicenamen `GCM` in Ihrem Aufruf an die Amazon-Pinpoint-API. Der Google Cloud Messaging (GCM)-Service wurde von Google am 10. April 2018 eingestellt. Die Amazon-Pinpoint-API verwendet jedoch den `GCM`-Servicenamen für Nachrichten, die sie über den FCM-Service sendet, um die Kompatibilität mit API-Code aufrechtzuerhalten, der vor der Einstellung des GCM-Service geschrieben wurde.

JavaScript (Node.js)

Verwenden Sie dieses Beispiel, um Push-Benachrichtigungen mithilfe von AWS-SDK für JavaScript in Node.js. zu senden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das SDK für JavaScript in Node.js bereits installiert und konfiguriert haben.

In diesem Beispiel wird auch davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen -Benutzer anzugeben. Weitere Informationen finden Sie unter Festlegen von Anmeldeinformationen im Entwicklerhandbuch für AWS-SDK für JavaScript in Node.js.

```
'use strict';

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

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

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

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

// The Amazon Pinpoint project ID that you want to use when you send this
```

```
// message. Make sure that the push channel is enabled for the project that
// you choose.
var applicationId = 'ce796be37f32f178af652b26eexample';

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

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

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

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

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

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

function CreateMessageRequest() {
  var token = recipient['token'];
  var service = recipient['service'];
  if (service == 'GCM') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'GCM'
```

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

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

  return messageRequest
}

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

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

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

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

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

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

SendMessage()
```

Python

Verwenden Sie dieses Beispiel, um Push-Benachrichtigungen mithilfe von AWS SDK for Python (Boto3) zu senden. Bei diesem Beispiel wird vorausgesetzt, dass Sie das SDK für Python (Boto3) bereits installiert und konfiguriert haben.

In diesem Beispiel wird auch davon ausgegangen, dass Sie eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, um den Zugriffsschlüssel und den geheimen Zugriffsschlüssel für einen vorhandenen -Benutzer anzugeben. Weitere Informationen finden Sie unter Anmeldeinformation in der API-Referenz zum AWS-SDK für Python (Boto3).

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

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

```
region = "us-east-1"

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

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

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

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

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

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

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

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

# Boolean that specifies whether the notification is sent as a silent
```

```
# notification (a notification that doesn't display on the recipient's device).
silent = False

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

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

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

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

    return message_request

# Show a success or failure message, and provide the response from the API.
```

```
def show_output(response):
    if response['MessageResponse']['Result'][recipient["token"]]['DeliveryStatus']
 == "SUCCESSFUL":
        status = "Message sent! Response information:\n"
    else:
        status = "The message wasn't sent. Response information:\n"
    print(status, json.dumps(response,indent=4))

# Send the message through the appropriate channel.
def send_message():

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

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

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

send_message()
```

Erstellen von benutzerdefinierten Kanälen in Amazon Pinpoint

Amazon Pinpoint bietet integrierte Unterstützung für das Senden von Nachrichten über die Kanäle Push-Benachrichtigung, E-Mail, SMS und Sprache. Sie können Amazon Pinpoint auch so konfigurieren, dass Nachrichten über andere Kanäle gesendet werden, indem Sie benutzerdefinierte Kanäle erstellen. Benutzerdefinierte Kanäle in Amazon Pinpoint ermöglichen es Ihnen, Nachrichten über jeden Dienst zu senden, der über eine API verfügt, dazu gehören auch Dienste von Drittanbietern. Sie können mit APIs interagieren, indem Sie einen Webhook verwenden oder eine AWS Lambda-Funktion aufrufen.

Die Segmente, an die Sie Kampagnen über benutzerdefinierte Kanäle senden, können Endpunkte aller Typen enthalten (d. h. Endpunkte, bei denen der Wert des `ChannelType`-Attributs EMAIL (E-MAIL), VOICE (SPRACHE), SMS, CUSTOM (BENUTZERDEFINIERT) oder einer der verschiedenen Endpunkttypen für Push-Benachrichtigung ist).

Erstellen einer Kampagne, die Nachrichten über einen benutzerdefinierten Kanal sendet

Um einer einzelnen Kampagne eine Lambda-Funktion oder einen Webhook zuzuweisen, verwenden Sie die Amazon-Pinpoint-API, um ein Kampagnenobjekt zu erstellen oder zu aktualisieren.

Das `MessageConfiguration`-Objekt in der Kampagne muss ebenfalls ein `CustomMessage`-Objekt enthalten. Dieses Objekt verfügt über ein Mitglied, `Data`. Der Wert von `Data` ist eine JSON-Zeichenfolge, die die Nachrichtennutzlast enthält, die Sie an den benutzerdefinierten Kanal senden möchten.

Die Kampagne muss ein `CustomDeliveryConfiguration`-Objekt enthalten. Geben Sie innerhalb des `CustomDeliveryConfiguration`-Objekts Folgendes an:

- `EndpointTypes`: Ein Array, das alle Endpunkttypen enthält, an die die Kampagne über benutzerdefinierte Kanäle gesendet werden soll. Es kann einen oder alle der folgenden Kanaltypen enthalten:
  - `ADM`
  - `APNS`
  - `APNS_SANDBOX`

- `APNS_VOIP`
- `APNS_VOIP_SANDBOX`
- `BAIDU`
- `CUSTOM`
- `EMAIL`
- `GCM`
- `SMS`
- `VOICE`

- `DeliveryUri`: Das Ziel, an das die Endpunkte gesendet werden. Sie können von Folgendem nur eines angeben:
  - den Amazon-Ressourcenname (ARN) einer Lambda-Funktion, die Sie ausführen möchten, wenn die Kampagne ausgeführt wird.
  - die URL des Webhooks, an den Sie Endpunktdaten senden möchten, wenn die Kampagne ausgeführt wird.

> **Note**
>
> Das `Campaign`-Objekt kann auch ein `Hook`-Objekt enthalten. Dieses Objekt wird nur zum Erstellen von Segmenten verwendet, die beim Ausführen einer Kampagne durch eine Lambda-Funktion angepasst werden. Weitere Informationen finden Sie unter Anpassung von Segmenten mit AWS Lambda.

Grundlegendes zu den Ereignisdaten, die über Amazon Pinpoint an benutzerdefinierte Kanäle gesendet werden

Bevor Sie eine Lambda-Funktion erstellen, die Nachrichten über einen benutzerdefinierten Kanal sendet, sollten Sie sich mit den Daten, die Amazon Pinpoint ausgibt, vertraut machen. Wenn eine Amazon-Pinpoint-Kampagne Nachrichten über einen benutzerdefinierten Kanal sendet, sendet sie eine Nutzlast an die Lambda-Zielfunktion, die dem folgenden Beispiel ähnelt:

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

Die Ereignisdaten stellen die folgenden Attribute bereit:

- `ApplicationId`: Die ID des Amazon-Pinpoint-Projekts, zu dem die Kampagne gehört.
- `CampaignId`: Die ID der Amazon-Pinpoint-Kampagne, die die Lambda-Funktion aufgerufen hat.
- `TreatmentId`: Die ID der Kampagnenvariante. Wenn Sie eine Standard-Kampagne erstellt haben, ist dieser Wert immer 0. Wenn Sie eine A/B-Testkampagne erstellt haben, ist dieser Wert eine ganze Zahl zwischen 0 und 4.
- `ActivityId`: Die ID der Aktivität, die von der Kampagne ausgeführt wird.

- `ScheduledTime`: Der Zeitpunkt, zu dem Amazon Pinpoint die Kampagne ausgeführt hat, wird im ISO 8601-Format angezeigt.
- `Endpoints`: Eine Liste der Endpunkte, die das Ziel der Kampagne waren. Jede Nutzlast kann bis zu 50 Endpunkte enthalten. Wenn das Segment, an das die Kampagne gesendet wurde, mehr als 50 Endpunkte enthält, ruft Amazon Pinpoint die Funktion wiederholt auf, mit bis zu 50 Endpunkten gleichzeitig, bis alle Endpunkte verarbeitet wurden.

Sie können diese Beispieldaten beim Erstellen und Testen Ihrer benutzerdefinierten Lambda-Kanalfunktion verwenden.

Konfigurieren von Webhooks

Wenn Sie einen Webhook verwenden, um Nachrichten über benutzerdefinierte Kanäle zu senden, muss die URL des Webhooks mit „https://“ beginnen. Die Webhook-URL darf nur alphanumerische Zeichen sowie die folgenden Symbole enthalten: Bindestrich (-), Punkt (.), Unterstrich (_), Tilde (~), Fragezeichen (?), Schrägstrich (/), Rautezeichen (#) und Semikolon (;). Die URL muss RFC3986 entsprechen.

Wenn Sie eine Kampagne erstellen, die eine Webhook-URL angibt, gibt Amazon Pinpoint einen HTTP `HEAD` an diese URL aus. Die Antwort auf die `HEAD`-Anforderung muss einen Header namens `X-Amz-Pinpoint-AccountId` enthalten. Der Wert dieses Headers muss Ihrer AWS-Konto-ID entsprechen.

Konfigurieren von Lambda-Funktionen

Dieser Abschnitt bietet einen Überblick über die Schritte, die Sie ausführen müssen, wenn Sie eine Lambda-Funktion erstellen, die Nachrichten über einen benutzerdefinierten Kanal sendet. Zuerst erstellen Sie die Funktion. Danach fügen Sie der Funktion eine Ausführungsrichtlinie hinzu. Diese Richtlinie ermöglicht es Amazon Pinpoint, die Richtlinie auszuführen, wenn eine Kampagne ausgeführt wird.

Eine Einführung in die Erstellung von Lambda-Funktionen finden Sie unter Erstellen von Lambda-Funktionen im AWS LambdaEntwicklerhandbuch.

Beispiel-Lambda-Funktion

Im folgenden Codebeispiel wird die Nutzlast verarbeitet und die Anzahl der Endpunkte jedes Endpunkttyps in CloudWatch protokolliert.

```
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)
```

Wenn eine Amazon-Pinpoint-Kampagne diese Lambda-Funktion ausführt, sendet Amazon Pinpoint der Funktion eine Liste der Segmentmitglieder. Die Funktion zählt die Anzahl der Endpunkte jedes `ChannelType`. Anschließend sendet es diese Daten an Amazon CloudWatch. Diese Metriken

können Sie auch im Abschnitt Metriken in der CloudWatch-Konsole anzeigen. Die Metriken sind im Namespace PinpointCustomChannelExecution verfügbar.

Sie können dieses Codebeispiel so ändern, dass es auch eine Verbindung mit der API eines externen Dienstes herstellt, um Nachrichten über diesen Dienst zu senden.

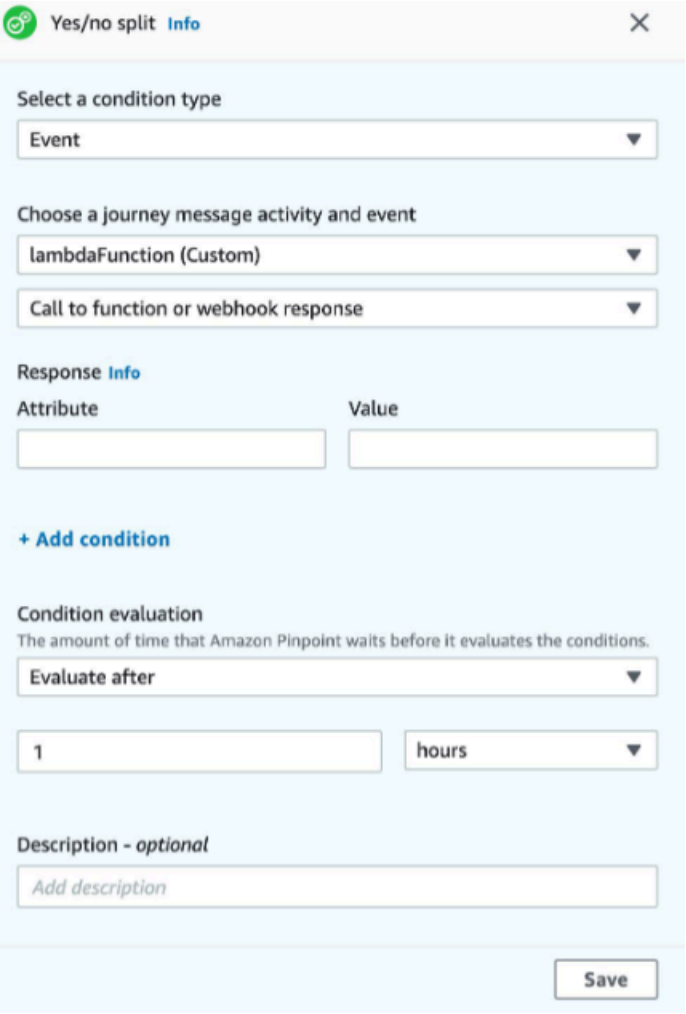
Antwortformat der Lambda-Funktion für Amazon Pinpoint

Wenn Sie die Journey Mehrfach- oder Ja/Nein-Split verwenden möchten, um den Endpunktpfad nach einer benutzerdefinierten Kanalaktivität zu bestimmen, müssen Sie Ihre Lambda-Funktionsantwort in einem Format strukturieren, das Amazon Pinpoint verstehen kann, und dann Endpunkte auf den richtigen Pfad senden.

Der Antwortstruktur sollte das folgende Format aufweisen:

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

Auf diese Weise können Sie dann einen Schlüssel und einen Wert auswählen, mit dem Sie den Pfad der Endpunkte bestimmen möchten.



Gewähren der Berechtigung zum Aufrufen der Lambda-Funktion an Amazon Pinpoint

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Berechtigungen für die Lambda-Funktionsrichtlinie hinzuzufügen, die Ihrer Lambda-Funktion zugewiesen ist. Um Amazon Pinpoint den Aufruf einer Funktion zu ermöglichen, verwenden Sie den Befehl add-permission, wie im folgenden Beispiel gezeigt:

```
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
```

Gehen Sie im vorhergehenden Befehl wie folgt vor:

- Ersetzen Sie *myFunction* durch den Namen der Lambda-Funktion.
- Ersetzen Sie *us-east-1* durch die AWS-Region, in der Sie Amazon Pinpoint verwenden.
- Ersetzen Sie *111122223333* durch Ihre AWS-Konto-ID.

Wenn Sie den Befehl `add-permission` ausführen, gibt Lambda die folgende Ausgabe zurück:

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

Der `Statement`-Wert ist eine JSON-Zeichenfolgenversion der Anweisung, die der Lambda-Funktionsrichtlinie hinzugefügt wurde.

Weiteres Einschränken der Ausführungsrichtlinie

Sie können die Ausführungsrichtlinie ändern, indem Sie sie auf ein bestimmtes Amazon-Pinpoint-Projekt beschränken. Ersetzen Sie dazu das `*` im vorangegangenen Beispiel durch die eindeutige ID des Projekts. Sie können die Richtlinie weiter einschränken, indem Sie sie auf eine bestimmte Kampagne beschränken. Um beispielsweise die Richtlinie so einzuschränken, dass nur eine Kampagne mit der Kampagnen-ID `95fee4cd1d7f5cd67987c1436example` in einem Projekt mit der Projekt-ID `dbaf6ec2226f0a9a8615e3ea5example` zugelassen wird, verwenden Sie den folgenden Wert für das `source-arn`-Attribut:

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

> **Note**
>
> Wenn Sie die Ausführung der Lambda-Funktion auf eine bestimmte Kampagne beschränken, müssen Sie die Funktion zunächst mit einer weniger restriktiven Richtlinie erstellen. Als Nächstes müssen Sie die Kampagne in Amazon Pinpoint erstellen und die Funktion auswählen. Abschließend müssen Sie die Ausführungsrichtlinie aktualisieren, damit diese auf die angegebene Kampagne referenziert wird.

Streamen von Amazon-Pinpoint-Ereignissen zu Kinesis

In Amazon Pinpoint ist ein Ereignis eine Aktion, die auftritt, wenn ein Benutzer mit einer Ihrer Anwendungen interagiert, Sie eine Nachricht aus einer Kampagne oder einer Journey senden oder eine Transaktions-SMS oder eine E-Mail-Nachricht senden. Wenn Sie beispielsweise eine E-Mail-Nachricht senden, treten mehrere Ereignisse auf:

- Wenn Sie die Nachricht senden, tritt ein Sendeereignis ein.
- Wenn die Nachricht den Posteingang des Empfängers erreicht, tritt ein Zustellereignis ein.
- Wenn der Empfänger die Nachricht öffnet, tritt ein Öffnen- Ereignis ein.

Sie können Amazon Pinpoint so konfigurieren dass Informationen zu Ereignissen an Amazon Kinesis gesendet werden. Die Kinesis-Plattform bietet Dienste, mit denen Sie Daten aus AWS Diensten in Echtzeit sammeln, verarbeiten und analysieren können. Amazon Pinpoint kann Ereignisdaten an Firehose senden, das diese Daten an AWS Datenspeicher wie Amazon S3 oder Amazon Redshift streamt. Amazon Pinpoint kann auch Daten an Kinesis Data Streams streamen, das mehrere Datenströme zur Verarbeitung durch Analyseanwendungen aufnimmt und speichert.

Der Amazon Pinpoint-Ereignis-Stream enthält Informationen zu Benutzerinteraktionen mit Anwendungen (Apps), die Sie mit Amazon Pinpoint verbinden. Er enthält auch Informationen über alle Nachrichten, die Sie von Kampagnen, über jeden Kanal und von Journeys senden. Dies kann auch alle benutzerdefinierten Ereignisse umfassen, die Sie definiert haben. Schließlich enthält er Informationen über die Transaktions-E-Mail-Nachrichten und SMS-Nachrichten, die Sie senden.

> **Note**
>
> Amazon Pinpoint streamt keine Informationen über transaktionale Push-Benachrichtigungen oder Sprachnachrichten.

Dieses Kapitel enthält Informationen zum Einrichten von Amazon Pinpoint zum Streamen von Ereignisdaten zu Kinesis. Er enthält auch Beispiele für die Ereignisdaten, die Amazon Pinpoint streamt.

Themen

- App-Ereignisse
- Kampagnenereignisse
- Journey-Ereignisse
- E-Mail-Ereignisse
- SMS-Ereignisse

Einrichten von Ereignis-Streaming

Sie können Amazon Pinpoint so einrichten, dass Ereignisdaten an einen Amazon Kinesis Kinesis-Stream oder einen Amazon Data Firehose-Lieferstream gesendet werden. Amazon Pinpoint kann Ereignisdaten für Kampagnen, Reisen und transaktionale E-Mail- und SMS-Nachrichten senden.

Dieser Abschnitt enthält Informationen zum programmgesteuerten Konfigurieren von Ereignis-Streaming. Sie können auch die Amazon-Pinpoint-Konsole verwenden, um Ereignis-Streaming zu konfigurieren. Weitere Informationen zum Konfigurieren von Ereignis-Streaming mithilfe der Amazon-Pinpoint-Konsole finden Sie unter Ereignis-Stream-Einstellungen im Amazon-Pinpoint-Benutzerhandbuch.

Voraussetzungen

Die Beispiele in diesem Abschnitt erfordern die folgende Eingabe:

- Die Anwendungs-ID einer Anwendung, die mit Amazon Pinpoint und Berichtsereignissen integriert ist. Weitere Informationen über die Integration finden Sie unter Integrieren von Amazon Pinpoint in Ihre Anwendung.
- Der Amazon-Ressourcenname (ARN) eines Kinesis-Streams oder Firehose-Lieferstreams in Ihrem AWS Konto. Informationen zum Erstellen dieser Ressourcen finden Sie unter Creating and Managing Streams im Amazon Kinesis Data Streams Developer Guide oder Creating an Amazon Data Firehose Delivery Stream im Amazon Data Firehose Developer Guide.
- Der ARN einer AWS Identity and Access Management (IAM-) Rolle, die Amazon Pinpoint autorisiert, Daten an den Stream zu senden. Weitere Informationen zum Erstellen einer Rolle finden Sie unter IAM-Rolle für das Streamen von Ereignissen an Kinesis.

AWS CLI

Das folgende AWS CLI Beispiel verwendet den Befehl. put-event-stream Dieser Befehl konfiguriert Amazon Pinpoint, um Ereignisse an einen Kinesis-Stream zu senden:

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

AWS SDK for Java

Im folgenden Java-Beispiel wird Amazon Pinpoint so konfiguriert, dass Ereignisse an einen Kinesis-Stream gesendet werden:

```
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);
}
```

In diesem Beispiel wird ein `WriteEventStream`-Objekt erstellt, das die ARNs des Kinesis-Streams und der IAM-Rolle speichert. Das `WriteEventStream`-Objekt wird an ein `PutEventStreamRequest`-Objekt übergeben, um Amazon Pinpoint für das Streamen von Ereignissen für eine bestimmte Anwendung zu konfigurieren. Das `PutEventStreamRequest`-Objekt wird an die `putEventStream`-Methode des Amazon-Pinpoint-Client übergeben.

Sie können einen Kinesis-Stream mehreren Anwendungen zuordnen. Wenn Sie dies tun, sendet Amazon Pinpoint in base64 verschlüsselte Ereignisdaten von jeder Anwendung an den Stream, sodass Sie die Daten als Sammlung analysieren können. In der folgenden Beispielmethode wird eine Liste mit Anwendungs(App)-IDs akzeptiert und anhand der Methode aus dem vorherigen Beispiel (`createEventStream`) jeder Anwendung ein Stream zugewiesen:

```
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());
}
```

Sie können zwar einen Stream mehreren Anwendungen zuweisen, jedoch nicht mehrere Streams einer Anwendung.

Deaktivieren des Ereignis-Streaming

Wenn Sie einen Kinesis-Stream einer Anwendung zuweisen, können Sie Ereignis-Streaming für diese Anwendung deaktivieren. Amazon Pinpoint streamt die Ereignisse nicht mehr an Kinesis, aber Sie können die Ereignisanalysen mithilfe der Amazon-Pinpoint-Konsole anzeigen.

AWS CLI

Verwenden Sie den -Befehl:delete-event-stream

```
aws pinpoint delete-event-stream --application-id application-id
```

AWS SDK for Java

Verwenden Sie die deleteEventStreamMethode des Amazon Pinpoint Pinpoint-Clients:

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

App-Ereignisse

Nachdem Sie Ihre Anwendung (App) mit Amazon Pinpoint integriert haben, kann Amazon Pinpoint Ereignisdaten über Benutzeraktivitäten und Nachrichtenzustellungen für die Anwendung streamen.

Beispiel

Das JSON-Objekt für ein App-Ereignis enthält die Daten, die im folgenden Beispiel gezeigt werden.

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

App-Ereignisattribute

In diesem Abschnitt werden die Attribute definiert, die im Ereignis-Stream der App enthalten sind.

Attribut	Beschreibung
`event_type`	Der Ereignistyp. Die möglichen Werte sind: • _session.start: Der Endpunkt hat eine neue Sitzung gestartet. • _session.stop: Der Endpunkt hat eine Sitzung beendet. • _userauth.sign_in: Der Endpunkt, der bei Ihrer App angemeldet ist. • _userauth.sign_up: Ein neuer Endpunkt hat den Registrierungsprozess in Ihrer App abgeschlossen. • _userauth.auth_fail: Der Endpunkt hat versucht, sich bei Ihrer App anzumelden, konnte den Vorgang jedoch nicht abschließ en. • _monetization.purchase: Der Endpunkt hat in Ihrer App einen Kauf getätigt. • _session.pause: Der Endpunkt hat eine Sitzung angehalten. Pausierte Sitzungen können fortgesetzt werden, sodass Sie weiterhin Metriken erfassen können, ohne eine völlig neue Sitzung zu starten. • _session.resume: Der Endpunkt hat eine Sitzung fortgesetzt.
`event_timestamp`	Der Zeitpunkt, zu dem das Ereignis gemeldet wurde, angezeigt als Unix-Zeit in Millisekunden.

Attribut	Beschreibung
`arrival_timestamp`	De Zeitpunkt, zu dem das Ereignis von Amazon Pinpoint empfangen wurde, angezeigt als Unix-Zeit in Millisekunden.
`event_version`	Die Version des Ereignis-JSON-Schemas. **Tip** Prüfen Sie diese Version in der Anwendung, mit der Ihr Ereignis verarbeitet wird, damit Sie wissen, wann die Anwendung infolge eines Schema-Updates aktualisiert werden soll.
`application`	Informationen über das Amazon-Pinpoint-Projekt, das dem Ereignis zugeordnet ist. Weitere Informationen finden Sie in der Tabelle Application (Anwendung).
`client`	Informationen über den Endpunkt, der das Ereignis gemeldet hat. Weitere Informationen finden Sie in der Tabelle Client.
`device`	Informationen über das Gerät, das das Ereignis gemeldet hat. Weitere Informationen finden Sie in der Tabelle Device (Gerät).
`session`	Informationen über die Sitzung, die das Ereignis generiert hat. Weitere Informationen finden Sie in der Tabelle Session (Sitzung).
`attributes`	Attribute, die dem Ereignis zugeordnet sind. Bei Ereignissen, die von Ihren Apps gemeldet werden, enthält dieses Objekt benutzerd efinierte Attribute, die Sie definieren.

Attribut	Beschreibung
`metrics`	Metriken, die sich auf das Ereignis beziehen. Sie können Ihre Apps optional so konfigurieren, dass benutzerdefinierte Metriken an Amazon Pinpoint gesendet werden.

Anwendung

Enthält Informationen über das Amazon Pinpoint-Projekt, dem das Ereignis zugeordnet ist.

Attribut	Beschreibung
`app_id`	Die eindeutige ID des Amazon-Pinpoint-Projekts, das das Ereignis gemeldet hat.
`cognito_identity_pool_id`	Die ID des Amazon-Cognito-Identitätspools, dem der Endpunkt zugeordnet ist.
`package_name`	Der Name des App-Pakets, z.B. `com.example.my_app` .
`sdk`	Informationen über das SDK, das zum Melden des Ereignisses verwendet wurde. Weitere Informationen finden Sie in der Tabelle SDK.
`title`	Gibt den Namen der App an.
`version_name`	Der Name der Version der App, z. B. `V2.5`.
`version_code`	Die Versionsnummer der App, z. B. `3`.

SDK

Enthält Informationen über das SDK, das zum Melden des Ereignisses verwendet wurde.

Attribut	Beschreibung
`name`	Der Name des SDKs, das zum Melden des Ereignisses verwendet wurde.
`version`	Die Version des SDKs.

Client

Enthält Informationen über den Endpunkt, der das Ereignis generiert hat.

Attribut	Beschreibung
`client_id`	Die ID des Endpunkts.
`cognito_id`	Das Amazon-Cognito-ID-Token, das dem Endpunkt zugeordnet ist.

Gerät

Enthält Informationen über das Gerät des Endpunkts, der das Ereignis generiert hat.

Attribut	Beschreibung
`locale`	Enthält Informationen über die Sprach- und Regioneneinstellungen für das Gerät des Endpunkts. Weitere Informationen finden Sie in der Tabelle Locale (Gebietsschema).
`make`	Der Hersteller des Endpunktgeräts.
`model`	Der Modellbezeichner des Endpunktgeräts.
`platform`	Informationen zum Betriebssystem auf dem Endpunktgerät. Weitere Informationen finden Sie in der Tabelle Platform (Plattform).

Locale

Enthält Informationen über die Sprach- und Regioneneinstellungen für das Gerät des Endpunkts.

Attribut	Beschreibung
`code`	Die Gebietsschema-ID, die dem Gerät zugeordnet ist.
`country`	Das Land oder die Region, die dem Gebietsschema des Geräts zugeordnet ist.
`language`	Die Sprache, die dem Gebietsschema des Geräts zugeordnet ist.

Plattform

Enthält Informationen zum Betriebssystem auf dem Endpunktgerät.

Attribut	Beschreibung
`name`	Der Name des Betriebssystems auf dem Gerät.
`version`	Das Betriebssystemversion des Geräts.

Sitzung

Enthält Informationen über die Sitzung, die das Ereignis generiert hat.

Attribut	Beschreibung
`session_id`	Eine eindeutige ID, die die Sitzung identifiziert.
`start_timestamp`	Das Datum und die Uhrzeit, zu der die Sitzung begann, angezeigt als Unix-Zeit in Millisekunden.

Attribut	Beschreibung
`stop_timestamp`	Das Datum und die Uhrzeit, zu der die Sitzung beendet wurde, angezeigt als Unix-Zeit in Millisekunden.

Kampagnenereignisse

Wenn Sie mit Amazon Pinpoint Kampagnen über einen beliebigen Kanal senden, kann Amazon Pinpoint Ereignisdaten über diese Kampagnen streamen. Dies schließt Ereignisdaten für alle E-Mail- oder SMS-Nachrichten ein, die Sie von einer Kampagne senden. Ausführliche Informationen zu den Daten, die für diese Arten von Nachrichten von Amazon Pinpoint gestreamt werden, finden Sie unter the section called “E-Mail-Ereignisse” und the section called “SMS-Ereignisse”.

Beispielereignis

Das JSON-Objekt für ein Kampagnenereignis enthält die im folgenden Beispiel gezeigten Daten.

```
{
  "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"
}
```

Kampagnen-Ereignisattribute

In diesem Abschnitt werden die Attribute definiert, die im Ereignis-Stream der Kampagne enthalten sind.

Attribut	Beschreibung
`event_type`	Der Ereignistyp. Die möglichen Werte sind: • _campaign.send: Amazon Pinpoint hat die Kampagne ausgeführt. • _campaign.opened_notification: Bei Kampagnen mit Push-Benachrichtig ungen gibt dieser Ereignistyp an, dass der Empfänger auf die Benachrichtigung getippt hat, um sie zu öffnen. • _campaign.received_foreground: Bei Kampagnen mit Push-Benachrichtig ungen gibt dieser Ereignistyp an, dass der Empfänger die Nachricht als Vordergru ndbenachrichtigung erhalten hat. • _campaign.received_background: Bei Kampagnen mit Push-Benachrichtig ungen gibt dieser Ereignistyp an, dass der Empfänger die Nachricht als Hintergru ndbenachrichtigung erhalten hat.

Attribut	Beschreibung
	Note _campaign.opened_notification, _campaign.received_foreground und _campaign.received_backgrou nd werden nur zurückgegeben, wenn Sie AWS Amplify verwenden. Weitere Informationen zum Integrier en Ihrer App in AWS Amplify. Siehe Verbinden Ihrer Frontend-Anwendung mit Amazon Pinpoint mit AWS Amplify.
`event_timestamp`	Der Zeitpunkt, zu dem das Ereignis gemeldet wurde, angezeigt als Unix-Zeit in Millisekunden.
`arrival_timestamp`	De Zeitpunkt, zu dem das Ereignis von Amazon Pinpoint empfangen wurde, angezeigt als Unix-Zeit in Millisekunden.
`event_version`	Die Version des Ereignis-JSON-Schemas. **Tip** Prüfen Sie diese Version in der Anwendung, mit der Ihr Ereignis verarbeitet wird, damit Sie wissen, wann die Anwendung infolge eines Schema-Updates aktualisiert werden soll.

Attribut	Beschreibung
`application`	Informationen über das Amazon-Pinpoint-Projekt, das dem Ereignis zugeordnet ist. Weitere Informationen finden Sie in der Tabelle Application (Anwendung).
`client`	Informationen über den Endpunkt, dem das Ereignis zugeordnet ist. Weitere Informationen finden Sie in der Tabelle Client.
`device`	Informationen über das Gerät, das das Ereignis gemeldet hat. Bei Kampagnen- und Transakti onsnachrichten ist dieses Objekt leer.
`session`	Informationen über die Sitzung, die das Ereignis generiert hat. Bei Kampagnen ist dieses Objekt leer.
`attributes`	Attribute, die dem Ereignis zugeordnet sind. Bei Ereignissen, die von einer Ihrer Apps gemeldet werden, kann dieses Objekt benutzerdefinierte Attribute enthalten, die von der App definiert werden. Bei Ereignissen, die beim Senden einer Kampagne erstellt werden, enthält dieses Objekt Attribute, die der Kampagne zugeordne t sind. Bei Ereignissen, die generiert werden, wenn Sie Transaktionsnachrichten senden, enthält dieses Objekt Informationen, die sich auf die Nachricht selbst beziehen. Weitere Informationen finden Sie in der Tabelle Attributes (Attribute).

Attribut	Beschreibung
`client_context`	Enthält ein `custom`-Objekt, das eine `endpoint`-Eigenschaft enthält. Die `endpoint`-Eigenschaft enthält den Inhalt des Endpunktd atensatzes für den Endpunkt, an den die Kampagne gesendet wurde.
`awsAccountId`	Die ID des AWS-Kontos, das zum Senden der Nachricht verwendet wurde.

Anwendung

Enthält Informationen über das Amazon Pinpoint-Projekt, dem das Ereignis zugeordnet ist.

Attribut	Beschreibung
`app_id`	Die eindeutige ID des Amazon-Pinpoint-Pr ojekts, das das Ereignis gemeldet hat.
`sdk`	Das SDK, das zum Melden des Ereignisses verwendet wurde.

Attribute

Enthält Informationen über die Kampagne, die das Ereignis generiert hat.

Attribut	Beschreibung
`treatment_id`	Wenn die Nachricht mit einer A/B-Testk ampagne gesendet wurde, stellt dieser Wert die Behandlungsnummer der Nachricht dar. Für Standardkampagnen ist dieser Wert `0`.
`campaign_activity_id`	Die eindeutige ID, die Amazon Pinpoint generiert, wenn das Ereignis eintritt.

Attribut	Beschreibung
`delivery_type`	Die Bereitstellungsmethode für die Kampagne. Verwechseln Sie dieses Attribut nicht mit dem `ChannelType` -Feld, das unter der `endpoint`-Eigenschaft von `client_context` angegeben ist. Das `ChannelType` -Feld basiert normalerweise auf dem Endpunkt, an den die Nachricht gesendet wird. Bei Kanälen, die nur einen Endpunkttyp unterstützen, haben die Felder `delivery_type` und `ChannelType` denselben Wert. Für den E-Mail-Kanal haben die Felder `delivery_type` und `ChannelType` beispielsweise denselben Wert EMAIL. Diese Bedingung gilt jedoch nicht immer für Kanäle, die unterschiedliche Endpunkttypen unterstützen, z. B. benutzerdefinierte Kanäle. Sie können einen benutzerdefinierten Kanal für verschiedene Endpunkte verwenden, z. B. EMAIL, SMS, CUSTOM usw. In diesem Fall identifiziert der `delivery_type` ein benutzerdefiniertes Bereitstellungsereignis, CUSTOM, und der `ChannelType` gibt den Typ des Endpunkts an, an den die Kampagne gesendet wurde, z. B. EMAIL, SMS, CUSTOM usw. Weitere Informationen zum Erstellen von benutzerdefinierten Kanälen finden Sie unter Erstellen von benutzerdefinierten Kanälen. Die möglichen Werte sind: • EMAIL • SMS • ADM

Attribut	Beschreibung
	• APNS • APNS_SANDBOX • APNS_VOIP • APNS_VOIP_SANDBOX • VOICE • GCM • BAIDU • PUSH • CUSTOM
`campaign_id`	Die eindeutige ID der Kampagne, von der die Nachricht gesendet wurde.

Attribut	Beschreibung
`campaign_send_status`	Gibt den Status der Kampagne für den Zielendpunkt an. Mögliche Werte sind: • SUCCESS: Die Kampagne wurde erfolgreich an den Endpunkt gesendet. • FAILURE: Die Kampagne wurde nicht an den Endpunkt gesendet. • DAILY_CAP: Die Kampagne wurde nicht an den Endpunkt gesendet, da bereits die maximale Anzahl an täglichen Nachrichten an den Endpunkt gesendet wurde. • EXPIRED: Die Kampagne wurde nicht an den Endpunkt gesendet, da das Senden die Einstellungen für die maximale Dauer oder die Senderate für die Kampagne überschre iten würde. • QUIET_TIME: Die Kampagne wurde aufgrund von Ruhezeitbeschränkungen nicht an den Endpunkt gesendet. • HOLDOUT: Die Kampagne wurde nicht an den Endpunkt gesendet, da der Endpunkt Mitglied der Holdout-Gruppe war. • DUPLICATE_ADDRESS: Das Segment enthält doppelte Endpunktadressen. Die Kampagne wurde einmal an die Endpunkta dresse gesendet. • QUIET_TIME: Die Kampagne wurde aufgrund von Ruhezeitbeschränkungen nicht an den Endpunkt gesendet. • CAMPAIGN_CAP: Die Kampagne wurde nicht an den Endpunkt gesendet, da von dieser Kampagne bereits die maximale

Attribut	Beschreibung
	Anzahl an Nachrichten an den Endpunkt gesendet wurde. • FAILURE_PERMANENT: Beim Senden an den Endpunkt ist ein permanenter Fehler aufgetreten. • TRANSIENT_FAILURE: Beim Senden an den Endpunkt ist ein vorübergehender Fehler aufgetreten. • THROTTLED: Das Senden wurde gedrosselt. • UNKNOWN: Unbekannter Fehler. • HOOK_FAILURE: Der Kampagnen-Hook ist fehlgeschlagen. • CUSTOM_DELIVERY_FAILURE: Die benutzerdefinierte Bereitstellung ist fehlgeschlagen. • RECOMMENDATION_FAILURE: Empfehlung fehlgeschlagen. • UNSUPPORTED_CHANNEL: Kanal wird nicht unterstützt.

Client

Enthält Informationen über den Endpunkt, auf den die Kampagne ausgerichtet war.

Attribut	Beschreibung
`client_id`	Die ID des Endpunkts, an den die Kampagne gesendet wurde.

Journey-Ereignisse

Wenn Sie eine Journey veröffentlichen, kann Amazon Pinpoint Ereignisdaten über die Journey streamen. Dies schließt Ereignisdaten für alle E-Mail-, SMS-, Push- oder benutzerdefinierten Nachrichten ein, die Sie von der Journey senden.

Im Folgenden finden Sie Informationen zu den Daten, die Amazon Pinpoint streamt:

- Informationen zu E-Mail-Nachrichten finden Sie unter the section called “E-Mail-Ereignisse”.
- Informationen zu SMS-Nachrichten finden Sie unter SMS-Ereignisse.

Beispielereignis

Das JSON-Objekt für ein Journey-Ereignis enthält die im folgenden Beispiel gezeigten Daten.

```
{
   "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"
}
```

Journey-Ereignisattribute

In diesem Abschnitt werden die Attribute definiert, die in den Ereignis-Stream-Daten enthalten sind, die Amazon Pinpoint für eine Journey generiert.

Attribut	Beschreibung
`event_type`	Der Ereignistyp. Bei Journey-Ereignissen ist der Wert für dieses Attribut immer `_journey.send` , was angibt, dass Amazon Pinpoint die Reise ausgeführt hat.
`event_timestamp`	Der Zeitpunkt, zu dem das Ereignis gemeldet wurde, angezeigt als Unix-Zeit in Millisekunden.
`arrival_timestamp`	De Zeitpunkt, zu dem das Ereignis von Amazon Pinpoint empfangen wurde, angezeigt als Unix-Zeit in Millisekunden.
`event_version`	Die Version des Ereignis-JSON-Schemas. **Tip** Prüfen Sie diese Version in der Anwendung, mit der Ihr Ereignis verarbeitet wird, damit Sie wissen,

Attribut	Beschreibung
	wann die Anwendung infolge eines Schema-Updates aktualisiert werden soll.
`application`	Informationen über das Amazon-Pinpoint-Projekt, das dem Ereignis zugeordnet ist. Weitere Informationen finden Sie in der Tabelle Application (Anwendung).
`client`	Informationen über den Endpunkt, dem das Ereignis zugeordnet ist. Weitere Informationen finden Sie in der Tabelle Client.
`device`	Informationen über das Gerät, das das Ereignis gemeldet hat. Bei Journeys ist dieses Objekt leer.
`session`	Informationen über die Sitzung, die das Ereignis generiert hat. Bei Journeys ist dieses Objekt leer.
`attributes`	Attribute, die der Journey und der Journey-Aktivität zugeordnet sind, die das Ereignis generiert hat. Weitere Informationen finden Sie in der Tabelle Attributes (Attribute).
`client_context`	Enthält ein `custom`-Objekt, das eine `endpoint`-Eigenschaft enthält. Die `endpoint`-Eigenschaft enthält den Inhalt des Endpunktd atensatzes für den Endpunkt, der dem Ereignis zugeordnet ist.
`awsAccountId`	Die ID des AWS Kontos, das für die Ausführun g der Reise verwendet wurde.

Anwendung

Enthält Informationen über das Amazon-Pinpoint-Projekt, dem das Ereignis zugeordnet ist.

Attribut	Beschreibung
`app_id`	Die eindeutige ID des Amazon-Pinpoint-Projekts, das das Ereignis gemeldet hat.
`sdk`	Das SDK, das zum Melden des Ereignisses verwendet wurde.

Client

Enthält Informationen über den Endpunkt, mit dem das Ereignis verknüpft ist.

Attribut	Beschreibung
`client_id`	Die ID des Endpunkts.

Attribute

Enthält Informationen über die Journey, die das Ereignis generiert hat.

Attribut	Beschreibung
`journey_run_id`	Die eindeutige ID des Journeylaufs, der das Ereignis generiert hat. Amazon Pinpoint generiert diese ID automatisch und weist sie jedem neuen Lauf einer Journey zu.
`journey_send_status`	Gibt den Zustellungsstatus der Nachricht an, die dem Ereignis zugeordnet ist. Mögliche Werte sind: • SUCCESS: Die Nachricht wurde erfolgreich an den Endpunkt gesendet.

Attribut	Beschreibung
	• FAILURE: Die Nachricht wurde nicht an den Endpunkt gesendet, da ein Fehler aufgetreten ist. • CUSTOM_DELIVERY_FAILURE: Die benutzerdefinierte Bereitstellung ist fehlgeschlagen. • FAILURE_PERMANENT: Beim Senden an den Endpunkt ist ein permanenter Fehler aufgetreten. **Tip** Sie können nach Ereignissen mit dem Status FAILURE_PERMANENT filtern und diese auf 403 `journey_send_status_code` setzen, um festzustellen, ob eine Zugriffsrichtlinie und eine Rollenverletzung vorliegt. Bei ausgehenden Kampagnen mit Spracheingabe sind diese Ausnahmen typisch für Fälle, in denen die Connect-Kampagnenausführungsrolle, die Amazon Pinpoint Pinpoint-Journeys Amazon Connect Connect-Kampagnen verknüpft, versehentlich für die Ausführung von Reisen während des Fluges gelöscht wird. • THROTTLED: Das Senden wurde gedrosselt. • UNSUPPORTED_CHANNEL: Kanal wird nicht unterstützt.

Attribut	Beschreibung
	• DAILY_CAP: Die Nachricht wurde nicht an den Endpunkt gesendet, da das Senden der Nachricht die maximale Anzahl von Nachricht en überschreiten würde, die die Journey oder das Projekt während eines 24-Stunde n-Zeitraums an einen einzelnen Endpunkt senden kann. • QUIET_TIME: Die Nachricht wurde aufgrund von Beschränkungen für die Journey oder das Projekt nicht gesendet. • QUIET_TIME_MISSING_TIMEZONE: Die Nachricht wurde nicht gesendet, da die Zeitzonenschätzung keine Zeitzone für den Endpunkt schätzen konnte und die Ruhezeit aktiviert ist.
`journey_id`	Die eindeutige ID der Journey, die das Ereignis generiert hat.
`journey_activity_id`	Die eindeutige ID der Journey-Aktivität, die das Ereignis generiert hat.
`journey_activity_type`	Der Journey-Aktivitätstyp des Ereignisses. Dies kann EMAIL, SMS, PUSH, CONTACT_C ENTER oder CUSTOM sein. **Note** VOICE ist kein unterstützter Journey-A ktivitätstyp.
`journey_send_status_message`	Die Beschreibung des Status des Sendeerei gnisses.
`journey_send_status_code`	Der HTTP-Statuscode der Anfrage.

E-Mail-Ereignisse

Wenn Sie E-Mail-Nachrichten senden, streamt Amazon Pinpoint Daten, die zusätzliche Informationen zu den folgenden Ereignistypen für diese Nachrichten bereitstellen:

- Sends (Sendevorgänge)
- Deliveries (Zustellungen)
- Unzustellbarkeit
- Complaints (Beschwerden)
- Opens (Öffnungsvorgänge)
- Clicks (Klickvorgänge)
- Ablehnungen
- Abbestellungen
- Rendering failures (Rendern von Fehlern)

Die Ereignistypen in der obigen Liste werden unter E-Mail-Ereignisattribute im Detail erläutert.

Abhängig von der API und den Einstellungen, die Sie zum Senden von E-Mail-Nachrichten verwenden, werden möglicherweise zusätzliche Ereignistypen oder andere Daten angezeigt. Wenn Sie beispielsweise Nachrichten mithilfe von Konfigurationssätzen senden, in denen Ereignisdaten zu Amazon Kinesis veröffentlicht werden, z. B. von Amazon Simple Email Service (Amazon SES), können die Daten auch Ereignisse für Vorlagenrendering-Fehler enthalten. Weitere Informationen zu diesen Daten finden Sie unter Überwachen mithilfe der Amazon-SES-Ereignisveröffentlichung im Entwicklerhandbuch für Amazon Simple Email Service.

Beispielereignisse

E-Mail-Sendevorgang

Das JSON-Objekt für ein E-Mail-Sendeereignis enthält die Daten wie im folgenden Beispiel.

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

E-Mail zugestellt

Das JSON-Objekt für ein E-Mail-Zustellereignis enthält die Daten wie im folgenden Beispiel.

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

E-Mail-Klick

Das JSON-Objekt für ein E-Mail-Klickereignis enthält die Daten wie im folgenden Beispiel.

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

E-Mail-Öffnungsvorgang

Das JSON-Objekt für ein E-Mail-Öffnungsvorgangsereignis enthält die Daten wie im folgenden Beispiel.

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

E-Mail-Ereignisattribute

In diesem Abschnitt werden die Attribute definiert, die in den Ereignis-Stream-Daten enthalten sind, die Amazon Pinpoint generiert, wenn Sie E-Mail-Nachrichten senden.

Attribut	Beschreibung
`event_type`	Der Ereignistyp. Die möglichen Werte sind: • _email.send: Amazon Pinpoint hat die Nachricht akzeptiert und versucht, sie dem Empfänger zuzustellen. • _email.delivered: Die Nachricht wurde an den Empfänger zugestellt. • _email.rejected: Amazon Pinpoint hat festgestellt, dass die Nachricht Malware enthält, und versucht nicht, sie zu versenden. • _email.hardbounce: Ein permanentes Problem hinderte Amazon Pinpoint daran, die Nachricht zuzustellen. Amazon Pinpoint wird nicht erneut versuchen, die Nachricht zuzustellen. • _email.softbounce: Ein temporäres Problem hinderte Amazon Pinpoint daran, die Nachricht zuzustellen. Amazon Pinpoint wird eine Zeit lang versuchen, die Nachricht erneut zuzustellen. Wenn die Nachricht immer noch nicht zugestellt werden kann, werden keine weiteren Versuche unternomm en. Der endgültige Status der E-Mail lautet dann SOFTBOUNCE. • _email.complaint: Der Empfänger hat die Nachricht erhalten und sie anschließend bei seinem E-Mail-Anbieter als Spam gemeldet (z. B. mithilfe des Features „Spam melden“ des E-Mail-Clients) • _email.open: Der Empfänger hat die Nachricht erhalten und sie in einem E-Mail-Cl ient geöffnet.

Attribut	Beschreibung
	• _email.click: Der Empfänger hat die Nachricht erhalten und auf einen darin enthaltenen Link geklickt. • _email.unsubscribe: Der Empfänger hat die Nachricht erhalten und auf einen darin Link zum Beenden des Abonnements geklickt. • _email.rendering_failure: Die E-Mail wurde aufgrund eines Rendering-Fehlers nicht gesendet. Dies kann auftreten, wenn Vorlagendaten fehlen oder die Vorlagenparameter nicht mit den Daten übereinstimmen.
`event_timestamp`	Die Uhrzeit, zu der die Nachricht gesendet wurde, wird als Unix-Zeit in Millisekunden angezeigt. Dieser Wert ist in der Regel für alle Ereignisse identisch, die für eine Nachricht generiert werden.
`arrival_timestamp`	De Zeitpunkt, zu dem das Ereignis von Amazon Pinpoint empfangen wurde, angezeigt als Unix-Zeit in Millisekunden.
`event_version`	Die Version des Ereignis-JSON-Schemas. **Tip** Prüfen Sie diese Version in der Anwendung, mit der Ihr Ereignis verarbeitet wird, damit Sie wissen, wann die Anwendung infolge eines Schema-Updates aktualisiert werden soll.

Attribut	Beschreibung
`application`	Informationen über das Amazon-Pinpoint-Projekt, das dem Ereignis zugeordnet ist. Weitere Informationen finden Sie in der Tabelle Application (Anwendung) .
`client`	Informationen zum App-Client, der auf dem Gerät installiert ist, das das Ereignis gemeldet hat. Weitere Informationen finden Sie in der Tabelle Client.
`device`	Informationen über das Gerät, das das Ereignis gemeldet hat. Weitere Informationen finden Sie in der Tabelle Device (Gerät). Bei E-Mail-Ereignissen ist dieses Objekt leer.
`session`	Bei E-Mail-Ereignissen ist dieses Objekt leer.
`attributes`	Attribute, die dem Ereignis zugeordnet sind. Weitere Informationen finden Sie in der Tabelle Attributes (Attribute). Bei Ereignissen, die von einer Ihrer Apps gemeldet werden, kann dieses Objekt benutzerdefinierte Attribute enthalten, die von der App definiert werden. Bei Ereignissen, die beim Senden einer Nachricht aus einer Kampagne oder Journey erstellt werden, enthält dieses Objekt Attribute, die der Kampagne oder Journey zugeordnet sind. Bei Ereignissen, die generiert werden, wenn Sie Transaktionsnachrichten senden, enthält dieses Objekt Informationen, die sich auf die Nachricht selbst beziehen.

Attribut	Beschreibung
`client_context`	Bei E-Mail-Ereignissen enthält dieses Objekt ein `custom`-Objekt, das ein `legacy_id entifier` Attribut enthält. Der Wert für das `legacy_identifier` -Attribut ist die ID des Projekts, von dem die Nachricht gesendet wurde.
`facets`	Zusätzliche Informationen über die Nachricht, z. B. die E-Mail-Header. Weitere Informationen finden Sie in der Tabelle Facets.
`awsAccountId`	Die ID des AWS-Kontos, das zum Senden der Nachricht verwendet wurde.

Anwendung

Enthält Informationen über das Amazon Pinpoint-Projekt, dem das Ereignis zugeordnet ist.

Attribut	Beschreibung
`app_id`	Die eindeutige ID des Amazon-Pinpoint-Pr ojekts, das das Ereignis gemeldet hat.
`sdk`	Das SDK, das zum Melden des Ereignisses verwendet wurde. Wenn Sie eine Transaktions-E-Mail-Nachricht durch einen direkten Amazon-Pinpoint-API-Aufruf oder über die Amazon-Pi npoint-Konsole senden, ist dieses Objekt leer.

Attribute

Enthält Informationen über die Kampagne oder Journey, die das Ereignis generiert hat.

Kampagne

Enthält Informationen über die Kampagne, die das Ereignis generiert hat.

Attribut	Beschreibung
`feedback`	Bei `_email.click` -Ereignissen ist der Wert für dieses Attributs die URL des Links, auf den der Empfänger in der Nachricht geklickt hat, um das Ereignis zu generieren. Bei anderen Ereignissen stellt dieser Wert den Ereignistyp dar (z. B. `received`, `opened` oder `clicked`).
`treatment_id`	Wenn die Nachricht mit einer A/B-Testk ampagne gesendet wurde, stellt dieser Wert die Behandlungsnummer der Nachricht dar. Für Standardkampagnen und Transaktions-E-Mail-Nachrichten ist dieser Wert `0`.
`campaign_activity_id`	Die eindeutige ID, die Amazon Pinpoint generiert, wenn das Ereignis eintritt.
`campaign_id`	Die eindeutige ID der Kampagne, die die Nachricht gesendet hat.

Journey

Enthält Informationen über die Journey, die das Ereignis generiert hat.

Attribut	Beschreibung
`journey_run_id`	Die eindeutige ID des Journeylaufs, der die Nachricht gesendet hat. Amazon Pinpoint generiert diese ID automatisch und weist sie jedem neuen Lauf einer Journey zu.
`feedback`	Bei `_email.click` -Ereignissen ist der Wert für dieses Attributs die URL des Links, auf den der Empfänger in der Nachricht geklickt hat, um das Ereignis zu generieren. Bei anderen Ereignissen stellt dieser Wert den Ereignist

Attribut	Beschreibung
	yp dar (z. B. `received`, `delivered` oder `opened`).
`journey_id`	Die eindeutige ID der Journey, die die Nachricht gesendet hat.
`journey_activity_id`	Die eindeutige ID der Journey-Aktivität, die die Nachricht gesendet hat.

Client

Die eindeutige Kennung des Kunden, auf den die Kampagne oder Journey abzielte.

Attribut	Beschreibung
`client_id`	Die Client-ID Der Wert ist die Endpunkt-ID für Kampagnen und Journeys und für transakti onales Senden ist es eine UUID.

Facets

Enthält Informationen zur Nachricht und zum Ereignistyp.

Attribut	Beschreibung
`email_channel`	Enthält ein Objekt `mail_event` , das zwei Objekte enthält: `mail` und ein Objekt, das dem Ereignistyp entspricht.

Mail

Enthält Informationen über den Inhalt der E-Mail-Nachricht sowie Metadaten zu der Nachricht.

Attribut	Beschreibung
`message_id`	Die eindeutige ID der Nachricht. Amazon Pinpoint generiert diese ID automatisch, wenn es die Nachricht akzeptiert.
`message_send_timestamp`	Das Datum und die Uhrzeit, zu der die Nachricht gesendet wurde, in dem Format angezeigt, das in RFC 822 angegeben ist.
`from_address`	Die E-Mail-Adresse, von der die Nachricht gesendet wurde.
`destination`	Ein Array mit den E-Mail-Adressen, an die die Nachricht gesendet wurde.
`headers_truncated`	Ein boolescher Wert, der angibt, ob die E-Mail-Header abgeschnitten wurden.
`headers`	Ein Objekt, das mehrere Name-Wert-Paare enthält, die den Headern in der Nachricht entsprechen. Dieses Objekt enthält in der Regel Informationen zu den folgenden Headern: • `From`: Die E-Mail-Adresse des Absenders. • `To`: Die E-Mail-Adresse des Empfängers. • `Subject`: Die Betreffzeile der E-Mail. **Tip** Der Betreff-Header ist bei campaign _email.send-Ereignissen nicht enthalten. • `MIME-Version` : Gibt an, dass die Nachricht im MIME-Format vorliegt. Wenn

Attribut	Beschreibung
	dieser Header vorhanden ist, lautet der Wert immer `1.0`. • `Content-Type` : Der MIME-Medientyp des Nachrichteninhalts.
`common_headers`	Enthält Informationen zu verschiedenen allgemeinen Kopfzeilen für E-Mail-Nachrichten . Die Informationen können das Datum, an dem die Nachricht gesendet wurde, sowie die Zeilen „An“, „Von“ und „Betreff“ der Nachricht enthalten.

SMS-Ereignisse

Wenn der SMS-Kanal für ein Projekt aktiviert ist, kann Amazon Pinpoint Ereignisdaten über SMS-Nachrichtenzustellungen für das Projekt streamen. Es kann bis zu 72 Stunden dauern, bis von Mobilfunkanbietern generierte SMS-Ereignisse empfangen werden. Sie sollten nicht verwendet werden, um festzustellen, ob es zu Verzögerungen bei der Zustellung ausgehender Nachrichten kommt. Wenn Amazon Pinpoint nach 72 Stunden kein finales Ereignis von einem Mobilfunkanbieter erhalten hat, gibt der Service automatisch einen UNKNOWN record_status zurück, da wir nicht wissen, was mit dieser Nachricht passiert ist.

Beispiel

Das JSON-Objekt für ein SMS-Ereignis enthält die Daten wie im folgenden Beispiel.

```
{
  "event_type": "_SMS.SUCCESS",
  "event_timestamp": 1553104954322,
  "arrival_timestamp": 1553104954064,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "123456789012"
```

```
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "sender_request_id": "565d4425-4b3a-11e9-b0a5-example",
    "campaign_activity_id": "cbcfc3c5e3bd48a8ae2b9cb41example",
    "origination_phone_number": "+12065550142",
    "destination_phone_number": "+14255550199",
    "record_status": "DELIVERED",
    "iso_country_code": "US",
    "treatment_id": "0",
    "number_of_message_parts": "1",
    "message_id": "1111-2222-3333",
    "message_type": "Transactional",
    "campaign_id": "52dc44b35c4742c98c5935269example"
  },
  "metrics": {
    "price_in_millicents_usd": 645.0
  },
  "awsAccountId": "123456789012"
}
```

SMS-Ereignisattribute

In diesem Abschnitt werden die Attribute definiert, die in den Ereignis-Stream-Daten enthalten sind, die Amazon Pinpoint generiert, wenn Sie SMS-Nachrichten senden.

Veranstaltung

Attribut	Beschreibung
`event_type`	Der Ereignistyp. Die möglichen Werte sind: • _SMS.BUFFERED: Die Nachricht wird noch an den Empfänger zugestellt. • _SMS.SUCCESS: Die Nachricht wurde vom Mobilfunkanbieter erfolgreich angenommen/ an den Empfänger zugestellt. • _SMS.FAILURE: Amazon Pinpoint konnte die Nachricht nicht an den Empfänger

Attribut	Beschreibung
	zustellen. Weitere Informationen über den Fehler, aufgrund dessen die Nachricht nicht zugestellt werden konnte, finden Sie unter `attributes.record_status` . • _SMS.OPTOUT: Der Kunde hat die Nachricht erhalten und mit dem Senden des Opt-Out-Schlüsselworts (normalerweise „STOP“) geantwortet.
`event_timestamp`	Der Zeitpunkt, zu dem das Ereignis gemeldet wurde, angezeigt als Unix-Zeit in Millisekunden.
`arrival_timestamp`	De Zeitpunkt, zu dem das Ereignis von Amazon Pinpoint empfangen wurde, angezeigt als Unix-Zeit in Millisekunden.
`event_version`	Die Version des Ereignis-JSON-Schemas. **Tip** Prüfen Sie diese Version in der Anwendung, mit der Ihr Ereignis verarbeitet wird, damit Sie wissen, wann die Anwendung infolge eines Schema-Updates aktualisiert werden soll.
`application`	Informationen über das Amazon-Pinpoint-Projekt, das dem Ereignis zugeordnet ist. Weitere Informationen finden Sie in der Tabelle Application (Anwendung).

Attribut	Beschreibung
`client`	Informationen zum App-Client, der auf dem Gerät installiert ist, das das Ereignis gemeldet hat. Weitere Informationen finden Sie in der Tabelle Client.
`device`	Informationen über das Gerät, das das Ereignis gemeldet hat. Weitere Informationen finden Sie in der Tabelle Device (Gerät). Bei SMS-Ereignissen ist dieses Objekt leer.
`session`	Bei SMS-Ereignissen ist dieses Objekt leer.
`attributes`	Attribute, die dem Ereignis zugeordnet sind. Bei Ereignissen, die von einer Ihrer Apps gemeldet werden, kann dieses Objekt benutzerdefinierte Attribute enthalten, die von der App definiert werden. Bei Ereignissen, die beim Senden einer Kampagne erstellt werden, enthält dieses Objekt Attribute, die der Kampagne zugeordne t sind. Bei Ereignissen, die generiert werden, wenn Sie Transaktionsnachrichten senden, enthält dieses Objekt Informationen, die sich auf die Nachricht selbst beziehen. Weitere Informationen finden Sie in der Tabelle Attributes (Attribute).
`metrics`	Zusätzliche Metriken, die dem Ereignis zugeordnet sind. Weitere Informationen finden Sie in der Tabelle Metrics (Metriken).
`awsAccountId`	Die ID des AWS-Kontos, das zum Senden der Nachricht verwendet wurde.

Anwendung

Enthält Informationen über das Amazon-Pinpoint-Projekt, dem das Ereignis zugeordnet ist, und gegebenenfalls das SDK, das zum Melden des Ereignisses verwendet wurde.

Attribut	Beschreibung
`app_id`	Die eindeutige ID des Amazon-Pinpoint-Pr ojekts, das das Ereignis gemeldet hat.
`sdk`	Das SDK, das zum Melden des Ereignisses verwendet wurde. Wenn Sie eine Transaktions-SMS-Nachricht durch einen direkten Amazon-Pinpoint-API-Aufruf oder über die Amazon-Pi npoint-Konsole senden, ist dieses Objekt leer.

Attribute

Enthält Informationen zu den Attributen, die dem Ereignis zugeordnet sind.

Attribut	Beschreibung
`sender_request_id`	Eine eindeutige ID, die der Anforderung zum Senden der SMS-Nachricht zugeordnet ist.
`campaign_activity_id`	Die eindeutige ID der Aktivität innerhalb der Kampagne.
`origination_phone_number`	Die Telefonnummer, von der die Nachricht gesendet wurde.
`destination_phone_number`	Die Telefonnummer, an die Sie versucht haben, die Nachricht zu senden.
`record_status`	Zusätzliche Informationen über den Status der Nachricht. Mögliche Werte sind: • SUCCESSFUL/DELIVERED: Die Nachricht wurde erfolgreich zugestellt.

Attribut	Beschreibung
	• PENDING: Die Nachricht wurde noch nicht an das Gerät des Empfängers zugestellt. • INVALID: Die Zieltelefonnummer ist ungültig. • UNREACHABLE: Das Gerät des Empfänger s ist derzeit nicht erreichbar oder nicht verfügbar. Beispielsweise könnte das Gerät ausgeschaltet oder vom Netzwerk getrennt sein. Sie können versuchen, die Nachricht später erneut zu senden. • UNKNOWN: Es ist ein Fehler aufgetreten, der die Zustellung der Nachricht verhindert hat. Dieser Fehler ist in der Regel vorüberge hend und Sie können versuchen, die Nachricht später erneut zu senden. • BLOCKED Das Gerät des Empfängers blockiert SMS-Nachrichten von der ursprüngl ichen Nummer. • CARRIER_UNREACHABLE: Ein Problem mit dem Mobilfunknetz des Empfängers verhinderte die Zustellung der Nachricht. Dieser Fehler ist in der Regel vorübergehend und Sie können versuchen, die Nachricht später erneut zu senden. • SPAM: Der Mobilfunkanbieter des Empfängers hat den Inhalt der Nachricht als Spam identifiziert und die Zustellung der Nachricht blockiert. • INVALID_MESSAGE: Der Text der SMS-Nachricht ist ungültig und kann nicht zugestellt werden. • CARRIER_BLOCKED: Der Mobilfunkanbieter des Empfängers hat die Zustellung dieser Nachricht blockiert. Dies tritt häufig auf, wenn

Attribut	Beschreibung
	der Anbieter den Inhalt der Nachricht als unerwünscht oder böswillig identifiziert. • TTL_EXPIRED: Die SMS-Nachricht konnte innerhalb eines bestimmten Zeitraums nicht zugestellt werden. Dieser Fehler ist in der Regel vorübergehend und Sie können versuchen, die Nachricht später erneut zu senden. • MAX_PRICE_EXCEEDED: Das Senden der Nachricht hätte zu einer Gebühr geführt, die das monatliche SMS-Ausgabenkontingent für Ihr Konto überschritten hätte. Sie können eine Erhöhung dieses Kontingents anfordern , indem Sie das Verfahren unter Anfordern von Erhöhungen Ihres monatlichen SMS-Ausgabenkontingents im Amazon-Pinpoint-Benutzerhandbuch ausführen. • OPTED_OUT: Die SMS-Nachricht wurde nicht gesendet, weil der Empfänger keine Nachrichten von Ihnen empfangen möchte. • NO_QUOTA_LEFT_ON_ACCOUNT: Auf Ihrem Konto ist nicht mehr genügend Ausgabenkontingent vorhanden, um die Nachricht zu senden. Sie können eine Erhöhung dieses Kontingents anfordern, indem Sie das Verfahren unter Anfordern von Erhöhungen Ihres monatlichen SMS-Ausgabenkontingents im Amazon-Pinpoint-Benutzerhandbuch ausführen. • NO_ORIGINATION_IDENTITY_AVAILABLE_TO_SEND: Ihr Konto enthält keine Telefonnummer, mit der die Nachricht an das Ziel gesendet werden kann.

Attribut	Beschreibung
	• DESTINATION_COUNTRY_NOT_SUPPORTED: Das Zielland ist gesperrt. Informationen zu allen unterstützten Ländern finden Sie unter Unterstützte Länder und Regionen (SMS-Kanal) • ACCOUNT_IN_SANDBOX: Ihr Konto befindet sich in der Sandbox und kann nur an verifizierte Zielnummern senden. Sie können die Zielnummer in der Amazon-Pinpoint-Konsole überprüfen oder den Prozess starten, um das Konto aus der Sandbox zu verschieben, siehe Übergang von der Amazon-Pinpoint-SMS-Sandbox zur Produktion. • RATE_EXCEEDED: Sie haben versucht, die Nachricht zu schnell zu senden, und wurden gedrosselt. Sie müssen Ihre Anrufrate verringern. Einzelheiten zu unseren Limits finden Sie unter Message Parts per Second (MPS)-Limits. • INVALID_ORIGINATION_IDENTITY: Die angegebene Ursprungsidentität ist ungültig. • ORIGINATION_IDENTITY_DOES_NOT_EXIST: Die angegebene Ursprungsidentität existiert nicht. • INVALID_DLT_PARAMETERS: Ungültige DLT-Parameter (erforderlich für Ziele in Indien) wurden angegeben. • INVALID_PARAMETERS: Es wurden ungültige Parameter angegeben. • ACCESS_DENIED: Ihr Konto ist für das Senden von Nachrichten gesperrt. Wenden Sie sich an den Kundensupport, um die

Attribut	Beschreibung
	Ursache herauszufinden und das Problem zu lösen. • INVALID_KEYWORD: Das angegebene Schlüsselwort ist ungültig. Das Schlüsselwort hat möglicherweise ein falsches Format oder ist in Ihrem Konto nicht festgelegt. • INVALID_SENDER_ID: Die angegebene Absender-ID ist ungültig. Die Absender-ID hat möglicherweise ein falsches Format oder eine falsche Länge. • INVALID_POOL_ID: Die angegebene Pool-ID ist ungültig. Die Pool-ID hat möglicher weise ein falsches Format oder gehört nicht zu Ihrem Konto. • SENDER_ID_NOT_SUPPORTED_FOR _DESTINATION: Das Zielland unterstützt die Sender-ID nicht. Sie müssen eine Telefonnu mmer oder eine andere Ursprungsidentität für das Senden verwenden. • INVALID_PHONE_NUMBER: Die angegeben e Ursprungstelefonnummer ist ungültig. Die Telefonnummer hat möglicherweise ein falsches Format oder eine falsche Länge.
`iso_country_code`	Das Land , die der Telefonnummer des Empfängers zugeordnet ist, im ISO 3166-1 Alpha-2-Format.
`treatment_id`	Die ID der Nachrichtenbehandlung, wenn die Nachricht in einer A/B-Kampagne gesendet wurde.

Attribut	Beschreibung
`treatment_id`	Wenn die Nachricht mit einer A/B-Testk ampagne gesendet wurde, stellt dieser Wert die Behandlungsnummer der Nachricht dar. Für Transaktions-SMS-Nachrichten ist dieser Wert 0.
`number_of_message_parts`	Die Anzahl der Mitteilungsteile, die Amazon Pinpoint erstellt hat, um die Nachricht zu senden. Im Allgemeinen können SMS-Nachrichten nur 160 GSM-7-Zeichen oder 67 Nicht-GSM -Zeichen enthalten, obwohl diese Limits je nach Land variieren können. Wenn Sie eine Nachricht senden, die diese Limits überschre itet, teilt Amazon Pinpoint die Nachrichten automatisch in kleinere Teile auf. Wir erstellen Rechnungen basierend auf der Anzahl der Mitteilungsteile, die Sie senden.
`message_id`	Die eindeutige ID, die Amazon Pinpoint generiert, wenn die Nachricht akzeptiert wird.
`message_type`	Nachrichtentyp Mögliche Werte sind Promotion al und Transactional. Sie geben diesen Wert beim Erstellen einer Kampagne oder beim Senden von Transaktionsnachrichten mithilfe der Operation SendMessages in der Amazon-Pinpoint-API an.
`campaign_id`	Die eindeutige ID der Amazon-Pinpoint-Ka mpagne, die die Nachricht gesendet hat.

Client

Enthält Informationen zum App-Client, der auf dem Gerät installiert ist, das das Ereignis gemeldet hat.

Attribut	Beschreibung
`client_id`	Bei Ereignissen, die von Apps generiert werden, ist dieser Wert die eindeutige ID des App-Clients, der auf dem Gerät installiert ist. Diese ID wird automatisch vom AWS Mobile SDK for iOS und AWS Mobile SDK for Android generiert. Bei Ereignissen, die generiert werden, wenn Sie Kampagnen und Transaktionsnachrichten senden, entspricht dieser Wert der ID des Endpunkts, an den Sie die Nachricht gesendet haben.
`cognito_id`	Die eindeutige ID, die dem App-Client im Amazon-Cognito-Identitätspool zugewiesen ist, der von Ihrer App genutzt wird

Gerät

Enthält Informationen über das Gerät, das das Ereignis gemeldet hat.

Attribut	Beschreibung
`locale`	Der Gerätestandort
`make`	Der Gerätehersteller, beispielsweise `Apple` oder `Samsung`
`model`	Das Gerätemodell, z. B. `iPhone`
`platform`	Die Geräteplattform, z. B. `ios` oder `android`

Metriken

Enthält Informationen zu Metriken, die dem Ereignis zugeordnet sind.

Attribut	Beschreibung
`price_in_millicents_usd`	Der Betrag, den wir Ihnen für das Senden der Nachricht berechnet haben. Dieser Preis wird in Tausendstel eines US-Cents angegeben. Wenn der Wert dieses Attributs beispielsweise 645 lautet, wird Ihnen für das Senden der Nachricht 0,645 ¢ in Rechnung gestellt (645/1 000 = 0,645 ¢ = 0,00645 $). **Note** Diese Eigenschaft wird nicht für Nachrichten mit dem `event_type` _SMS.BUFFERED angezeigt.

Abfragen von Amazon Pinpoint-Analysedaten

Zusätzlich zur Verwendung der Analyseseiten auf der Amazon-Pinpoint-Konsole können Sie Amazon-Pinpoint-Analytics-APIs verwenden, um Analysedaten für eine Teilmenge von Standardmetriken abzufragen, die Einblicke in Trends zu Benutzerengagement, Kampagnenreichweite usw. bieten. Diese Metriken, auch als Key Performance Indicators (KPIs) bezeichnet, sind messbare Werte, die Ihnen helfen können, die Leistung Ihrer Projekte, Kampagnen und Journeys zu überwachen und zu bewerten.

Wenn Sie die APIs zum Abfragen von Analysedaten verwenden, können Sie die Daten mithilfe des Reporting-Tools Ihrer Wahl analysieren, ohne sich bei der Amazon-Pinpoint-Konsole anmelden oder rohe Ereignisdaten aus Quellen wie Amazon-Pinpoint-Streams analysieren zu müssen. Sie können beispielsweise ein benutzerdefiniertes Dashboard erstellen, das wöchentliche Kampagnenergebnisse anzeigt oder detaillierte Analysen zu den Zustellraten für Ihre Kampagnen bereitstellt.

Sie können die Daten mithilfe der Amazon Pinpoint REST-API, der AWS Command Line Interface (AWS CLI) oder eines AWS SDK abfragen. Um die Daten abzufragen, senden Sie eine Anforderung an die Amazon-Pinpoint-API und verwenden unterstützte Parameter, um die gewünschten Daten und Filter anzugeben, die Sie anwenden möchten. Nachdem Sie Ihre Abfrage gesendet haben, gibt Amazon Pinpoint die Abfrageergebnisse in einer JSON-Antwort zurück. Sie können die Ergebnisse dann an einen anderen Service oder eine Anwendung übergeben – für intensivere Analysen, zum Speichern oder zum Erstellen von Berichten.

Unterstützte Metriken

Amazon Pinpoint bietet programmgesteuerten Zugriff auf Analysedaten für verschiedene Arten von Standardmetriken:

- Anwendungsmetriken: Diese Metriken bieten Einblicke in die Trends für alle Kampagnen und Transaktionsnachrichten, die einem Projekt zugeordnet sind. Sie können eine Anwendungsmetrik beispielsweise verwenden, um eine Aufschlüsselung der Anzahl der Nachrichten abzurufen, die von Empfängern für jede Kampagne geöffnet wurden, die einem Projekt zugeordnet ist. Um auf die Daten für eine Anwendungsmetrik zuzugreifen, verwenden Sie die Ressource Anwendungsmetriken der Amazon-Pinpoint-API.
- Kampagnenmetriken: Diese Metriken geben Aufschluss über die Leistung einzelner Kampagnen. Sie können beispielsweise eine Kampagnenmetrik verwenden, um zu bestimmen, an wie

viele Endpunkte eine Kampagnennachricht gesendet wurde. Um auf die Daten für eine Kampagnenmetrik zuzugreifen, verwenden Sie die Ressource Kampagnenmetriken der Amazon-Pinpoint-API.

- Journey-Engagement-Metriken: Diese Metriken geben einen Einblick in die Leistung der einzelnen Journeys. Beispielsweise können Sie eine Journey-Engagement-Metrikverwenden, um eine Aufschlüsselung der Anzahl der Nachrichten zu erhalten, die von den Teilnehmern an jeder Aktivität einer Journey geöffnet wurden. Um auf die Daten für eine Journey-Engagement-Metrik zuzugreifen, verwenden Sie die Ressource Journey-Engagement-Metriken der Amazon-Pinpoint-API.
- Journey-Ausführungsmetriken: Diese Metriken geben Aufschluss über die Entwicklung der Teilnahme an einzelnen Journeys. Beispielsweise können Sie mit Hilfe einer Journey-Ausführungsmetrik bestimmen, wie viele Teilnehmer die Aktivitäten einer Journey durchlaufen. Um auf die Daten für eine Journey-Ausführungsmetrik zuzugreifen, verwenden Sie die Ressource Journey-Ausführungsmetrik der Amazon-Pinpoint-API.
- Journey-Aktivitätsausführungsmetriken: Diese Metriken geben Aufschluss über die Entwicklung der Beteiligungstrends für einzelne Aktivitäten einer Journey. Beispielsweise können Sie mit Hilfe einer Journey-Aktivitätsausführungsmetriken bestimmen, wie viele Teilnehmer eine Aktivität abgeschlossen haben. Um auf die Daten für eine Journey-Aktivitätsausführungsmetrik zuzugreifen, verwenden Sie die Ressource Journey-Aktivitätsausführungsmetriken der Amazon-Pinpoint-API.

Eine vollständige Liste der Standardmetriken, die Sie programmgesteuert abfragen können, finden Sie unter Standardmetriken.

Amazon Pinpoint sammelt und kumuliert automatisch Daten für alle unterstützten Metriken, für alle Ihre Projekte, Kampagnen und Reisen. Zudem werden die Daten kontinuierlich aktualisiert. Der resultierende Datenlatenz-Zeitrahmen ist auf ungefähr zwei Stunden beschränkt. Beachten Sie jedoch, dass es bei bestimmten Metriken zu Datenverzögerungen kommen kann. Dies liegt daran, dass die Daten für einige Metriken auf Informationen basieren, die wir von den E-Mail-Anbietern der Empfänger erhalten. Einige Anbieter senden uns diese Informationen sofort, während andere sie seltener senden.

Amazon Pinpoint speichert die Daten 90 Tage lang. Um die Daten für mehr als 90 Tage zu speichern oder in Echtzeit auf Rohdaten zuzugreifen, können Sie ein Amazon Pinpoint Pinpoint-Projekt so konfigurieren, dass Ereignisdaten an Amazon Kinesis Data Streams oder Amazon Data Firehose gestreamt werden. Weitere Informationen zum Konfigurieren von Ereignisströmen finden Sie unter Streamen von Amazon-Pinpoint-Ereignissen zu Kinesis.

Abfragegrundlagen

Um die Daten für eine Metrik abzufragen, senden Sie eine `get`-Anfrage an die entsprechende Metrik-Ressource der Amazon Pinpoint-API. In Ihrer Anforderung definieren Sie Ihre Abfrage, indem Sie unterstützte Parameter für die folgenden Abfragekomponenten verwenden:

- Projekt: Geben Sie ein Projekt an, indem Sie die Projekt-ID als Wert für den `application-id`-Parameter angeben. Dieser Parameter ist für alle Metriken erforderlich.
- Kampagne: Geben Sie eine Kampagne an, indem Sie die Kampagnen-ID als Wert für den `campaign-id`-Parameter angeben. Dieser Parameter ist nur für Kampagnenmetriken erforderlich.
- Journey: Geben Sie eine Journey an, indem Sie die Journey-ID als Wert für den `journey-id`-Parameter angeben. Dieser Parameter wird nur für Journey-Engagement-Metriken und Journey-Ausführungsmetriken sowie für Journey-Aktivitätsausführungsmetrik benötigt.
- Journey-Aktivität: Geben Sie eine Journey-Aktivität an, indem Sie die ID der Journey-Aktivität als Wert für den `journey-activity-id`-Parameter angeben. Dieser Parameter wird nur für Journey-Aktivitätsausführungsmetriken benötigt.
- Datumsbereich: Um die Daten optional nach Datumsbereich zu filtern, geben Sie mithilfe der unterstützten Start- und Endzeitparameter das erste und letzte Datum und die Uhrzeit des Datumsbereichs an. Die Werte sollten im erweiterten ISO 8601-Format vorliegen und die koordinierte Weltzeit (UTC) verwenden, z. B. `2019-07-19T20:00:00Z` für 20.00 Uhr UTC am 19. Juli 2019.

  Datumsbereiche werden inklusiv angegeben und dürfen maximal 31 Kalendertage umfassen. Darüber hinaus müssen das erste Datum und die erste Uhrzeit früher als 90 Tage ab dem aktuellen Tag liegen. Wenn Sie keinen Datumsbereich angeben, werden die Daten für die letzten 31 Kalendertage von Amazon Pinpoint zurückgegeben. Datumsbereichsparameter werden von allen Metriken unterstützt, mit Ausnahme von Journey-Aausführungsmetriken und Journey-Aktivitätsausführungsmetriken.
- Metrik: Geben Sie die Metrik an, indem Sie den Namen der Metrik als Wert für den `kpi-name`-Parameter angeben. Dieser Wert beschreibt die zugeordnete Metrik und besteht aus zwei oder mehr Begriffen, die aus alphanumerischen Kleinbuchstaben bestehen, die durch einen Bindestrich getrennt sind. Beispiele sind `email-open-rate` und `successful-delivery-rate`. Dieser Parameter wird für alle Metriken mit Ausnahme von Journey-Aausführungsmetriken und Journey-Aktivitätsausführungsmetriken benötigt. Eine vollständige Liste der unterstützten Metriken und dem jeweils zu verwendenden `kpi-name`-Wert finden Sie unter Standardmetriken.

Nachdem Sie Ihre Abfrage gesendet haben, gibt Amazon Pinpoint die Abfrageergebnisse in einer JSON-Antwort zurück. In der Antwort variiert die Struktur der Ergebnisse in Abhängigkeit von der Metrik, die Sie abgefragt haben.

Einige Metriken liefern nur einen Wert, z. B. die Anzahl der Nachrichten, die im Rahmen einer Kampagne zugestellt wurden. Andere Metriken liefern mehrere Werte und gruppieren diese in der Regel nach einem relevanten Feld, z. B. der Anzahl der Nachrichten, die bei jedem Lauf einer Kampagne zugestellt wurden, gruppiert nach Kampagnenlauf. Wenn eine Metrik mehrere Werte bereitstellt und gruppiert, enthält die JSON-Antwort ein Feld, das angibt, welches Feld zur Gruppierung der Daten verwendet wurde. Weitere Informationen zur Struktur von Abfrageergebnissen finden Sie unter Verwenden von Abfrageergebnissen.

IAM-Richtlinien zum Abfragen von Amazon-Pinpoint-Analysedaten

Durch die Verwendung der Amazon-Pinpoint-API können Sie Analysedaten für eine Teilmenge von Standardmetriken abfragen, die auch als Key Performance Indicators (KPIs) bezeichnet werden und für Amazon Pinpoint-Projekte, -Kampagnen und -Journeys gelten. Diese Metriken können Ihnen helfen, die Leistung von Projekten, Kampagnen und Journey zu überwachen und zu bewerten.

Um den Zugriff auf diese Daten zu verwalten, können Sie AWS Identity and Access Management (IAM)-Richtlinien erstellen, die Berechtigungen für IAM-Rollen oder -Benutzer definieren, die zum Zugriff auf die Daten berechtigt sind. Um eine detaillierte Kontrolle des Zugriffs auf diese Daten zu unterstützen, bietet Amazon Pinpoint mehrere verschiedene Aktionen, die Sie in IAM-Richtlinien angeben können. Es gibt eine eindeutige Aktion zum Anzeigen von Analysedaten auf der Amazon-Pinpoint-Konsole (`mobiletargeting:GetReports`) und es gibt andere Aktionen für den programmatischen Zugriff auf Analysedaten mithilfe der Amazon-Pinpoint-API.

Um IAM-Richtlinien zu erstellen, die den Zugriff auf Analysedaten verwalten, können Sie die AWS Management Console, die AWS CLI oder die IAM-API verwenden. Beachten Sie, dass die Registerkarte Visueller Editor auf der AWS Management Console derzeit keine Aktionen zum Anzeigen oder Abfragen von Amazon-Pinpoint-Analysedaten enthält. Sie können jedoch die erforderlichen Aktionen manuell zu IAM-Richtlinien hinzufügen, indem Sie die Registerkarte JSON auf der Konsole verwenden.

Die folgende Richtlinie ermöglicht beispielsweise den programmgesteuerten Zugriff auf alle Analysedaten für alle Ihre Projekte, Kampagnen und Journeys in allen AWS Regionen:

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

Dabei steht *accountId* für Ihre AWS-Konto-ID.

Als bewährte Methode sollten Sie jedoch Richtlinien erstellen, die dem Prinzip der geringsten Rechte folgen. Mit anderen Worten, Sie sollten Richtlinien erstellen, die nur die Berechtigungen enthalten, die zum Ausführen einer bestimmten Aufgabe erforderlich sind. Um diese Praxis zu unterstützen und eine detailliertere Steuerung zu implementieren, können Sie den programmgesteuerten Zugriff auf die Analysedaten für nur ein bestimmtes Projekt in einer bestimmten AWS Region beschränken. Zum Beispiel:

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

Wobei gilt:

- *region* ist der Name der AWS-Region, in der das Projekt gehostet wird.
- *accountId* ist Ihre AWS-Konto-ID.
- *projectId* ist der Bezeichner für das Projekt, auf das Sie Zugriff gewähren möchten.

Ebenso ermöglicht die folgende Beispielrichtlinie den programmatischen Zugriff auf die Analysedaten nur für eine bestimmte Kampagne:

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

Wobei gilt:

- *region* ist der Name der AWS-Region, in der das Projekt gehostet wird.
- *accountId* ist Ihre AWS-Konto-ID.
- *projectId* ist der Bezeichner für das Projekt, das der Kampagne zugeordnet ist.
- *campaignId* ist die Kennung für die Kampagne, auf die Sie Zugriff gewähren möchten.

Und die folgende Beispielrichtlinie ermöglicht den programmgesteuerten Zugriff auf alle Analysedaten, sowohl Engagement- als auch Ausführungsdaten, für eine bestimmte Journey und die Aktivitäten, die diese Journey umfassen:

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

Wobei gilt:

- *region* ist der Name der AWS-Region, in der das Projekt gehostet wird.
- *accountId* ist Ihre AWS-Konto-ID.
- *projectId* ist die ID für das Projekt, das der Journey zugeordnet ist.
- *journeyId* ist die ID für die Journey, zu der Sie Zugang gewähren möchten.

Eine vollständige Liste der Amazon-Pinpoint-API-Aktionen, die Sie in IAM-Richtlinien verwenden können, finden Sie unter Amazon-Pinpoint-Aktionen für IAM-Richtlinien. Ausführliche Informationen zum Erstellen und Verwalten von IAM-Richtlinien finden Sie im IAM-Benutzerhandbuch.

Standardmetriken von Amazon Pinpoint Analytics

Sie können Amazon-Pinpoint-Analytics-APIs verwenden, um Analysedaten für eine Teilmenge von Standardmetriken abzufragen, die für Amazon-Pinpoint-Projekte, -Kampagnen und -Journeys gelten. Diese Metriken, auch als Key Performance Indicators (KPIs) bezeichnet, sind messbare Werte, die Ihnen helfen können, die Leistung von Projekten, Kampagnen und Journeys zu überwachen und zu bewerten.

Amazon Pinpoint bietet programmgesteuerten Zugriff auf Analysedaten für verschiedene Arten von Standardmetriken:

- Anwendungsmetriken: Diese Metriken bieten Einblicke in die Trends für alle Kampagnen und Transaktionsnachrichten, die einem Projekt zugeordnet sind, auch als Anwendung bezeichnet. Sie können eine Anwendungsmetrik beispielsweise verwenden, um eine Aufschlüsselung der Anzahl der Nachrichten abzurufen, die von Empfängern für jede Kampagne geöffnet wurden, die einem Projekt zugeordnet ist.
- Kampagnenmetriken: Diese Metriken geben Aufschluss über die Leistung einzelner Kampagnen. Sie können eine Kampagnenmetrik beispielsweise verwenden, um die Anzahl der Endpunkte zu bestimmen, an die eine Kampagnennachricht gesendet wurde, oder die Anzahl dieser Nachrichten, die an Endpunkte übermittelt wurden.
- Journey-Engagement-Metriken: Diese Metriken geben einen Einblick in die Leistung der einzelnen Journeys. Beispielsweise können Sie eine Journey-Engagement-Metrikverwenden, um eine Aufschlüsselung der Anzahl der Nachrichten zu erhalten, die von den Teilnehmern an jeder Aktivität einer Journey geöffnet wurden.
- Journey-Ausführungsmetriken: Diese Metriken geben Aufschluss über die Entwicklung der Teilnahme an einzelnen Journeys. Beispielsweise können Sie mit Hilfe einer Journey-Ausführungsmetrik bestimmen, wie viele Teilnehmer eine Journey begonnen haben.
- Journey-Aktivitätsausführungsmetriken: Diese Metriken geben Aufschluss über die Entwicklung der Beteiligungstrends für einzelne Aktivitäten einer Journey. Beispielsweise können Sie mit einer Journey-Aktivitätsusführungsmetrik bestimmen, wie viele Teilnehmer eine Aktivität gestartet haben und wie viele Teilnehmer jeden Pfad in einer Aktivität abgeschlossen haben.

In den Themen in diesem Abschnitt sind die einzelnen Metriken aufgeführt und beschrieben, die Sie für jeden Metriktyp abfragen können.

Themen

Anwendungsmetriken für Kampagnen

In der folgenden Tabelle werden Standardanwendungsmetriken aufgeführt und beschrieben, die Sie abfragen können, um die Leistung aller Kampagnen zu bewerten, die einem Amazon-Pinpoint-Projekt zugeordnet sind. Um Daten für diese Metriken abzufragen, verwenden Sie die Ressource Anwendungsmetriken der Amazon-Pinpoint-API. Die Spalte kpi-name in der Tabelle gibt den Wert an, der für den `kpi-name`-Parameter in einer Abfrage verwendet werden soll.

Kennzahl	Kpi-name	Beschreibung
Delivery rate (Zustellungsrate)	`successful-delivery-rate`	Prozentsatz der Nachrichten, die Empfängern zugestellt wurden, für alle Kampagnen, die einem Projekt zugeordnet sind. Diese Metrik wird als Anzahl der Nachrichten berechnet , die von allen Kampagnen für ein Projekt gesendet und Empfängern zugestell t wurden, dividiert durch die

Kennzahl	Kpi-name	Beschreibung
		Anzahl der Nachrichten, die von all diesen Kampagnen gesendet wurden.
Zustellungsrate, gruppiert nach Datum	`successful-delivery-rate-grouped-by-date`	Prozentsatz der Nachrichten, die an jedem Tag im angegebenen Zeitraum einem Empfänger zugestellt wurden, für alle Kampagnen, die einem Projekt zugeordnet sind. Diese Metrik wird als Anzahl der Nachrichten gesendet, die von allen Kampagnen für ein Projekt gesendet und Empfängern zugestellt wurden, dividiert durch die Anzahl der Nachrichten, die von all diesen Kampagnen an jedem Tag im angegebenen Zeitraum gesendet wurden. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Email open rate (E-Mail-Öffnungsrate)	`email-open-rate`	Prozentsatz der E-Mail-Nachrichten, die von Empfängern geöffnet wurden, für alle Kampagnen, die einem Projekt zugeordnet sind. Diese Metrik wird als Anzahl der E-Mail-Nachrichten berechnet, die von allen Kampagnen für ein Projekt gesendet und von Empfängern geöffnet wurden, dividiert durch die Anzahl der E-Mail-Nachrichten, die von all diesen Kampagnen gesendet und Empfängern zugestellt wurden.

Kennzahl	Kpi-name	Beschreibung
E-Mail-Öffnungsrate, gruppiert nach Kampagne	`email-open-rate-grouped-by-campaign`	Prozentsatz der E-Mail-Nachrichten, die vom Empfänger geöffnet wurden, für jede Kampagne, die einem Projekt zugeordnet ist. Diese Metrik wird als Anzahl der E-Mail-Nachrichten berechnet, die von einer Kampagne gesendet und von Empfängern geöffnet wurden, dividiert durch die Anzahl der E-Mail-Nachrichten, die von der Kampagne gesendet und Empfängern zugestellt wurden. Die Abfrageergebnisse für diese Metrik werden nach der Kampagnen-ID (`CampaignId`) gruppiert, einer Zeichenfolge, die eine Kampagne eindeutig identifiziert.
Endpunktzustellungen	`unique-deliveries`	Anzahl der eindeutigen Endpunkte, an die Nachrichten gesendet wurden, für alle Kampagnen, die einem Projekt zugeordnet sind.

Kennzahl	Kpi-name	Beschreibung
Endpunktzustellungen, gruppiert nach Kampagne	`unique-deliveries-grouped-by-campaign`	Anzahl der eindeutigen Endpunkte, an die Nachrichten gesendet wurden, für jede Kampagne, die einem Projekt zugeordnet ist. Die Abfrageergebnisse für diese Metrik werden nach der Kampagnen-ID (`CampaignId`) gruppiert, einer Zeichenfolge, die eine Kampagne eindeutig identifiziert.
Endpunktzustellungen, gruppiert nach Datum	`unique-deliveries-grouped-by-date`	Anzahl der eindeutigen Endpunkte, an die an jedem Tag im angegebenen Zeitraum Nachrichten gesendet wurden, für alle Kampagnen, die einem Projekt zugeordnet sind. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Zugestellte Nachrichten, gruppiert nach Kampagne	`successful-deliveries-grouped-by-campaign`	Anzahl der Nachrichten, die Empfängern zugestellt wurden, für jede Kampagne, die einem Projekt zugeordnet ist. Diese Metrik wird als Anzahl der Nachrichten berechnet, die von einer Kampagne gesendet wurden, abzüglich der Anzahl der Nachrichten, die von der Kampagne gesendet wurden und den Empfängern aufgrund permanenter Unzustellbarkeit nicht zugestellt werden konnten. Die Abfrageergebnisse für diese Metrik werden nach der Kampagnen-ID (`CampaignId`) gruppiert, einer Zeichenfolge, die eine Kampagne eindeutig identifiziert.

Kennzahl	Kpi-name	Beschreibung
Push open rate (Öffnungsrate nach Push)	`push-open-rate`	Prozentsatz der Push-Benachrichtigungen, die von Empfängern geöffnet wurden, für alle Kampagnen, die einem Projekt zugeordnet sind. Diese Metrik wird als Anzahl der von allen Kampagnen für ein Projekt gesendeten und von Empfängern geöffneten Push-Benachrichtigungen berechnet, dividiert durch die Anzahl der Push-Benachrichtigungen, die von all diesen Kampagnen gesendet und Empfängern zugestellt wurden.

Kennzahl	Kpi-name	Beschreibung
Push-Öffnungsrate, gruppiert nach Kampagne	`push-open-rate-grouped-by-campaign`	Prozentsatz der Push-Benachrichtigungen, die von Empfängern geöffnet wurden, für jede Kampagne, die einem Projekt zugeordnet ist. Diese Metrik wird als die Anzahl der Push-Benachrichtigungen berechnet, die von einer Kampagne gesendet und von Empfängern geöffnet wurden, dividiert durch die Anzahl der Push-Benachrichtigungen, die von der Kampagne gesendet und Empfängern zugestellt wurden. Die Abfrageergebnisse für diese Metrik werden nach der Kampagnen-ID (`CampaignId`) gruppiert, einer Zeichenfolge, die eine Kampagne eindeutig identifiziert.

Anwendungsmetriken für transaktionsbezogene E-Mail-Nachrichten

In der folgenden Tabelle werden Standardanwendungsmetriken aufgeführt und beschrieben, die Sie abfragen können, um Trends für alle Transaktions-E-Mail-Nachrichten zu überwachen, die einem Amazon-Pinpoint-Projekt zugeordnet sind. Um Daten für diese Metriken abzufragen, verwenden Sie die Ressource Anwendungsmetriken der Amazon-Pinpoint-API. Die Spalte kpi-name in der Tabelle gibt den Wert an, der für den `kpi-name`-Parameter in einer Abfrage verwendet werden soll.

Beachten Sie, dass diese Metriken keine Daten zu E-Mail-Nachrichten enthalten, die von Kampagnen gesendet wurden. Sie stellen lediglich Daten zu transaktionsbezogenen E-Mail-Nachrichten bereit.

Um Daten nach Nachrichten zu durchsuchen, die von einer oder mehreren Kampagnen gesendet wurden, verwenden Sie eine Kampagnenmetrik oder eine Anwendungsmetrik für Kampagnen.

Kennzahl	Kpi-name	Beschreibung
Clicks (Klickvorgänge)	`txn-emails-clicked`	Anzahl der Klicks von Empfängern auf die Links in den Nachrichten. Wenn ein Empfänger auf mehrere Links in einer Nachricht oder mehrmals auf denselben Link geklickt hat, wird jeder dieser Klickvorgänge gezählt.
Klicks, gruppiert nach Datum	`txn-emails-clicked-grouped-by-date`	Die Anzahl der Klickvorgänge von Empfängern auf Links in den Nachrichten für jeden Tag im angegebenen Zeitraum. Wenn ein Empfänger auf mehrere Links in einer Nachricht oder mehrmals auf denselben Link geklickt hat, wird jeder dieser Klickvorg änge gezählt. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.
Complaint rate (Beschwer derate)	`txn-emails-complai nt-rate`	Der Prozentsatz der Nachricht en, die von den Empfänger n als unerwünschte E-Mails gemeldet wurden. Diese Metrik wird berechnet als die Anzahl der Nachricht en, die von Empfängern

Kennzahl	Kpi-name	Beschreibung
		als unerwünschte E-Mails gemeldet wurden, geteilt durch die Anzahl der gesendeten Nachrichten.
Beschwerderate, gruppiert nach Datum	`txn-emails-complaint-rate-grouped-by-date`	Der Prozentsatz der Nachrichten, die von den Empfängern als unerwünschte E-Mails gemeldet wurden, für jeden Tag im angegebenen Zeitraum. Diese Metrik wird berechnet als die Anzahl der Nachrichten, die von Empfängern als unerwünschte E-Mails gemeldet wurden, geteilt durch die Anzahl der gesendeten Nachrichten für jeden Tag im angegebenen Zeitraum. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.
Complaints (Beschwerden)	`txn-emails-with-complaints`	Die Anzahl der Nachrichten, die von Empfängern als unerwünschte E-Mails gemeldet wurden.

Kennzahl	Kpi-name	Beschreibung
Beschwerden, gruppiert nach Datum	`txn-emails-with-complaints-grouped-by-date`	Die Anzahl der Nachrichten, die von Empfängern als unerwünschte E-Mails gemeldet wurden, für jeden Tag im angegebenen Zeitraum. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.
Deliveries (Zustellungen)	`txn-emails-delivered`	Die Anzahl der Nachrichten, die Empfängern zugestellt wurden. Diese Metrik wird als die Anzahl der gesendeten Nachrichten berechnet, abzüglich der Anzahl der Nachrichten, die aufgrund einer temporären oder permanenten Unzustellbarkeit oder aufgrund einer Ablehnung nicht zugestellt werden konnten. Eine Nachricht wird zurückgewiesen, wenn Amazon Pinpoint feststellt, dass sie Malware enthält. Amazon Pinpoint versucht nicht, abgelehnte Nachrichten zu senden.

Kennzahl	Kpi-name	Beschreibung
Zustellungen, gruppiert nach Datum	`txn-emails-delivered-grouped-by-date`	Die Anzahl der Nachrichten, die an Empfänger zugestellt wurden, für jeden Tag im angegebenen Zeitraum. Diese Metrik wird berechnet als die Anzahl der gesendeten Nachrichten, abzüglich der Anzahl der Nachrichten, die aufgrund einer temporären oder permanenten Unzustellbarkeit oder einer Ablehnung nicht zugestellt werden konnten (für jeden Tag im angegebenen Zeitraum). Eine Nachricht wird zurückgewiesen, wenn Amazon Pinpoint feststellt, dass sie Malware enthält. Amazon Pinpoint versucht nicht, abgelehnte Nachrichten zu senden. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Delivery rate (Zustellungsrate)	`txn-emails-delivery-rate`	Der Prozentsatz der Nachrichten, die Empfänger zugestellt wurden. Diese Metrik wird berechnet als die Anzahl der Nachrichten, die gesendet und Empfängern zugestellt wurden, dividiert durch die Anzahl der gesendeten Nachrichten.
Zustellungsrate, gruppiert nach Datum	`txn-emails-delivery-rate-grouped-by-date`	Der Prozentsatz der Nachrichten, die an Empfänger zugestellt wurden, für jeden Tag im angegebenen Zeitraum. Diese Metrik wird berechnet als die Anzahl der Nachrichten, die an Empfänger gesendet und zugestellt wurden, geteilt durch die Anzahl der gesendeten Nachrichten, für jeden Tag im angegebenen Zeitraum. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Hard bounces (Permanente Unzustellbarkeiten)	`txn-emails-hard-bounced`	Die Anzahl der Nachrichten, die aufgrund einer permanenten Unzustellbarkeit nicht an Empfänger übermittelt werden konnten. Eine permanente Unzustellbarkeit tritt auf, wenn ein anhaltendes Problem die Zustellung einer Nachricht verhindert, z. B. wenn die E-Mail-Adresse eines Empfängers nicht existiert.
Permanente Unzustellbarkeiten, gruppiert nach Datum	`txn-emails-hard-bounced-grouped-by-date`	Die Anzahl der Nachrichten, die aufgrund einer permanenten Unzustellbarkeit nicht an die Empfänger zugestellt werden konnten (für jeden Tag im angegebenen Zeitraum). Eine permanente Unzustellbarkeit tritt auf, wenn ein anhaltendes Problem die Zustellung einer Nachricht verhindert, z. B. wenn die E-Mail-Adresse eines Empfängers nicht existiert. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.
Opens (Öffnungsvorgänge)	`txn-emails-opened`	Die Anzahl der Nachrichten, die von Empfängern geöffnet wurden.

Kennzahl	Kpi-name	Beschreibung
Öffnungsvorgänge, gruppiert nach Datum	`txn-emails-opened-grouped-by-date`	Die Anzahl der Nachrichten, die von Empfängern geöffnet wurde (für jeden Tag im angegebenen Zeitraum). Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.
Sends (Sendevorgänge)	`txn-emails-sent`	Anzahl der gesendeten Nachrichten.
Sendevorgänge, gruppiert nach Datum	`txn-emails-sent-grouped-by-date`	Die Anzahl der gesendeten Nachrichten (für jeden Tag im angegebenen Zeitraum). Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.
Temporäre Unzustellbarkeiten	`txn-emails-soft-bounced`	Die Anzahl der Nachrichten, die aufgrund einer temporären Unzustellbarkeit nicht an Empfänger zugestellt werden konnten. Eine temporäre Unzustellbarkeit tritt auf, wenn ein vorübergehendes Problem die Zustellung einer Nachricht verhindert, z. B. wenn der Posteingang eines Empfängers voll ist oder wenn der empfangende Server vorübergehend nicht verfügbar ist.

Kennzahl	Kpi-name	Beschreibung
Temporäre Unzustellbarkeiten, gruppiert nach Datum	`txn-emails-soft-bounced-grouped-by-date`	Die Anzahl der Nachrichten, die aufgrund einer temporären Unzustellbarkeit nicht an Empfänger zugestellt werden konnten (für jeden Tag im angegebenen Zeitraum). Eine temporäre Unzustellbarkeit tritt auf, wenn ein vorübergehendes Problem die Zustellung einer Nachricht verhindert, z. B. wenn der Posteingang eines Empfängers voll ist oder wenn der empfangende Server vorübergehend nicht verfügbar ist. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Eindeutige Benutzerklickereignisse	`txn-emails-unique-clicks`	Die Anzahl der eindeutigen Empfänger (Endpunkte), die auf Links in Nachrichten geklickt haben. Im Gegensatz zur Metrik Clicks (Klickvorgänge) meldet diese Metrik die Anzahl der eindeutigen Empfänger, die auf Links geklickt haben, und nicht die Anzahl der aufgetretenen Klickereignisse. Wenn beispielsweise ein einzelner Empfänger auf mehrere Links in derselben Nachricht oder auf denselben Link mehr als einmal geklickt hat, meldet diese Metrik nur ein Klickereignis für diesen Empfänger.

Kennzahl	Kpi-name	Beschreibung
Eindeutige Benutzerklickereignisse, gruppiert nach Datum	`txn-emails-unique-clicks-grouped-by-date`	Die Anzahl der eindeutigen Empfänger (Endpunkte), die für jeden Tag im angegebenen Zeitraum auf Links in Nachrichten geklickt haben. Im Gegensatz zur Metrik für Clicks, grouped by date (Klickereignisse, gruppiert nach Datum) meldet diese Metrik die Anzahl der eindeutigen Empfänger, die auf Links geklickt haben, und nicht die Anzahl der aufgetretenen Klickereignisse. Wenn beispielsweise ein einzelner Empfänger auf mehrere Links in derselben Nachricht oder auf denselben Link mehr als einmal geklickt hat, meldet diese Metrik nur ein Klickereignis für diesen Empfänger. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Eindeutige Benutzeröffnungsvorgänge	`txn-emails-unique-opens`	Die Anzahl der eindeutigen Empfänger (Endpunkte), die Nachrichten geöffnet haben. Anders als die Metrik Opens (Öffnungsvorgänge) meldet diese Metrik die Anzahl der eindeutigen Empfänger, die Nachrichten geöffnet haben, und nicht die Anzahl der aufgetretenen Öffnungsvorgänge. Wenn beispielsweise ein einzelner Empfänger dieselbe Nachricht mehrmals öffnet, meldet diese Metrik nur einen Öffnungsvorgang für diesen Empfänger.

Kennzahl	Kpi-name	Beschreibung
Eindeutige Öffnungsvorgänge, gruppiert nach Datum	`txn-emails-unique-opens-grouped-by-date`	Die Anzahl der eindeutigen Empfänger (Endpunkte), die Nachrichten geöffnet haben (für jeden Tag im angegeben en Zeitraum). Im Gegensatz zur Metrik Opens, grouped by date (Öffnungsvorgänge, gruppiert nach Datum) meldet diese Metrik die Anzahl der eindeutigen Empfänger, die Nachrichten geöffnet haben, und nicht die Anzahl der aufgetretenen Öffnungsv orgänge. Wenn beispiels weise ein einzelner Empfänger dieselbe Nachricht mehrmals öffnet, meldet diese Metrik nur einen Öffnungsvorgang für diesen Empfänger. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Anwendungsmetriken für transaktionsbezogene SMS-Nachrichten

In der folgenden Tabelle werden Standardanwendungsmetriken aufgeführt und beschrieben, die Sie abfragen können, um Trends für alle Transaktions-SMS-Nachrichten zu überwachen, die einem Amazon-Pinpoint-Projekt zugeordnet sind. Um Daten für diese Metriken abzufragen, verwenden Sie die Ressource Anwendungsmetriken der Amazon-Pinpoint-API. Die Spalte kpi-name in der Tabelle gibt den Wert an, der für den `kpi-name`-Parameter in einer Abfrage verwendet werden soll.

Beachten Sie, dass diese Metriken keine Daten über SMS-Nachrichten liefern, die von Kampagnen gesendet wurden. Sie stellen lediglich Daten zu transaktionsnezogenen SMS-Nachrichten bereit. Um Daten nach Nachrichten zu durchsuchen, die von einer oder mehreren Kampagnen gesendet wurden, verwenden Sie eine Kampagnenmetrik oder eine Anwendungsmetrik für Kampagnen.

Kennzahl	Kpi-name	Beschreibung
Durchschnittspreis pro Nachricht, gruppiert nach Land	`txn-sms-average-price-grouped-by-country`	Die durchschnittlichen Kosten für das Senden jeder Nachricht für jedes Land oder jede Region, an die Nachrichten gesendet wurden. Der Preis wird in Tausendstel eines US-Cents angegeben. Wenn der Wert dieses Attributs beispiels weise 645 lautet, wird Ihnen für das Senden der Nachricht 0,645 ¢ in Rechnung gestellt (645/1 000 = 0,645 ¢ = 0,00645 $). Diese Metrik wird als Gesamtkosten aller Nachricht en berechnet, die an Empfänger in jedem Land oder Region gesendet wurden, geteilt durch die Anzahl der Nachrichten, die an Empfänger in jedem dieser Länder und Regionen gesendet wurden. Die Abfrageergebnisse für diese Metrik werden im Format ISO 3166-1 alpha-2

Kennzahl	Kpi-name	Beschreibung
		nach Land oder Region gruppiert.

Kennzahl	Kpi-name	Beschreibung
Durchschnittspreis pro Nachrichtenteil, gruppiert nach Land	`txn-sms-average-price-by-parts-grouped-by-country`	Die durchschnittlichen Kosten für das Senden jedes Nachrichtenteils für jedes Land oder jede Region, an die Nachrichten gesendet wurden. Ein Nachrichtenteil ist ein Teil einer SMS-Nachricht. Der Preis wird in Tausendstel eines US-Cents angegeben. Wenn der Wert dieses Attributs beispielsweise 645 lautet, wird Ihnen für das Senden der Nachricht 0,645 ¢ in Rechnung gestellt (645/1 000 = 0,645 ¢ = 0,00645 $). Diese Metrik wird als Gesamtkosten aller Nachrichtenteile berechnet, die an Empfänger in jedem Land oder Region gesendet wurden, geteilt durch die Anzahl der Nachrichtenteile, die an Empfänger in jedem dieser Länder und Regionen gesendet wurden. Die Abfrageergebnisse für diese Metrik werden im Format ISO 3166-1 alpha-2 nach Land oder Region gruppiert.

Kennzahl	Kpi-name	Beschreibung
Deliveries (Zustellungen)	`txn-sms-delivered`	Die Anzahl der Nachrichten, die Empfängern zugestellt wurden.
Zustellungen, gruppiert nach Land	`txn-sms-delivered-grouped-by-country`	Die Anzahl der Nachrichten, die an Empfänger zugestell t wurden, für jedes Land oder jede Region, an die die Nachrichten gesendet wurden. Die Abfrageergebnisse für diese Metrik werden im Format ISO 3166-1 alpha-2 nach Land oder Region gruppiert.
Zustellungen, gruppiert nach Datum	`txn-sms-delivered-grouped-by-date`	Die Anzahl der Nachrichten, die an Empfänger zugestell t wurden, für jeden Tag im angegebenen Zeitraum. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.
Zustellungsfehler	`txn-sms-error-dist ribution`	Die Anzahl der Fehler beim Versuch, die Nachrichten zu senden, für jede Art von aufgetretenem Fehler. Die Abfrageergebnisse für diese Metrik werden nach Fehlercode für jede aufgetret ene Fehlerart gruppiert.

Kennzahl	Kpi-name	Beschreibung
Delivery rate (Zustellungsrate)	`txn-sms-delivery-rate`	Der Prozentsatz der Nachrichten, die Empfänger zugestellt wurden. Diese Metrik wird berechnet als die Anzahl der Nachrichten, die gesendet und Empfängern zugestellt wurden, dividiert durch die Anzahl der gesendeten Nachrichten.
Zustellungsrate, gruppiert nach Datum	`txn-sms-delivery-rate-grouped-by-date`	Der Prozentsatz der Nachrichten, die an Empfänger zugestellt wurden, für jeden Tag im angegebenen Zeitraum. Diese Metrik wird berechnet als die Anzahl der Nachrichten, die an Empfänger gesendet und zugestellt wurden, geteilt durch die Anzahl der gesendeten Nachrichten, für jeden Tag im angegebenen Zeitraum. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Zugestellte Nachrichtenteile	`txn-sms-delivered-by-parts`	Die Anzahl der Nachricht enteile, die an Teilnehme r zugestellt wurden. Ein Nachrichtenteil ist ein Teil einer SMS-Nachricht. Wenn eine SMS-Nachricht mehr Zeichen enthält, als das SMS-Protokoll zulässt, teilt Amazon Pinpoint die Nachricht in so viele Nachrichtenteile auf, wie erforderlich sind, um die Nachricht an einen Empfänger zu senden.
Zugestellte Nachrichtenteile, gruppiert nach Ländern	`txn-sms-delivered-by-parts-grouped-by-country`	Die Anzahl der Nachricht enteile, die an Empfänger zugestellt wurden, für jedes Land oder jede Region, an die die Nachrichtenteile gesendet wurden. Ein Nachrichtenteil ist ein Teil einer SMS-Nachricht. Die Abfrageergebnisse für diese Metrik werden im Format ISO 3166-1 alpha-2 nach Land oder Region gruppiert.

Kennzahl	Kpi-name	Beschreibung
Gesendete Nachrichtenteile	`txn-sms-sent-by-parts`	Die Anzahl der gesendeten Nachrichtenteile. Ein Nachrichtenteil ist ein Teil einer SMS-Nachricht. Wenn eine SMS-Nachricht mehr Zeichen enthält, als das SMS-Protokoll zulässt, teilt Amazon Pinpoint die Nachricht in so viele Nachrichtenteile auf, wie erforderlich sind, um die Nachricht an einen Empfänger zu senden.
Gesendete Nachrichtenteile, gruppiert nach Ländern	`txn-sms-sent-by-parts-grouped-by-country`	Die Anzahl der gesendeten Nachrichtenteile für jedes Land oder jede Region, an die Nachrichten gesendet wurden. Ein Nachrichtenteil ist ein Teil einer SMS-Nachricht. Die Abfrageergebnisse für diese Metrik werden im Format ISO 3166-1 alpha-2 nach Land oder Region gruppiert.
Messages sent (Gesendete Nachrichten)	`txn-sms-sent`	Anzahl der gesendeten Nachrichten.

Kennzahl	Kpi-name	Beschreibung
Gesendete Nachrichten, gruppiert nach Ländern	`txn-sms-sent-grouped-by-country`	Die Anzahl der gesendeten Nachrichten für jedes Land oder jede Region, an die Nachrichten gesendet wurden. Die Abfrageergebnisse für diese Metrik werden im Format ISO 3166-1 alpha-2 nach Land oder Region gruppiert.
Gesendete Nachrichten, gruppiert nach Datum	`txn-sms-sent-grouped-by-date`	Die Anzahl der gesendeten Nachrichten (für jeden Tag im angegebenen Zeitraum). Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Gesamtpreis, gruppiert nach Land	`txn-sms-total-price-grouped-by-country`	Die Gesamtkosten für das Senden der Nachrichten für jedes Land oder jede Region, an die Nachrichten gesendet wurden. Der Preis wird in Tausendstel eines US-Cents angegeben. Wenn der Wert dieses Attributs beispiels weise 645 lautet, wird Ihnen für das Senden der Nachricht 0,645 ¢ in Rechnung gestellt (645/1 000 = 0,645 ¢ = 0,00645 $). Die Abfrageergebnisse für diese Metrik werden im Format ISO 3166-1 alpha-2 nach Land oder Region gruppiert.

Kampagnenmetriken

In der folgenden Tabelle werden Standardkampagnenmetriken aufgeführt und beschrieben, die Sie abfragen können, um die Leistung einer einzelnen Kampagne zu bewerten. Um Daten für diese Metriken abzufragen, verwenden Sie die Ressource Kampagnenmetriken der Amazon-Pinpoint-API. Die Spalte kpi-name in der Tabelle gibt den Wert an, der für den `kpi-name`-Parameter in Ihrer Abfrage verwendet werden soll.

Kennzahl	Kpi-name	Beschreibung
Bounce rate (Unzustellbarkeits rate)	`hard-bounce-rate`	Prozentsatz der E-Mail-Na chrichten, die den Empfänger n nicht zugestellt werden konnten, für alle Kampagnen

Kennzahl	Kpi-name	Beschreibung
		ausführungen. Diese Metrik berücksichtigt nur permanente Unzustellbarkeiten, d. h. Nachrichten, bei denen es aufgrund eines Problems mit der E-Mail-Adresse des Empfängers dazu kommt, dass die Nachricht dauerhaft nicht zugestellt werden kann. Diese Metrik wird berechnet als Anzahl der unzustellbaren E-Mail-Nachrichten, die von allen Kampagnenausführungen gesendet wurden, dividiert durch die Anzahl der E-Mail-Nachrichten, die von all diesen Kampagnenausführungen gesendet wurden.

Kennzahl	Kpi-name	Beschreibung
Unzustellbarkeitsrate, gruppiert nach Kampagnen ausführung	`hard-bounce-rate-grouped-by-campaign-activity`	Prozentsatz der E-Mail-Nachrichten, die den Empfängern nicht zugestellt werden konnten, für jede Kampagnen ausführung. Diese Metrik berücksichtigt nur permanente Unzustellbarkeiten, d. h. Nachrichten, bei denen es aufgrund eines Problems mit der E-Mail-Adresse des Empfängers dazu kommt, dass die Nachricht dauerhaft nicht zugestellt werden kann. Diese Metrik wird berechnet als Anzahl der unzustellbaren E-Mail-Nachrichten, die von einer Kampagnen ausführung gesendet wurden, dividiert durch die Anzahl der E-Mail-Nachrichten, die von der Kampagnenausführung gesendet wurden. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnen ausführung eindeutig identifiziert.

Kennzahl	Kpi-name	Beschreibung
Delivery rate (Zustellungsrate)	`successful-delivery-rate`	Prozentsatz der Nachrichten, die Empfängern zugestellt wurden, für alle Kampagnen ausführungen. Diese Metrik wird berechnet als Anzahl der Nachrichten, die von allen Kampagnen ausführungen gesendet und den Empfängern zugestell t wurden, dividiert durch die Anzahl der Nachrichten, die von all diesen Kampagnen ausführungen gesendet wurden.

Kennzahl	Kpi-name	Beschreibung
Zustellrate, gruppiert nach Kampagnenausführung	`successful-delivery-rate-grouped-by-campaign-activity`	Prozentsatz der Nachrichten, die Empfängern zugestellt wurden, für jede Kampagnenausführung. Diese Metrik wird berechnet als die Anzahl der Nachrichten, die von einer Kampagnenausführung gesendet und Empfängern zugestellt wurden, dividiert durch die Anzahl der Nachrichten, die von der Kampagnenausführung gesendet wurden. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnenausführung eindeutig identifiziert.

Kennzahl	Kpi-name	Beschreibung
Zustellungsrate, gruppiert nach Datum	`successful-delivery-rate-grouped-by-date`	Prozentsatz der Nachrichten, die Empfängern an jedem Tag im angegebenen Zeitraum zugestellt wurden, für alle Kampagnenausführungen. Diese Metrik wird als Anzahl der Nachrichten gesendet, die von allen Kampagnen ausführungen gesendet und Empfängern zugestell t wurden, dividiert durch die Anzahl der Nachrichten, die von all diesen Kampagnen ausführungen an jedem Tag im angegebenen Zeitraum gesendet wurden. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.

Kennzahl	Kpi-name	Beschreibung
Email open rate (E-Mail-Öffnungsrate)	`email-open-rate`	Prozentsatz der E-Mail-Nachrichten, die von Empfängern geöffnet wurden, für alle Kampagnenausführungen. Diese Metrik wird als Anzahl der E-Mail-Nachrichten berechnet, die von allen Kampagnenausführungen gesendet und von Empfängern geöffnet wurden, dividiert durch die Anzahl der E-Mail-Nachrichten, die von all diesen Kampagnenausführungen gesendet und Empfängern zugestellt wurden.

Kennzahl	Kpi-name	Beschreibung
E-Mail-Öffnungsrate, gruppiert nach Kampagnenausführung	`email-open-rate-grouped-by-campaign-activity`	Prozentsatz der E-Mail-Nachrichten, die von Empfängern geöffnet wurden, für jede Kampagnenausführung. Diese Metrik wird als Anzahl der E-Mail-Nachrichten berechnet, die von einer Kampagnenausführung gesendet und von Empfängern geöffnet wurden, dividiert durch die Anzahl der E-Mail-Nachrichten, die von der Kampagnenausführung gesendet und Empfängern zugestellt wurden. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnenausführung eindeutig identifiziert.

Kennzahl	Kpi-name	Beschreibung
Geöffnete E-Mails, gruppiert nach Kampagnenausführung	`direct-email-opens-grouped-by-campaign-activity`	Anzahl der E-Mail-Nachrichten, die von Empfängern geöffnet wurden, für jede Kampagnenausführung. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnenausführung eindeutig identifiziert.
Endpunktzustellungen	`unique-deliveries`	Anzahl der eindeutigen Endpunkte, an die Nachrichten gesendet wurden, für alle Kampagnenausführungen.
Endpunktzustellungen, gruppiert nach Kampagnenausführungen	`unique-deliveries-grouped-by-campaign-activity`	Anzahl der eindeutigen Endpunkte, an die Nachrichten gesendet wurden, für jede Kampagnenausführung. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnenausführung eindeutig identifiziert.

Kennzahl	Kpi-name	Beschreibung
Endpunktzustellungen, gruppiert nach Datum	`unique-deliveries-grouped-by-date`	Anzahl der eindeutigen Endpunkte, an die an jedem Tag im angegebenen Zeitraum Nachrichten gesendet wurden, für alle Kampagnenausführungen. Die Abfrageergebnisse für diese Metrik werden im erweiterten ISO-8601-Format nach Kalendertag gruppiert.
Links, auf die geklickt wurde, gruppiert nach Kampagnen ausführung	`clicks-grouped-by-campaign-activity`	Anzahl der Klicks von Empfängern auf die Links in der E-Mail-Nachricht, für jede Kampagnenausführung. Wenn ein Empfänger auf mehrere Links in der Nachricht oder mehrmals auf denselben Link geklickt hat, wird jeder dieser Klicks gezählt. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnen ausführung eindeutig identifiziert.

Kennzahl	Kpi-name	Beschreibung
Zugestellte Nachrichten, gruppiert nach Kampagnen ausführung	`successful-deliveries-grouped-by-campaign-activity`	Anzahl der Nachrichten, die Empfängern zugestellt wurden, für jede Kampagnen ausführung. Diese Metrik wird als Anzahl der Nachrichten berechnet , die von einer Kampagnen ausführung gesendet wurden, abzüglich der Anzahl der Nachrichten, die den Empfängern der Kampagnen ausführung aufgrund einer permanenten Unzustell barkeit nicht zugestellt werden konnten. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfo lge, die eine Kampagnen ausführung eindeutig identifiz iert.

Kennzahl	Kpi-name	Beschreibung
Gesendete Nachrichten, gruppiert nach Kampagnenausführung	`attempted-deliveries-grouped-by-campaign-activity`	Anzahl der gesendeten Nachrichten für jede Kampagnenausführung. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnenausführung eindeutig identifiziert.
Push open rate (Öffnungsrate nach Push)	`push-open-rate`	Prozentsatz der Push-Benachrichtigungen, die von Empfängern geöffnet wurden, für jede Kampagnenausführung. Diese Metrik wird als Anzahl der Push-Benachrichtigungen berechnet, die von allen Kampagnenausführungen gesendet und von Empfängern geöffnet wurden, dividiert durch die Anzahl der Push-Benachrichtigungen, die von all diesen Kampagnenausführungen gesendet und Empfängern zugestellt wurden.

Kennzahl	Kpi-name	Beschreibung
Push-Öffnungsrate, gruppiert nach Kampagnenausführung	`push-open-rate-grouped-by-campaign-activity`	Prozentsatz der Push-Benachrichtigungen, die von Empfängern geöffnet wurden, für jede Kampagnenausführung. Diese Metrik wird als Anzahl der Push-Benachrichtigungen berechnet, die von einer Kampagnenausführung gesendet und von den Empfängern geöffnet wurden, dividiert durch die Anzahl der Push-Benachrichtigungen, die von der Kampagnenausführung gesendet und Empfängern zugestellt wurden. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnenausführung eindeutig identifiziert.

Kennzahl	Kpi-name	Beschreibung
Insgesamt geöffnete Push-Benachrichtigungen, gruppiert nach Kampagnenausführung	`direct-push-opens-grouped-by-campaign-activity`	Anzahl der Push-Benachrichtigungen, die von Empfängern geöffnet wurden, für jede Kampagnenausführung. Die Abfrageergebnisse für diese Metrik werden nach Kampagnenaktivitäts-ID (`CampaignActivityId`) gruppiert, also einer Zeichenfolge, die eine Kampagnenausführung eindeutig identifiziert.
Total SMS spend (Gesamte SMS-Ausgaben)	`sms-spend`	Für alle Kampagnen der Gesamtbetrag, ausgedrückt in Millicent, der für den Versand von SMS ausgegeben wurde.

Journey-Engagement-Metriken

Die folgende Tabelle beschreibt Standard-Journey-Engagement-Metriken, die Sie abfragen können, um Trends für alle E-Mail-Nachrichten zu überwachen, die von einer Amazon-Pinpoint-Journey gesendet wurden. Um Daten für diese Metriken abzufragen, verwenden Sie die Ressource Journey-Engagement-Metriken der Amazon-Pinpoint-API. Die Spalte kpi-name in der Tabelle gibt den Wert an, der für den `kpi-name`-Parameter in einer Abfrage verwendet werden soll.

Kennzahl	Kpi-name	Beschreibung
Clicks (Klickvorgänge)	`journey-emails-clicked`	Die Anzahl der Klicks, die Teilnehmer auf Links in den Nachrichten ausgeführt haben. Wenn ein einzelner Teilnehmer mehrere Links in einer

Kennzahl	Kpi-name	Beschreibung
		Nachricht angeklickt hat oder mehr als einmal auf denselben Link geklickt hat, wird jeder Klick in die Zählung einbezoge n.
Klicks, gruppiert nach Aktivität	`emails-clicked-gro uped-by-journey-ac tivity`	Für jede Aktivität der Journey die Anzahl der Klicks der Teilnehmer auf Links in den Nachrichten. Wenn ein einzelner Teilnehmer mehrere Links in einer Nachricht angeklickt hat oder mehr als einmal auf denselben Link geklickt hat, wird jeder Klick in die Zählung einbezogen. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyAc tivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.
Complaints (Beschwerden)	`journey-emails-com plained`	Die Anzahl der Nachrichten, die von den Teilnehmern als unerwünschte oder unerwünsc hte E-Mail gemeldet wurden.

Kennzahl	Kpi-name	Beschreibung
Beschwerden, gruppiert nach Aktivitäten	`emails-complained-grouped-by-journey-activity`	Für jede Aktivität der Journey die Anzahl der Nachrichten, die von den Teilnehmern als unaufgeforderte oder unerwünschte E-Mail gemeldet wurden. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyActivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.
Deliveries (Zustellungen)	`journey-emails-delivered`	Die Anzahl der Nachrichten, die an Teilnehmer zugestellt wurden. Diese Metrik wird berechnet als die Anzahl der gesendeten Nachrichten, abzüglich der Anzahl der Nachrichten, die aufgrund eines Soft- oder Hard-Bounce nicht zugestellt werden konnten oder weil sie abgelehnt wurden.

Kennzahl	Kpi-name	Beschreibung
Zustellungen, gruppiert nach Aktivitäten	`emails-delivered-grouped-by-journey-activity`	Für jede Aktivität der Journey die Anzahl der Nachrichten, die den Teilnehmern zugestellt wurden. Diese Metrik wird berechnet als die Anzahl der gesendeten Nachrichten, abzüglich der Anzahl der Nachrichten, die aufgrund eines Soft- oder Hard-Bounce nicht zugestellt werden konnten oder weil sie abgelehnt wurden, für jede Aktivität der Journey. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyActivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.
Hard bounces (Permanente Unzustellbarkeiten)	`journey-emails-hardbounced`	Die Anzahl der Nachrichten, die den Teilnehmern aufgrund eines Hard-Bounce nicht zugestellt werden konnten. Eine permanente Unzustellbarkeit tritt auf, wenn ein anhaltendes Problem die Zustellung einer Nachricht verhindert, z. B. wenn die E-Mail-Adresse eines Teilnehmers nicht existiert.

Kennzahl	Kpi-name	Beschreibung
Hard-Bounces, gruppiert nach Aktivität	`emails-hardbounced-grouped-by-journey-activity`	Für jede Aktivität der Journey die Anzahl der Nachrichten, die den Teilnehmern aufgrund eines Hard-Bounce nicht zugestellt werden konnten. Eine permanente Unzustellbarkeit tritt auf, wenn ein anhaltendes Problem die Zustellung einer Nachricht verhindert, z. B. wenn die E-Mail-Adresse eines Teilnehmers nicht existiert. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyActivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.
Opens (Öffnungsvorgänge)	`journey-emails-opened`	Die Anzahl der Nachrichten, die von Teilnehmern geöffnet wurden.
Geöffnet, gruppiert nach Aktivität	`emails-opened-grouped-by-journey-activity`	Für jede Aktivität der Journey die Anzahl der Nachrichten, die von den Teilnehmern geöffnet wurden. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyActivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.

Kennzahl	Kpi-name	Beschreibung
Ablehnungen	`journey-emails-rejected`	Die Anzahl der Nachrichten, die nicht an die Teilnehmer gesendet wurden, weil sie abgelehnt wurden. Eine Nachricht wird zurückgewiesen, wenn Amazon Pinpoint feststellt, dass sie Malware enthält. Amazon Pinpoint versucht nicht, abgelehnte Nachrichten zu senden.
Ablehnungen, gruppiert nach Aktivität	`emails-rejected-grouped-by-journey-activity`	Für jede Aktivität der Journey die Anzahl der Nachrichten, die nicht an die Teilnehmer gesendet wurden, weil sie abgelehnt wurden. Eine Nachricht wird zurückgewiesen, wenn Amazon Pinpoint feststellt, dass sie Malware enthält. Amazon Pinpoint versucht nicht, abgelehnte Nachrichten zu senden. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyActivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.
Sends (Sendevorgänge)	`journey-emails-sent`	Anzahl der gesendeten Nachrichten.

Kennzahl	Kpi-name	Beschreibung
Gesendet, gruppiert nach Aktivität	`emails-sent-grouped-by-journey-activity`	Für jede Aktivität der Journey die Anzahl der gesendeten Nachrichten. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyActivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.
Temporäre Unzustellbarkeiten	`journey-emails-softbounced`	Die Anzahl der Nachrichten, die aufgrund eines Soft-Bounce nicht an die Teilnehmer übermittelt werden konnten. Eine temporäre Unzustellbarkeit tritt auf, wenn ein vorübergehendes Problem die Zustellung einer Nachricht verhindert, z. B. wenn der Posteingang eines Teilnehmers voll ist oder wenn der empfangende Server vorübergehend nicht verfügbar ist.

Kennzahl	Kpi-name	Beschreibung
Soft-Bounces, gruppiert nach Aktivität	`emails-softbounced-grouped-by-journey-activity`	Für jede Aktivität der Journey die Anzahl der Nachrichten, die aufgrund eines Soft-Bounce nicht an die Teilnehmer zugestellt werden konnten. Eine temporäre Unzustellbarkeit tritt auf, wenn ein vorübergehendes Problem die Zustellung einer Nachricht verhindert, z. B. wenn der Posteingang eines Teilnehmers voll ist oder wenn der empfangende Server vorübergehend nicht verfügbar ist. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyActivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.
Abbestellungen	`journey-emails-unsubscribed`	Die Anzahl der Male, die Teilnehmer auf Abmeldelinks in den Nachrichten geklickt haben. Wenn ein einzelner Teilnehmer mehrmals auf den gleichen Abmeldelink geklickt hat, wird jeder Klick in die Zählung einbezogen.

Kennzahl	Kpi-name	Beschreibung
Abmeldungen, gruppiert nach Aktivität	`emails-unsubscribed-grouped-by-journey-activity`	Für jede Aktivität der Journey die Anzahl der Male, in denen die Teilnehmer auf Abmeldelinks in den Nachrichten geklickt haben. Wenn ein einzelner Teilnehmer mehrmals auf den gleichen Abmeldelink geklickt hat, wird jeder Klick in die Zählung einbezogen. Die Abfrageergebnisse für diese Metrik werden nach der Aktivitäts-ID (`JourneyActivityId`) gruppiert, die eine Zeichenfolge ist, die eine Aktivität eindeutig identifiziert.

Journey-Ausführungsmetriken

Die folgende Tabelle beschreibt Standardausführungsmetriken, die Sie abfragen können, um den Status der Teilnehmer einer Amazon-Pinpoint-Journey zu bewerten. Um Daten für diese Metriken abzufragen, verwenden Sie die Ressource Journey-Ausführungsmetriken der Amazon-Pinpoint-API. Die Spalte Field (Feld) in der Tabelle identifiziert den Namen des Feldes, das in den Suchergebnissen für jede Metrik angezeigt wird.

Kennzahl	Feld	Beschreibung
Aktive Teilnehmer	`ENDPOINT_ACTIVE`	Die Anzahl der Teilnehmer, die aktiv durch die Aktivitäten der Journey durchlaufen. Diese Metrik berechnet sich aus der Anzahl der Teilnehmer, die in die Journey eingetret

Kennzahl	Feld	Beschreibung
		en sind, abzüglich der Anzahl der Teilnehmer, die die Journey verlassen haben, und der Anzahl der Teilnehmer, die aus der Journey entfernt wurden.
Teilnehmer-Stornierungen	`CANCELLED`	Die Anzahl der Teilnehme r, die die Journey nicht abgeschlossen haben, weil die Journey storniert wurde.
Teilnehmerabgänge	`ENDPOINT_LEFT`	Die Anzahl der Teilnehme r, die die Journey verlassen haben.
Teilnehmer	`ENDPOINT_ENTERED`	Die Anzahl der Teilnehme r, die die Journey begonnen haben.
Teilnehmerausnahmen, Wiedereintrittslimits	`REENTRY_CAP_EXCEEDED`	Die Anzahl der Teilnehmer, die die Journey nicht beendet haben, weil sie die maximale Anzahl der Male überschri tten hätten, die ein einzelner Teilnehmer die Journey wieder aufnehmen kann.

Kennzahl	Feld	Beschreibung
Teilnehmerausnahmen, Ablehnungen	`ACTIVE_ENDPOINT_REJECTED`	Die Anzahl der Teilnehmer, die die Journey nicht starten können, weil sie bereits ein aktiver Teilnehmer an der Journey sind. Ein Teilnehmer wird abgelehnt, wenn der Teilnehmer eine Journey startet und Sie anschließend die Endpunktdefinition des Teilnehmers auf eine Weise aktualisieren, die sich auf die Einbeziehung der Teilnehmer in ein Segment (basierend auf Segmentkriterien) oder die Journey (basierend auf Aktivitätsbedingungen) auswirkt.

Journey-Aktivitätsausführungsmetriken

Die folgende Tabelle beschreibt Standardausführungsmetriken, die Sie abfragen können, um den Status der Teilnehmer in jedem Typ der individuellen Aktivität einer Amazon-Pinpoint-Journey zu bewerten. Um Daten für diese Metriken abzufragen, verwenden Sie die Ressource Journey-Aktivität-Ausführungsmetriken der Amazon-Pinpoint-API. Die Spalte Metrics (Metriken) in der Tabelle listet die Felder auf, die in den Suchergebnissen für jede Aktivitätsart erscheinen. Sie enthält auch eine kurze Beschreibung der einzelnen Felder.

Aktivitätstyp	Metriken
Ja/Nein-Split (`CONDITIONAL_SPLIT`)	Die Metriken sind: • `Branch_FALSE` : Die Anzahl der Teilnehmer, die die Bedingungen der Aktivität nicht

Aktivitätstyp	Metriken
	erfüllten und mit der Aktivität auf dem „Nein“-Pfad fortfahren. • `Branch_TRUE` : Die Anzahl der Teilnehmer, die die Bedingungen der Aktivität erfüllten und mit der Aktivität auf dem „Ja“-Pfad fortfahren. Für jeden Pfad in der Aktivität stehen zusätzliche Metriken zur Verfügung. Informationen zu diesen Metriken finden Sie in der Zeile für die Aktivitätsart der Tabelle.
Holdout (`HOLDOUT`)	Die Metriken sind: • `HOLDOUT`: Die Anzahl der Teilnehmer, die im Rahmen des Holdout-Prozentsatzes für die Aktivität von der Journey entfernt wurden. • `PASSED`: Die Anzahl der Teilnehmer, die zur nächsten Aktivität der Journey gewechselt sind.

Aktivitätstyp	Metriken
E-Mail (`MESSAGE`)	Die Metriken sind: • `DAILY_CAP_EXCEEDED` : Die Anzahl der Nachrichten, die nicht gesendet wurden, weil sie die maximale Anzahl von Nachricht en überschritten hätten, die ein einzelner Teilnehmer innerhalb eines 24-Stunden-Zeitraums empfangen kann. • `FAILURE_PERMANENT` : Die Anzahl der Nachrichten, die aufgrund eines permanent en Problems nicht gesendet wurden. • `QUIET_TIME` : Die Anzahl der Nachricht en, die nicht gesendet wurden, weil sie in Ruhezeiten für die Zeitzone des Teilnehmers zugestellt worden wären. • `SERVICE_FAILURE` : Die Anzahl der Nachrichten, die aufgrund eines Problems mit Amazon Pinpoint nicht gesendet wurden. • `SUCCESS`: Die Anzahl der Nachrichten, die erfolgreich an die Teilnehmer zugestellt wurden. • `THROTTLED` : Die Anzahl der Nachrichten, die nicht gesendet wurden, weil dadurch die Sendekontingente für Ihr Amazon-Pinpoint-Konto überschritten worden wären. • `TRANSIENT_FAILURE` : Die Anzahl der Nachrichten, die aufgrund eines temporären Problems nicht gesendet wurden. • `UNKNOWN`: Die Anzahl der Nachrichten, die aufgrund eines unbekannten Problems nicht gesendet wurden.

Aktivitätstyp	Metriken
Mehrfach-Split (`MULTI_CONDITIONAL_SPLIT`)	Für jeden Pfad der Aktivität die Anzahl der Teilnehmer, die zur Aktivität auf dem Pfad übergegangen sind. Die Abfrageergebnisse für diese Metrik sind nach Pfad gruppiert: `Branch_#`, wobei `#` der numerische Bezeichner für einen Pfad ist, zum Beispiel `Branch_1` für den ersten Pfad der Aktivität. Für jeden Pfad in der Aktivität stehen zusätzlic he Metriken zur Verfügung. Informationen zu diesen Metriken finden Sie in der Zeile für die Aktivitätsart der Tabelle.
Zufalls-Split (`RANDOM_SPLIT`)	Für jeden Pfad der Aktivität die Anzahl der Teilnehmer, die zur Aktivität auf dem Pfad übergegangen sind. Die Abfrageergebnisse für diese Metrik sind nach Pfad gruppiert: `Branch_#`, wobei `#` der numerische Bezeichner für einen Pfad ist, zum Beispiel `Branch_1` für den ersten Pfad der Aktivität. Für jeden Pfad in der Aktivität stehen zusätzlic he Metriken zur Verfügung. Informationen zu diesen Metriken finden Sie in der Zeile für die Aktivitätsart der Tabelle.

Aktivitätstyp	Metriken
Warten (`WAIT`)	Die Metriken sind: • `WAIT_FINISHED` : Die Anzahl der Teilnehmer, die die angegebene Wartezeit beendet haben. • `WAIT_SKIPPED` : Die Anzahl der Teilnehmer, die nicht die angegebene Zeit gewartet haben, typischerweise weil sie die Aktivität oder Journey nach der geplanten Endzeit für die Aktivität begonnen haben. • `WAIT_STARTED` : Die Anzahl der Teilnehmer, die mit dem Warten begonnen haben und die die angegebene Zeitspanne nicht übersprungen oder beendet haben.

Aktivitätstyp	Metriken
Kontaktcenter (`CONTACT_CENTER`)	Die Metriken sind: • `CALL_QUEUED` : Die Anzahl der Teilnehmer, die sich bei Amazon Connect eingewählt haben und in die Warteschlange gestellt wurden. Beinhaltet Wahlwiederholungsversuche. • `CONTINUE_WAITING` : Die Anzahl der Teilnehmer, die weiterhin auf Wahlversuche warten. • `DIAL_FAILURE` : Die Anzahl der Teilnehmer mit fehlgeschlagenen täglichen Versuchen. • `DROPPED`: Die Anzahl der Teilnehmer, die zum Zeitpunkt des Versandes die in früheren Journey-Aktivitäten festgelegten Bedingungen nicht mehr erfüllen. • `TIMEOUT`: Die Anzahl der Teilnehmer, die nach mehreren Wählversuchen keinen Amazon-Connect-Dispositionscode erhalten haben. • `WAIT_FINISHED` : Die Anzahl der Teilnehmer, die die angegebene Wartezeit beendet haben. • `WAIT_FOR_QUIET_HOURS` : Die Anzahl der Teilnehmer, die darauf warten, dass die Ruhezeit beendet ist, um an den Kanal zu senden. • `WAIT_STARTED` : Die Anzahl der Teilnehmer, die mit dem Warten begonnen haben und die die angegebene Zeitspanne nicht übersprungen oder beendet haben.

Ausführungsmetriken zu Journey und Kampagne

Sie können Standardausführungsmetriken abfragen, um den Status der Teilnehmer in jedem Typ der individuellen Aktivität einer Amazon-Pinpoint-Journey oder -Kampagne zu bewerten. Um Daten für diese Metriken abzufragen, verwenden Sie die Ressource Journey-Laufaktivität-Ausführungsmetriken oder Kampagnenmetriken der Amazon-Pinpoint-API. Die folgende Tabelle listet die Felder auf, die in den Suchergebnissen für jede Aktivitätsart erscheinen.

Metrikname	Gilt für Journeys, Kampagnen oder beides	Beschreibung
ENDPOINT_PRODUCED	beides	Die Anzahl der Endpunkte , die ursprünglich aus dem Segment oder Ereignis erstellt wurden, bevor eine Filterung vorgenommen wurde.
ENDPOINTS_FROM_USER	beides	Wenn der Kunde nur über ein Benutzer-ID-Segment verfügt, werden alle Endpunkte dieser Benutzer hinzugefügt. Diese Metrik misst die Anzahl der Endpunkte, die auf diese Weise hinzugefügt wurden.
ENDPOINT_OPT_OUT	beides	Der Endpunkt wurde deaktivie rt und hat nicht an der Kampagne oder Journey teilgenommen.
ENDPOINT_INACTIVE	beides	Der Endpunkt war inaktiv und hat nicht an der Kampagne oder Journey teilgenommen.
FILTERED_OUT_BY_SE GMENT	beides	Der Endpunkt entsprach nicht den Segmentfiltern und hat weder an der Kampagne noch an der Journey teilgenommen.

Metrikname	Gilt für Journeys, Kampagnen oder beides	Beschreibung
ENDPOINT_MISSING_ADDRESS	beides	Dem Endpunkt fehlte eine Adresse und er hat weder an der Kampagne noch an der Journey teilgenommen.
ENDPOINT_MISSING_CHANNEL	beides	Dem Endpunkt fehlte ein Kanal und er hat weder an der Kampagne noch an der Journey teilgenommen.
ENDPOINT_MISSING_TIMEZONE	beides	Dem Endpunkt fehlte ein Wert für die Zeitzone und er wurde herausgefiltert. Dies passiert nur, wenn ein Zeitzonenwert erforderlich ist.
ENDPOINT_TIMEZONE_MISMATCH	beides	Der Endpunkt befand sich in einer Zeitzone, die zu diesem Zeitpunkt nicht in der Ausführung enthalten war.
ENDPOINT_CHANNEL_MISMATCH	Kampagnen	Für die Kampagne ist keine Nachricht für den Kanaltyp dieses Endpunkts konfiguriert.
DUPLICATE_ENDPOINT	beides	Doppelte Endpunkte wurden gefunden und dedupliziert.
DUPLICATE_USER	beides	Doppelte Benutzer wurden gefunden und aus einem Segment, das nur eine Benutzer-ID hat, dedupliziert. Wenn sie dieselbe Benutzer-ID haben, wird die Metrik 1 ausgegeben.

Metrikname	Gilt für Journeys, Kampagnen oder beides	Beschreibung
PAUSED	Journeys	Aus der Ausführung entfernt, weil die Journey unterbrochen wurde.
ENDED	Journeys	Aus der Ausführung entfernt, weil die Journey beendet wurde.
TREATMENT_HOLDOUT	Kampagnen	Dies wird in A/B-Kampagnen für Endgeräte ausgegebe n, deren Kohorten nicht der aktuellen Behandlung entsprechen. Bei einer A/B-Aufteilung von 50/50 geben beispielsweise 50 % der Endpunkte diese Metrik für jede Behandlung aus
ENDPOINT_ESTIMATED _TIMEZONE	Journeys	Die Zeitzonenschätzung war in der Lage, eine Zeitzone für den Endpunkt zu schätzen.

Abfragen von Amazon-Pinpoint-Analysedaten für Kampagnen

Zusätzlich zur Verwendung der Analyseseiten auf der Amazon-Pinpoint-Konsole können Sie Amazon-Pinpoint-Analyse-APIs verwenden, um Analysedaten für eine Teilmenge von Standardmetriken abzufragen, die Einblicke in Bereitstellungs- und Interaktionstrends für Kampagnen bieten.

Jede dieser Metriken ist ein messbarer Wert, auch als Key Performance Indicator (KPI) bezeichnet, der Ihnen helfen kann, die Leistung einer oder mehrerer Kampagnen zu überwachen und zu bewerten. Beispielsweise können Sie mit einer Metrik herausfinden, an wie viele Endpunkte eine Kampagnennachricht gesendet wurde oder wie viele dieser Nachrichten an die vorgesehenen Endpunkte zugestellt wurden.

Amazon Pinpoint sammelt und aggregiert diese Daten automatisch für alle Ihre Kampagnen. Der Service speichert die Daten für 90 Tage. Wenn Sie eine mobile App mithilfe eines mobilen AWS-SDK in Amazon Pinpoint integriert haben, erweitert Amazon Pinpoint diese Unterstützung, um zusätzliche Metriken einzuschließen, z. B. den Prozentsatz der Push-Benachrichtigungen, die von Empfängern geöffnet wurden. Hinweise zum Integrieren einer mobilen Anwendung finden Sie unter Integrieren von Amazon Pinpoint in Ihre Anwendung.

Wenn Sie Amazon-Pinpoint-Analyse-APIs zum Abfragen von Daten verwenden, können Sie verschiedene Optionen auswählen, die den Bereich, die Daten, die Gruppierung und die Filter für Ihre Abfrage definieren. Dazu verwenden Sie Parameter, die das Projekt, die Kampagne und die Metrik angeben, die Sie abfragen möchten, zusätzlich zu allen datumsbasierten Filtern, die Sie anwenden möchten.

In diesem Thema wird erläutert und es werden Beispiele aufgeführt, wie Sie diese Optionen auswählen und die Daten für eine oder mehrere Kampagnen abfragen.

Voraussetzungen

Bevor Sie Analysedaten für eine oder mehrere Kampagnen abfragen, ist es sinnvoll, die folgenden Informationen zu sammeln, die Sie zur Definition Ihrer Abfrage verwenden:

- Projekt-ID: Der eindeutige Bezeichner für das Projekt, das der Kampagne oder den Kampagnen zugeordnet ist. In der Amazon-Pinpoint-API wird dieser Wert in der -Eigenschaft gespeichert. Auf der Amazon-Pinpoint-Konsole wird dieser Wert als Projekt-ID auf der Seite Alle Projekte angezeigt.
- Kampagnen-ID: Die eindeutige Kennung für die Kampagne, wenn Sie die Daten nur für eine Kampagne abfragen möchten. In der Amazon-Pinpoint-API wird dieser Wert in der `campaign-id`-Eigenschaft gespeichert. Dieser Wert wird nicht auf der Konsole angezeigt.
- Datumsbereich: Optional das erste und letzte Datum und die Uhrzeit des Datumsbereichs, für den die Daten abgefragt werden sollen. Datumsbereiche werden inklusiv angegeben und dürfen maximal 31 Kalendertage umfassen. Darüber hinaus müssen sie vor 90 Tagen ab dem aktuellen Tag beginnen. Wenn Sie keinen Datumsbereich angeben, fragt Amazon Pinpoint die Daten für die letzten 31 Kalendertage automatisch ab.
- Metriktyp: Der Typ der abzufragenden Metrik. Es gibt zwei Typen, Anwendungsmetriken und Kampagnenmetriken. Eine Anwendungsmetrik liefert Daten für alle Kampagnen, die einem Projekt zugeordnet sind, auch als Anwendung bezeichnet. Eine Kampagnenmetrik liefert Daten für nur eine Kampagne.

- Metrik: Der Name der abzufragenden Metrik, genauer gesagt der `kpi-name`-Wert für die Metrik. Eine vollständige Liste der unterstützten Metriken und den `kpi-name`-Wert für die einzelnen Metriken finden Sie unter Standardmetriken.

Es hilft auch zu bestimmen, ob Sie die Daten nach einem relevanten Feld gruppieren möchten. In diesem Fall können Sie die Analyse und Berichterstellung vereinfachen, indem Sie eine Metrik auswählen, mit der Daten automatisch gruppiert werden sollen. Amazon Pinpoint stellt beispielsweise mehrere Standardmetriken bereit, die den Prozentsatz der Nachrichten melden, die an Empfänger einer Kampagne übermittelt wurden. Eine dieser Metriken gruppiert die Daten automatisch nach Datum (`successful-delivery-rate-grouped-by-date`). Eine weitere Metrik gruppiert die Daten automatisch nach Kampagnenlauf (`successful-delivery-rate-grouped-by-campaign-activity`). Eine dritte Metrik gibt einfach einen einzigen Wert zurück – den Prozentsatz der Meldungen, die bei allen Kampagnenläufen (`successful-delivery-rate`) an die Empfänger zugestellt wurden.

Wenn Sie keine Standardmetrik finden können, die Daten so gruppiert, wie Sie es wünschen, können Sie eine Reihe von Abfragen entwickeln, die die gewünschten Daten zurückgeben. Sie können dann die Abfrageergebnisse manuell aufschlüsseln oder in benutzerdefinierte Gruppen zusammenfassen, die Sie entwerfen.

Schließlich ist es wichtig zu überprüfen, ob Sie berechtigt sind, auf die Daten zuzugreifen, die Sie abfragen möchten. Weitere Informationen finden Sie unter IAM-Richtlinien zum Abfragen von Amazon-Pinpoint-Analysedaten.

Abfragen von Daten für eine Kampagne

Um die Daten für eine Kampagne abzufragen, verwenden Sie die Kampagnenmetriken-API und geben Werte für die folgenden erforderlichen Parameter an:

- application-id: Die Projekt-ID, bei der es sich um den eindeutigen Bezeichner für das Projekt handelt, das der Kampagne zugeordnet ist. In Amazon Pinpoint haben die Begriffe Projekt und Anwendung dieselbe Bedeutung.
- campaign-id: Der eindeutige Bezeichner für die Kampagne.
- kpi-name: Der Name der abzufragenden Metrik. Dieser Wert beschreibt die zugeordnete Metrik und besteht aus zwei oder mehr Begriffen, die aus alphanumerischen Kleinbuchstaben bestehen, die durch einen Bindestrich getrennt sind. Eine vollständige Liste der unterstützten Metriken und den `kpi-name`-Wert für die einzelnen Metriken finden Sie unter Standardmetriken.

Sie können auch einen Filter anwenden, der die Daten für einen bestimmten Zeitraum abfragt. Wenn Sie keinen Datumsbereich angeben, werden die Daten für die letzten 31 Kalendertage von Amazon Pinpoint zurückgegeben. Um die Daten nach verschiedenen Datumsangaben zu filtern, verwenden Sie die unterstützten Datumsbereichsparameter, um das erste und letzte Datum und die Uhrzeit des Datumsbereichs anzugeben. Die Werte sollten im erweiterten ISO 8601-Format vorliegen und die koordinierte Weltzeit (UTC) verwenden, z. B. `2019-07-19T20:00:00Z` für 20.00 Uhr UTC am 19. Juli 2019. Datumsbereiche werden inklusiv angegeben und dürfen maximal 31 Kalendertage umfassen. Darüber hinaus müssen das erste Datum und die erste Uhrzeit früher als 90 Tage ab dem aktuellen Tag liegen.

In den folgenden Beispielen wird gezeigt, wie Analysedaten für eine Kampagne mithilfe der Amazon-Pinpoint-REST-API, der AWS CLI und des AWS SDK for Java abgefragt werden. Sie können jedes unterstützte AWS SDK verwenden, um Analysedaten für eine Kampagne abzufragen. Die AWS CLI-Beispiele sind für Microsoft Windows formatiert. Ersetzen Sie für Unix, Linux und macOS das Zeilenfortsetzungszeichen (^) durch einen umgekehrten Schrägstrich (\).

REST API

Um Analysedaten für eine Kampagne mithilfe der Amazon-Pinpoint-REST-API abzufragen, senden Sie eine HTTP(S)-GET-Anforderung an den Kampagnenmetriken-URI. Geben Sie im URI die entsprechenden Werte für die erforderlichen Pfadparameter an:

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

Wobei gilt:

- *`endpoint`* ist der Amazon-Pinpoint-Endpunkt für die AWS-Region, die das Projekt hostet, das der Kampagne zugeordnet ist.
- *`application-id`* ist der eindeutige Bezeichner für das Projekt, das der Kampagne zugeordnet ist.
- *`campaign-id`* ist der eindeutige Bezeichner für die Kampagne.
- *`kpi-name`* ist der `kpi-name`-Wert für die abzufragende Metrik.

Alle Parameter sollten URL-codiert sein.

Wenn Sie einen Filter anwenden möchten, der die Daten für einen bestimmten Datumsbereich abfragt, hängen Sie die `start-time`- und `end-time`-Abfrageparameter und -werte an den

URI an. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Verwenden Sie ein kaufmännisches Und-Zeichen (&), um die Parameter zu trennen.

Die folgende Anforderung ruft beispielsweise die Anzahl der eindeutigen Endpunkte ab, an die Nachrichten von allen Ausführungen einer Kampagne vom 19. Juli 2019 bis zum 26. Juli 2019 übermittelt wurden:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456789012345example/
campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Wobei gilt:

- `pinpoint.us-east-1.amazonaws.com` ist der Amazon-Pinpoint-Endpunkt für die AWS-Region, die das Projekt hostet.
- `12345678901234567890123456789012345example` ist der eindeutige Bezeichner für das Projekt, das der Kampagne zugeordnet ist.
- `80b8efd84042ff8d9c96ce2f8example` ist der eindeutige Bezeichner für die Kampagne.
- `unique-deliveries` ist der `kpi-name`-Wert für die Endpunktzustellungen-Kampagnenmetrik, d. h. die Metrik, die die Anzahl der eindeutigen Endpunkte angibt, an die Nachrichten von allen Ausführungen einer Kampagne übermittelt wurden.
- `2019-07-19T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-07-26T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

AWS CLI

Verwenden Sie den get-campaign-date-range-kpi-Befehl und geben Sie die entsprechenden Werte für die erforderlichen Parameter an, um Analysedaten für eine Kampagne mithilfe der AWS CLI abzufragen:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id application-id ^
```

```
    --campaign-id campaign-id ^
    --kpi-name kpi-name
```

Wobei gilt:

- *application-id* ist der eindeutige Bezeichner für das Projekt, das der Kampagne zugeordnet ist.
- *campaign-id* ist der eindeutige Bezeichner für die Kampagne.
- *kpi-name* ist der `kpi-name`-Wert für die abzufragende Metrik.

Um einen Filter anzuwenden, der die Daten für einen bestimmten Zeitraum abfragt, fügen Sie der Abfrage die `start-time`- und `end-time`-Parameter und -Werte hinzu. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Die folgende Anforderung ruft beispielsweise die Anzahl der eindeutigen Endpunkte ab, an die Nachrichten von allen Ausführungen einer Kampagne vom 19. Juli 2019 bis zum 26. Juli 2019 übermittelt wurden:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id 12345678901234567890123456789012345example ^
    --campaign-id 80b8efd84042ff8d9c96ce2f8example ^
    --kpi-name unique-deliveries ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Wobei gilt:

- `12345678901234567890123456789012345example` ist der eindeutige Bezeichner für das Projekt, das der Kampagne zugeordnet ist.
- `80b8efd84042ff8d9c96ce2f8example` ist der eindeutige Bezeichner für die Kampagne.
- `unique-deliveries` ist der `kpi-name`-Wert für die Endpunktzustellungen-Kampagnenmetrik, d. h. die Metrik, die die Anzahl der eindeutigen Endpunkte angibt, an die Nachrichten von allen Ausführungen einer Kampagne übermittelt wurden.
- `2019-07-19T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-07-26T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

SDK for Java

Um Analysedaten für eine Kampagne unter Verwendung der AWS SDK for Java abzufragen, verwenden Sie die Methode GetCampaignDateRangeKpiRequest der Campaign Metrics-API. Geben Sie die entsprechenden Werte für die erforderlichen Parameter an:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withCampaignId("campaignId")
        .withKpiName("kpiName")
```

Wobei gilt:

- *applicationId* ist der eindeutige Bezeichner für das Projekt, das der Kampagne zugeordnet ist.
- *campaignId* ist die eindeutige Kennung für die Kampagne.
- *kpiName* ist der `kpi-name`-Wert für die abzufragende Metrik.

Um einen Filter anzuwenden, der die Daten für einen bestimmten Datumsbereich abfragt, schließen Sie die `startTime`- und `endTime`-Parameter und -Werte in die Abfrage ein. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Die folgende Anforderung ruft beispielsweise die Anzahl der eindeutigen Endpunkte ab, an die Nachrichten von allen Ausführungen einer Kampagne vom 19. Juli 2019 bis zum 26. Juli 2019 übermittelt wurden:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withCampaignId("80b8efd84042ff8d9c96ce2f8example")
        .withKpiName("unique-deliveries")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Wobei gilt:

- `12345678901234567890123456example` ist der eindeutige Bezeichner für das Projekt, das der Kampagne zugeordnet ist.
- `80b8efd84042ff8d9c96ce2f8example` ist der eindeutige Bezeichner für die Kampagne.

- `unique-deliveries` ist der `kpi-name`-Wert für die Endpunktzustellungen-Kampagnenmetrik, d. h. die Metrik, die die Anzahl der eindeutigen Endpunkte angibt, an die Nachrichten von allen Ausführungen einer Kampagne übermittelt wurden.
- `2019-07-19T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-07-26T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

Nachdem Sie Ihre Abfrage gesendet haben, gibt Amazon Pinpoint die Abfrageergebnisse in einer JSON-Antwort zurück. Die Struktur der Ergebnisse hängt von der Metrik ab, die Sie abgefragt haben. Einige Metriken geben nur einen Wert zurück. Beispielsweise gibt die Kampagnenmetrik Endpunkt-Zustellungen (`unique-deliveries`), die in den vorherigen Beispielen verwendet wurde, einen Wert zurück – die Anzahl der eindeutigen Endpunkte, an die Nachrichten gesendet wurden, für jede Kampagnenausführung. In diesem Fall lautet die JSON-Antwort wie folgt:

```
{
    "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"
    }
}
```

Andere Metriken geben mehrere Werte zurück und gruppieren die Werte nach einem relevanten Feld. Wenn eine Metrik mehrere Werte zurückgibt, enthält die JSON-Antwort ein Feld, das angibt, welches Feld zum Gruppieren der Daten verwendet wurde.

Weitere Informationen zur Struktur von Abfrageergebnissen finden Sie unter Verwenden von Abfrageergebnissen.

Abfragen von Daten für mehrere Kampagnen

Es gibt zwei Möglichkeiten, die Daten für mehrere Kampagnen abzufragen. Der beste Weg hängt davon ab, ob Sie die Daten für Kampagnen abfragen möchten, die alle demselben Projekt zugeordnet sind. Wenn ja, hängt er außerdem davon ab, ob Sie die Daten für alle Kampagnen oder nur oder nur für eine Teilmenge dieser Kampagnen abfragen möchten.

Um die Daten für Kampagnen abzufragen, die verschiedenen Projekten zugeordnet sind, oder nur für eine Teilmenge der Kampagnen, die demselben Projekt zugeordnet sind, ist es am besten, eine Reihe von einzelnen Abfragen zu erstellen und auszuführen, eine für jede Kampagne, für die Sie die Daten abfragen möchten. Im vorangegangenen Abschnitt wird erläutert, wie die Daten nur für eine Kampagne abgefragt werden.

Um die Daten für alle Kampagnen abzufragen, die demselben Projekt zugeordnet sind, können Sie die Application Metrics-API verwenden. Geben Sie Werte für die folgenden erforderlichen Parameter an:

- application-id: Die Projekt-ID, bei der es sich um den eindeutigen Bezeichner für das Projekt handelt. In Amazon Pinpoint haben die Begriffe Projekt und Anwendung dieselbe Bedeutung.
- kpi-name: Der Name der abzufragenden Metrik. Dieser Wert beschreibt die zugeordnete Metrik und besteht aus zwei oder mehr Begriffen, die aus alphanumerischen Kleinbuchstaben bestehen, die durch einen Bindestrich getrennt sind. Eine vollständige Liste der unterstützten Metriken und den `kpi-name`-Wert für die einzelnen Metriken finden Sie unter Standardmetriken.

Sie können die Daten auch nach Datumsbereich filtern. Wenn Sie keinen Datumsbereich angeben, werden die Daten für die letzten 31 Kalendertage von Amazon Pinpoint zurückgegeben. Um die Daten nach verschiedenen Datumsangaben zu filtern, verwenden Sie die unterstützten Datumsbereichsparameter, um das erste und letzte Datum und die Uhrzeit des Datumsbereichs anzugeben. Die Werte sollten im erweiterten ISO 8601-Format vorliegen und die koordinierte Weltzeit (UTC) verwenden, z. B. `2019-07-19T20:00:00Z` für 20.00 Uhr UTC am 19. Juli 2019. Datumsbereiche werden inklusiv angegeben und dürfen maximal 31 Kalendertage umfassen.

Darüber hinaus müssen das erste Datum und die erste Uhrzeit früher als 90 Tage ab dem aktuellen Tag liegen.

In den folgenden Beispielen wird gezeigt, wie Analysedaten für eine Kampagne mithilfe der Amazon-Pinpoint-REST-API, der AWS CLI und des AWS SDK for Java abgefragt werden. Sie können jedes unterstützte AWS SDK verwenden, um Analysedaten für eine Kampagne abzufragen. Die AWS CLI-Beispiele sind für Microsoft Windows formatiert. Ersetzen Sie für Unix, Linux und macOS das Zeilenfortsetzungszeichen (^) durch einen umgekehrten Schrägstrich (\).

REST API

Um Analysedaten für mehrere Kampagnen mithilfe der Amazon-Pinpoint-REST-API abzufragen, senden Sie eine HTTP(S)-GET-Anforderung an den Anwendungsmetriken-URI. Geben Sie im URI die entsprechenden Werte für die erforderlichen Pfadparameter an:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Wobei gilt:

- *endpoint* ist der Amazon-Pinpoint--Endpunkt für die AWS-Region, die das Projekt hostet, das den Kampagnen zugeordnet ist.
- *application-id* ist der eindeutige Bezeichner für das Projekt, das den Kampagnen zugeordnet ist.
- *kpi-name* ist der `kpi-name`-Wert für die abzufragende Metrik.

Alle Parameter sollten URL-codiert sein.

Wenn Sie einen Filter anwenden möchten, der die Daten für einen bestimmten Datumsbereich abruft, hängen Sie die `start-time`- und `end-time`-Parameter und -Werte an den URI an. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Verwenden Sie ein kaufmännisches Und-Zeichen (&), um die Parameter zu trennen.

Die folgende Anforderung ruft beispielsweise die Anzahl der eindeutigen Endpunkte ab, an die Nachrichten von jeder Kampagne eines Projekts vom 19. Juli 2019 bis zum 26. Juli 2019 übermittelt wurden:

```
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
```

Wobei gilt:

- `pinpoint.us-east-1.amazonaws.com` ist der Amazon-Pinpoint-Endpunkt für die AWS-Region, die das Projekt hostet.
- `12345678901234567890123456789012345example` ist der eindeutige Bezeichner für das Projekt, das den Kampagnen zugeordnet ist.
- `unique-deliveries-grouped-by-campaign` ist der `kpi-name`-Wert für die Endpunktzustellungen, gruppiert nach Kampagne-Anwendungsmetrik. Dies ist die Metrik, die die Anzahl der eindeutigen Endpunkte zurückgibt, an die Nachrichten von jeder Kampagne übermittelt wurden.
- `2019-07-19T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-07-26T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

AWS CLI

Verwenden Sie den get-application-date-range-kpi-Befehl und geben Sie die entsprechenden Werte für die erforderlichen Parameter an, um Analysedaten für mehrere Kampagnen mithilfe der AWS CLI abzufragen:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Wobei gilt:

- *`application-id`* ist der eindeutige Bezeichner für das Projekt, das den Kampagnen zugeordnet ist.
- *`kpi-name`* ist der `kpi-name`-Wert für die abzufragende Metrik.

Um einen Filter anzuwenden, der die Daten für einen bestimmten Zeitraum abruft, schließen Sie die `start-time`- und `end-time`-Parameter und -Werte in die Abfrage ein. Mithilfe dieser

Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Die folgende Anforderung ruft beispielsweise die Anzahl der eindeutigen Endpunkte ab, an die Nachrichten von jeder Kampagne eines Projekts vom 19. Juli 2019 bis zum 26. Juli 2019 übermittelt wurden:

```
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
```

Wobei gilt:

- `12345678901234567890123456example` ist der eindeutige Bezeichner für das Projekt, das der Kampagne zugeordnet ist.
- `unique-deliveries-grouped-by-campaign` ist der `kpi-name`-Wert für die Endpunktzustellungen, gruppiert nach Kampagne-Anwendungsmetrik. Dies ist die Metrik, die die Anzahl der eindeutigen Endpunkte zurückgibt, an die Nachrichten von jeder Kampagne übermittelt wurden.
- `2019-07-19T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-07-26T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

SDK for Java

Um Analysedaten für mehrere Kampagnen unter Verwendung der AWS SDK for Java abzufragen, verwenden Sie die GetApplicationDateRangeKpiRequest-Methode der Application Metrics-API. Geben Sie die entsprechenden Werte für die erforderlichen Parameter an:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Wobei gilt:

- *applicationId* ist der eindeutige Bezeichner für das Projekt, das den Kampagnen zugeordnet ist.
- *kpiName* ist der `kpi-name`-Wert für die abzufragende Metrik.

Um einen Filter anzuwenden, der die Daten für einen bestimmten Zeitraum abruft, schließen Sie die `startTime`- und `endTime`-Parameter und -Werte in die Abfrage ein. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Die folgende Anforderung ruft beispielsweise die Anzahl der eindeutigen Endpunkte ab, an die Nachrichten von jeder Kampagne eines Projekts vom 19. Juli 2019 bis zum 26. Juli 2019 übermittelt wurden:

```
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")));
```

Wobei gilt:

- `12345678901234567890123456example` ist der eindeutige Bezeichner für das Projekt, das den Kampagnen zugeordnet ist.
- `unique-deliveries-grouped-by-campaign` ist der `kpi-name`-Wert für die Endpunktzustellungen, gruppiert nach Kampagne-Anwendungsmetrik. Dies ist die Metrik, die die Anzahl der eindeutigen Endpunkte zurückgibt, an die Nachrichten von jeder Kampagne übermittelt wurden.
- `2019-07-19T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-07-26T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

Nachdem Sie Ihre Abfrage gesendet haben, gibt Amazon Pinpoint die Abfrageergebnisse in einer JSON-Antwort zurück. Die Struktur der Ergebnisse hängt von der Metrik ab, die Sie abgefragt haben. Einige Metriken geben nur einen Wert zurück. Andere Metriken geben mehrere Werte zurück, und diese Werte werden nach einem relevanten Feld gruppiert. Wenn eine Metrik mehrere Werte

zurückgibt, enthält die JSON-Antwort ein Feld, das angibt, welches Feld zum Gruppieren der Daten verwendet wurde.

Beispielsweise gibt die Anwendungsmetrik Endpunkt-Zustellungen, nach Kampagnen gruppiert (`unique-deliveries-grouped-by-campaign`), die in den vorherigen Beispielen verwendet wurde, mehrere Werte zurück – die Anzahl der eindeutigen Endpunkte, an die Nachrichten gesendet wurden, für jede einem Projekt zugeordnete Kampagne. In diesem Fall lautet die JSON-Antwort wie folgt:


```
{
    "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 diesem Fall gibt das `GroupedBys`-Feld an, dass die Werte nach Kampagnen-ID (`CampaignId`) gruppiert sind.

Weitere Informationen zur Struktur von Abfrageergebnissen finden Sie unter Verwenden von Abfrageergebnissen.

Abfragen von Amazon-Pinpoint-Analysedaten für Transaktionsnachrichten

Zusätzlich zur Verwendung der Analyseseiten auf der Amazon-Pinpoint-Konsole können Sie Amazon-Pinpoint-Analyse-APIs verwenden, um Analysedaten für eine Teilmenge von Standardmetriken abzufragen, die Einblicke in Übermittlungs- und Interaktionstrends für die Transaktionsnachrichten liefern, die für ein Projekt gesendet wurden.

Jede dieser Metriken ist ein messbarer Wert, auch als Key Performance Indicator (KPI) bezeichnet, der Ihnen helfen kann, die Leistung von transaktionalen Nachrichten zu überwachen und zu bewerten. Beispielsweise können Sie mit einer Metrik herausfinden, wie viele Transaktions-E-Mail- oder SMS-Nachrichten Sie gesendet haben oder wie viele dieser Nachrichten an Empfänger zugestellt wurden. Amazon Pinpoint sammelt und aggregiert diese Daten automatisch für alle Transaktions-E-Mail- und SMS-Nachrichten, die Sie für ein Projekt senden. Der Service speichert die Daten für 90 Tage.

Wenn Sie Amazon-Pinpoint-Analyse-APIs zum Abfragen von Daten verwenden, können Sie verschiedene Optionen auswählen, die den Bereich, die Daten, die Gruppierung und die Filter für Ihre Abfrage definieren. Dazu verwenden Sie Parameter, die das Projekt und die Metrik angeben, die Sie abfragen möchten, zusätzlich zu allen datumsbasierten Filtern, die Sie anwenden möchten.

In diesem Thema wird erläutert und es werden Beispiele aufgeführt, wie Sie diese Optionen auswählen und Transaktions-Messaging-Daten für ein Projekt abfragen.

Voraussetzungen

Bevor Sie Analysedaten für Transaktionsmeldungen abfragen, ist es sinnvoll, die folgenden Informationen zu sammeln, die Sie zur Definition Ihrer Abfrage verwenden:

- Projekt-ID: Der eindeutige Bezeichner für das Projekt, von dem die Nachrichten gesendet wurden. In der Amazon-Pinpoint-API wird dieser Wert in der `application-id`-Eigenschaft gespeichert. Auf der Amazon-Pinpoint-Konsole wird dieser Wert als Projekt-ID auf der Seite Alle Projekte angezeigt.
- Datumsbereich: Optional das erste und letzte Datum und die Uhrzeit des Datumsbereichs, für den die Daten abgefragt werden sollen. Datumsbereiche werden inklusiv angegeben und dürfen maximal 31 Kalendertage umfassen. Darüber hinaus müssen sie vor 90 Tagen ab dem aktuellen Tag beginnen. Wenn Sie keinen Datumsbereich angeben, fragt Amazon Pinpoint die Daten für die letzten 31 Kalendertage automatisch ab.
- Metrik: Der Name der abzufragenden Metrik, genauer gesagt der `kpi-name`-Wert für die Metrik. Eine vollständige Liste der unterstützten Metriken und den `kpi-name`-Wert für die einzelnen Metriken finden Sie unter Standardmetriken.

Es hilft auch zu bestimmen, ob Sie die Daten nach einem relevanten Feld gruppieren möchten. In diesem Fall können Sie die Analyse und Berichterstellung vereinfachen, indem Sie eine Metrik auswählen, mit der Daten automatisch gruppiert werden sollen. Amazon Pinpoint stellt beispielsweise mehrere Standardmetriken bereit, die die Anzahl der Transaktions-SMS-Nachrichten melden,

die an Empfänger gesendet wurden. Eine dieser Metriken gruppiert die Daten automatisch nach Datum (`txn-sms-delivered-grouped-by-date`). Eine weitere Metrik gruppiert die Daten automatisch nach Land oder Region (`txn-sms-delivered-grouped-by-country`). Eine dritte Metrik gibt einfach einen einzigen Wert zurück – die Anzahl der Meldungen, die an die Empfänger zugestellt wurden (`txn-sms-delivered`). Wenn Sie keine Standardmetrik finden können, die Daten so gruppiert, wie Sie es wünschen, können Sie eine Reihe von Abfragen entwickeln, die die gewünschten Daten zurückgeben. Sie können dann die Abfrageergebnisse manuell aufschlüsseln oder in benutzerdefinierte Gruppen zusammenfassen, die Sie entwerfen.

Schließlich ist es wichtig zu überprüfen, ob Sie berechtigt sind, auf die Daten zuzugreifen, die Sie abfragen möchten. Weitere Informationen finden Sie unter IAM-Richtlinien zum Abfragen von Amazon-Pinpoint-Analysedaten.

Abfragen von Daten für Transaktions-E-Mail-Nachrichten

Um die Daten für Transaktions-E-Mail-Nachrichten abzufragen, die für ein Projekt gesendet wurden, verwenden Sie die Anwendungsmetriken-API und geben Werte für die folgenden erforderlichen Parameter an:

- application-id: Die Projekt-ID, bei der es sich um den eindeutigen Bezeichner für das Projekt handelt. In Amazon Pinpoint haben die Begriffe Projekt und Anwendung dieselbe Bedeutung.
- kpi-name: Der Name der abzufragenden Metrik. Dieser Wert beschreibt die zugeordnete Metrik und besteht aus zwei oder mehr Begriffen, die aus alphanumerischen Kleinbuchstaben bestehen, die durch einen Bindestrich getrennt sind. Eine vollständige Liste der unterstützten Metriken und den `kpi-name`-Wert für die einzelnen Metriken finden Sie unter Standardmetriken.

Sie können auch einen Filter anwenden, der die Daten für einen bestimmten Zeitraum abfragt. Wenn Sie keinen Datumsbereich angeben, werden die Daten für die letzten 31 Kalendertage von Amazon Pinpoint zurückgegeben. Um die Daten nach verschiedenen Datumsangaben zu filtern, verwenden Sie die unterstützten Datumsbereichsparameter, um das erste und letzte Datum und die Uhrzeit des Datumsbereichs anzugeben. Die Werte sollten im erweiterten ISO 8601-Format vorliegen und die koordinierte Weltzeit (UTC) verwenden, z. B. `2019-09-06T20:00:00Z` für 20.00 Uhr UTC am 06. September 2019. Datumsbereiche werden inklusiv angegeben und dürfen maximal 31 Kalendertage umfassen. Darüber hinaus müssen das erste Datum und die erste Uhrzeit früher als 90 Tage ab dem aktuellen Tag liegen.

In den folgenden Beispielen wird gezeigt, wie Analysedaten für Transaktions-E-Mail-Nachrichten mithilfe der Amazon-Pinpoint-REST-API, der AWS CLI und des AWS SDK for Java abgefragt werden.

Sie können jedes unterstützte AWS SDK verwenden, um Analysedaten für Transaktionsnachrichten abzufragen. Die AWS CLI-Beispiele sind für Microsoft Windows formatiert. Ersetzen Sie für Unix, Linux und macOS das Zeilenfortsetzungszeichen (^) durch einen umgekehrten Schrägstrich (\).

REST API

Um Analysedaten für Transaktions-E-Mail-Nachrichten mithilfe der Amazon-Pinpoint-REST-API abzufragen, senden Sie eine HTTP(S)-GET-Anforderung an den Anwendungsmetriken-URI. Geben Sie im URI die entsprechenden Werte für die erforderlichen Pfadparameter an:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Wobei gilt:

- *Endpunkt* ist der Amazon-Pinpoint-Endpunkt für die AWS-Region, die das Projekt hostet.
- *application-id* ist der eindeutige Bezeichner für das Projekt.
- *kpi-name* ist der `kpi-name`-Wert für die abzufragende Metrik.

Alle Parameter sollten URL-codiert sein.

Wenn Sie einen Filter anwenden möchten, der die Daten für einen bestimmten Datumsbereich abfragt, hängen Sie die `start-time`- und `end-time`-Abfrageparameter und -werte an den URI an. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Verwenden Sie ein kaufmännisches Und-Zeichen (&), um die Parameter zu trennen.

Die folgende Anforderung ruft beispielsweise die Anzahl der Transaktions-E-Mail-Nachrichten ab, die für ein Projekt vom 6. September 2019 bis zum 13. September 2019 gesendet wurden:

```
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
```

Wobei gilt:

- `pinpoint.us-east-1.amazonaws.com` ist der Amazon-Pinpoint-Endpunkt für die AWS-Region, die das Projekt hostet.

- `12345678901234567890123456example` ist der eindeutige Bezeichner für das Projekt.
- `txn-emails-sent` ist der `kpi-name`-Wert für die Sendevorgänge-Anwendungsmetrik, d. h. die Metrik, die die Anzahl der Transaktions-E-Mail-Nachrichten angibt, die für ein Projekt gesendet wurden.
- `2019-09-06T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-09-13T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

AWS CLI

Um Analysedaten für transaktionale E-Mail-Nachrichten mit dem Befehl AWS CLI abzufragen, verwenden Sie den Befehl get-application-date-range-kpi und geben Sie die entsprechenden Werte für die erforderlichen Parameter an:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Wobei gilt:

- *`application-id`* ist der eindeutige Bezeichner für das Projekt.
- *`kpi-name`* ist der `kpi-name`-Wert für die abzufragende Metrik.

Um einen Filter anzuwenden, der die Daten für einen bestimmten Zeitraum abfragt, fügen Sie der Abfrage die `start-time`- und `end-time`-Parameter und -Werte hinzu. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Die folgende Anforderung ruft beispielsweise die Anzahl der Transaktions-E-Mail-Nachrichten ab, die für ein Projekt vom 6. September 2019 bis zum 13. September 2019 gesendet wurden:

```
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
```

Wobei gilt:

- `12345678901234567890123example` ist der eindeutige Bezeichner für das Projekt.
- `txn-emails-sent` ist der `kpi-name`-Wert für die Sendevorgänge-Anwendungsmetrik, d. h. die Metrik, die die Anzahl der Transaktions-E-Mail-Nachrichten angibt, die für ein Projekt gesendet wurden.
- `2019-09-06T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-09-13T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

SDK for Java

Um Analysedaten für transaktionale E-Mail-Nachrichten unter Verwendung der AWS SDK for Java abzufragen, verwenden Sie die Methode GetApplicationDateRangeKpiRequest der Application Metrics-API. Geben Sie die entsprechenden Werte für die erforderlichen Parameter an:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Wobei gilt:

- *`applicationId`* ist der eindeutige Bezeichner für das Projekt.
- *`kpiName`* ist der `kpi-name`-Wert für die abzufragende Metrik.

Um einen Filter anzuwenden, der die Daten für einen bestimmten Datumsbereich abfragt, schließen Sie die `startTime`- und `endTime`-Parameter und -Werte in die Abfrage ein. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Die folgende Anforderung ruft beispielsweise die Anzahl der Transaktions-E-Mail-Nachrichten ab, die für ein Projekt vom 6. September 2019 bis zum 13. September 2019 gesendet wurden:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123example")
        .withKpiName("txn-emails-sent")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
```

```
        .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

Wobei gilt:

- `12345678901234567890123example` ist der eindeutige Bezeichner für das Projekt.
- `txn-emails-sent` ist der `kpi-name`-Wert für die Sendevorgänge-Anwendungsmetrik, d. h. die Metrik, die die Anzahl der Transaktions-E-Mail-Nachrichten angibt, die für ein Projekt gesendet wurden.
- `2019-09-06T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-09-13T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

Nachdem Sie Ihre Abfrage gesendet haben, gibt Amazon Pinpoint die Abfrageergebnisse in einer JSON-Antwort zurück. Die Struktur der Ergebnisse hängt von der Metrik ab, die Sie abgefragt haben. Einige Metriken geben nur einen Wert zurück. Beispielsweise gibt die Anwendungsmetrik Sendungen (`txn-emails-sent`), die in den vorherigen Beispielen verwendet wurde, einen Wert zurück – die Anzahl der Transaktions-E-Mail-Nachrichten, die von einem Projekt gesendet wurden. In diesem Fall lautet die JSON-Antwort wie folgt:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123example",
        "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"
```

```
    }
}
```

Andere Metriken geben mehrere Werte zurück und gruppieren die Werte nach einem relevanten Feld. Wenn eine Metrik mehrere Werte zurückgibt, enthält die JSON-Antwort ein Feld, das angibt, welches Feld zum Gruppieren der Daten verwendet wurde.

Weitere Informationen zur Struktur von Abfrageergebnissen finden Sie unter Verwenden von Abfrageergebnissen.

Abfragen von Daten für Transaktions-SMS-Nachrichten

Um die Daten für Transaktions-SMS-Nachrichten abzufragen, die für ein Projekt gesendet wurden, verwenden Sie die Anwendungsmetriken-API und geben Werte für die folgenden erforderlichen Parameter an:

- application-id: Die Projekt-ID, bei der es sich um den eindeutigen Bezeichner für das Projekt handelt. In Amazon Pinpoint haben die Begriffe Projekt und Anwendung dieselbe Bedeutung.
- kpi-name: Der Name der abzufragenden Metrik. Dieser Wert beschreibt die zugeordnete Metrik und besteht aus zwei oder mehr Begriffen, die aus alphanumerischen Kleinbuchstaben bestehen, die durch einen Bindestrich getrennt sind. Eine vollständige Liste der unterstützten Metriken und den `kpi-name`-Wert für die einzelnen Metriken finden Sie unter Standardmetriken.

Sie können auch einen Filter anwenden, der die Daten für einen bestimmten Zeitraum abfragt. Wenn Sie keinen Datumsbereich angeben, werden die Daten für die letzten 31 Kalendertage von Amazon Pinpoint zurückgegeben. Um die Daten nach verschiedenen Daten zu filtern, verwenden Sie die unterstützten Datumsbereichsparameter, um das erste Datum und die erste Uhrzeit sowie das letzte Datum und die letzte Uhrzeit des Datumsbereichs anzugeben. Die Werte sollten im erweiterten ISO 8601-Format vorliegen und die koordinierte Weltzeit (UTC) verwenden, z. B. `2019-09-06T20:00:00Z` für 20.00 Uhr UTC am 06. September 2019. Datumsbereiche werden inklusiv angegeben und dürfen maximal 31 Kalendertage umfassen. Darüber hinaus müssen das erste Datum und die erste Uhrzeit früher als 90 Tage ab dem aktuellen Tag liegen.

In den folgenden Beispielen wird gezeigt, wie Analysedaten für Transaktions-SMS-Nachrichten mithilfe der Amazon-Pinpoint-REST-API, der AWS CLI und des AWS SDK for Java abgefragt werden. Sie können jedes unterstützte AWS SDK verwenden, um Analysedaten für Transaktionsnachrichten abzufragen. Die AWS CLI-Beispiele sind für Microsoft Windows formatiert. Ersetzen Sie für Unix, Linux und macOS das Zeilenfortsetzungszeichen (^) durch einen umgekehrten Schrägstrich (\).

REST API

Um Analysedaten für Transaktions-SMS-Nachrichten mithilfe der Amazon-Pinpoint-REST-API abzufragen, senden Sie eine HTTP(S)-GET-Anforderung an den Anwendungsmetriken-URI. Geben Sie im URI die entsprechenden Werte für die erforderlichen Pfadparameter an:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Wobei gilt:

- *Endpunkt* ist der Amazon-Pinpoint-Endpunkt für die AWS-Region, die das Projekt hostet.
- *application-id* ist der eindeutige Bezeichner für das Projekt.
- *kpi-name* ist der `kpi-name`-Wert für die abzufragende Metrik.

Alle Parameter sollten URL-codiert sein.

Wenn Sie einen Filter anwenden möchten, der die Daten für einen bestimmten Datumsbereich abruft, hängen Sie die `start-time`- und `end-time`-Parameter und -Werte an den URI an. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Verwenden Sie ein kaufmännisches Und-Zeichen (&), um die Parameter zu trennen.

Die folgende Anforderung ruft beispielsweise die Anzahl der Transaktions-SMS-Nachrichten ab, die jeden Tag vom 6. September 2019 bis zum 8. September 2019 gesendet wurden:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123example/
kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-08T23:59:59Z
```

Wobei gilt:

- `pinpoint.us-east-1.amazonaws.com` ist der Amazon-Pinpoint-Endpunkt für die AWS-Region, die das Projekt hostet.
- `12345678901234567890123example` ist der eindeutige Bezeichner für das Projekt.
- `txn-sms-sent-grouped-by-date` ist der `kpi-name`-Wert für die Sendevorgänge, gruppiert nach Datum-Anwendungsmetrik. Dabei handelt es sich um die Metrik, die die Anzahl der Transaktions-SMS-Nachrichten zurückgibt, die an jedem Tag des Datumsbereichs gesendet wurden.

- `2019-09-06T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-09-08T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

AWS CLI

Um Analysedaten für transaktionale SMS-Nachrichten mit dem Befehl AWS CLI abzufragen, verwenden Sie den Befehl get-application-date-range-kpi und geben Sie die entsprechenden Werte für die erforderlichen Parameter an:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Wobei gilt:

- *`application-id`* ist der eindeutige Bezeichner für das Projekt.
- *`kpi-name`* ist der `kpi-name`-Wert für die abzufragende Metrik.

Um einen Filter anzuwenden, der die Daten für einen bestimmten Zeitraum abruft, schließen Sie die `start-time`- und `end-time`-Parameter und -Werte in die Abfrage ein. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Die folgende Anforderung ruft beispielsweise die Anzahl der Transaktions-SMS-Nachrichten ab, die jeden Tag vom 6. September 2019 bis zum 8. September 2019 gesendet wurden:

```
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
```

Wobei gilt:

- `12345678901234567890123456example` ist der eindeutige Bezeichner für das Projekt.
- `txn-sms-sent-grouped-by-date` ist der `kpi-name`-Wert für die Sendevorgänge, gruppiert nach Datum-Anwendungsmetrik. Dabei handelt es sich um die Metrik, die die Anzahl der

Transaktions-SMS-Nachrichten zurückgibt, die an jedem Tag des Datumsbereichs gesendet wurden.

- `2019-09-06T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-09-08T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

SDK for Java

Um Analysedaten für transaktionale SMS-Nachrichten unter Verwendung der AWS SDK for Java abzufragen, verwenden Sie die GetApplicationDateRangeKpiRequest-Methode der Application Metrics-API und geben Sie die entsprechenden Werte für die erforderlichen Parameter an:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Wobei gilt:

- *`applicationId`* ist der eindeutige Bezeichner für das Projekt.
- *`kpiName`* ist der `kpi-name`-Wert für die abzufragende Metrik.

Um einen Filter anzuwenden, der die Daten für einen bestimmten Zeitraum abruft, schließen Sie die `startTime`- und `endTime`-Parameter und -Werte in die Abfrage ein. Mithilfe dieser Parameter können Sie das erste und letzte Datum und die Uhrzeit im erweiterten ISO 8601-Format eines Inklusivdatumsbereichs angeben, für den die Daten abgerufen werden sollen. Die folgende Anforderung ruft beispielsweise die Anzahl der Transaktions-SMS-Nachrichten ab, die jeden Tag vom 6. September 2019 bis zum 8. September 2019 gesendet wurden:

```
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")));
```

Wobei gilt:

- `12345678901234567890123456example` ist der eindeutige Bezeichner für das Projekt.

- `txn-sms-sent-grouped-by-date` ist der `kpi-name`-Wert für die Sendevorgänge, gruppiert nach Datum-Anwendungsmetrik. Dabei handelt es sich um die Metrik, die die Anzahl der Transaktions-SMS-Nachrichten zurückgibt, die an jedem Tag des Datumsbereichs gesendet wurden.
- `2019-09-06T00:00:00Z` ist das erste Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.
- `2019-09-08T23:59:59Z` ist das letzte Datum und die Uhrzeit, für die Daten abgerufen werden sollen, als Teil eines einschließenden Datumsbereichs.

Nachdem Sie Ihre Abfrage gesendet haben, gibt Amazon Pinpoint die Abfrageergebnisse in einer JSON-Antwort zurück. Die Struktur der Ergebnisse hängt von der Metrik ab, die Sie abgefragt haben. Einige Metriken geben nur einen Wert zurück. Andere Metriken geben mehrere Werte zurück und gruppieren diese Werte nach einem relevanten Feld. Wenn eine Metrik mehrere Werte zurückgibt, enthält die JSON-Antwort ein Feld, das angibt, welches Feld zum Gruppieren der Daten verwendet wurde.

Beispielsweise gibt die Anwendungsmetrik Sends, gruppiert nach Datum (`txn-sms-sent-grouped-by-date`), die in den vorherigen Beispielen verwendet wurde, mehrere Werte zurück — die Anzahl der Transaktions-SMS-Nachrichten, die an jedem Tag innerhalb des angegebenen Datumsbereichs gesendet wurden. In diesem Fall lautet die JSON-Antwort wie folgt:

```
{
    "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 diesem Fall gibt das `GroupedBys`-Feld an, dass die Werte nach Kalendertag (`Date`) gruppiert sind. Dies bedeutet, dass:

- 29 Nachrichten am 6. September 2019 gesendet wurden.
- 35 Nachrichten am 7. September 2019 gesendet wurden.
- 10 Nachrichten am 8. September 2019 gesendet wurden.

Weitere Informationen zur Struktur von Abfrageergebnissen finden Sie unter Verwenden von Abfrageergebnissen.

Verwenden von Abfrageergebnissen von Amazon Pinpoint Analytics

Wenn Sie Amazon-Pinpoint-Analyse-APIs zum Abfragen von Analysedaten verwenden, gibt Amazon Pinpoint die Ergebnisse in einer JSON-Antwort zurück. Bei Anwendungsmetriken, Kampagnenmetriken und Journey-Engagement-Metriken entsprechen die Daten in der Antwort einem Standard JSON-Schema für das Reporting von Amazon-Pinpoint-Analysedaten.

Das bedeutet, dass Sie mit der Programmiersprache oder dem Tool Ihrer Wahl eine benutzerdefinierte Lösung implementieren können, die die Daten für eine oder mehrere dieser Metriken abfragt, die Ergebnisse jeder Abfrage erfasst und die Ergebnisse dann in eine Tabelle, ein Objekt oder einen anderen Speicherort schreibt. Sie können dann mit den Abfrageergebnissen an diesem Speicherort arbeiten, indem Sie einen anderen Service oder eine andere Anwendung verwenden.

Beispielsweise ist Folgendes möglich:

- Ein benutzerdefiniertes Dashboard erstellen, das regelmäßig eine Reihe von Metriken abfragt und die Ergebnisse mithilfe des bevorzugten Datenvisualisierungsrahmens anzeigt.
- Erstellen Sie einen Bericht, der die Interaktionsraten verfolgt, indem Sie die entsprechenden Metriken abfragen und die Ergebnisse in einem Diagramm oder einem anderen Berichtstyp anzeigen, den Sie entwerfen.
- Analysedaten analysieren und in ein bestimmtes Speicherformat schreiben und anschließend die Ergebnisse in eine Langzeitspeicherlösung portieren.

Beachten Sie, dass Amazon-Pinpoint-Analyse-APIs nicht dazu gedacht sind, persistente Objekte zu erstellen oder zu speichern, die Sie anschließend in einem Amazon-Pinpoint-Projekt oder Ihrem Amazon-Pinpoint-Konto lesen oder verwenden können. Stattdessen sollen die APIs Ihnen dabei helfen, Analysedaten abzurufen und diese Daten an andere Services und Anwendungen zur weiteren Analyse, Speicherung oder Berichterstellung zu übertragen. Sie tun dies zum Teil, indem sie dieselbe JSON-Antwortstruktur und dasselbe Schema für alle Analysedaten verwenden, die Sie programmgesteuert nach Anwendungsmetriken, Kampagnenmetriken und Journey-Engagement-Metriken abfragen können.

In diesem Thema werden die Struktur, die Objekte und die Felder in einer JSON-Antwort auf eine Anfrage nach einer Anwendungsmetrik, einer Kampagnenmetrik oder einer Journey-Engagement-Metrik erläutert. Informationen zu den Feldern in einer JSON-Antwort auf eine Abfrage für eine Journey-Ausführungs- oder Journey-Aktivitätsausführungsmetrik finden Sie unter Standardmetriken von Amazon Pinpoint Analytics.

JSON-Struktur

Um Ihnen beim Analysieren und Verwenden von Abfrageergebnissen zu helfen, verwenden die Amazon-Pinpoint-Analytics-APIs die gleiche JSON-Antwortstruktur für alle Amazon-Pinpoint-Analysedaten, die Sie programmgesteuert für Anwendungs-, Kampagnen- und Journey-Engagement-Metriken abfragen können. Jede JSON-Antwort gibt die Werte zur Definition der Abfrage an, wie beispielsweise die Projekt-ID (`ApplicationId`), an. Die Antwort beinhaltet ein (nur ein einziges) `KpiResult`-Objekt. Das `KpiResult`-Objekt enthält die Gesamtergebnismenge für eine Abfrage.

Jedes `KpiResult`-Objekt enthält ein `Rows`-Objekt. Dies ist ein Array von Objekten, die die Abfrageergebnisse und relevante Metadaten zu den Werten in diesen Ergebnissen enthalten. Struktur und Inhalt eines `Rows`-Objekts weisen folgende allgemeine Merkmale auf:

- Jede Zeile mit Abfrageergebnissen ist ein separates JSON-Objekt mit dem Namen `Values` im `Rows`-Objekt. Wenn eine Abfrage beispielsweise drei Werte zurückgibt, enthält das `Rows`-Objekt drei `Values`-Objekte. Jedes `Values`-Objekt enthält ein individuelles Ergebnis für die Abfrage.
- Jede Spalte mit Abfrageergebnissen ist eine Eigenschaft des `Values`-Objekts, auf das sie angewendet wird. Der Name der Spalte wird im `Key`-Feld des `Values`-Objekts gespeichert.
- Für gruppierte Abfrageergebnisse verfügt jedes `Values`-Objekt über ein zugeordnetes `GroupedBys`-Objekt. Das `GroupedBys`-Objekt gibt an, in welchem Feld die Ergebnisse gruppiert wurden. Es liefert auch den Gruppierungswert für das zugehörige `Values`-Objekt.
- Wenn die Abfrageergebnisse für eine Metrik null sind, ist das `Rows`-Objekt leer.

Neben diesen allgemeinen Merkmalen variieren Struktur und Inhalt des `Rows`-Objekts je nach Metrik. Dies liegt daran, dass zwei Arten von Metriken von Amazon Pinpoint unterstützt werden:einwertige Metriken und mehrwertige Metriken.

Eine Einzelwertmetrik liefert nur einen kumulativen Wert. Ein Beispiel ist der Prozentsatz der Nachrichten, die von allen Läufen einer Kampagne an die Empfänger zugestellt wurden. Eine Mehrwertmetrik liefert mehr als einen Wert und gruppiert diese Werte nach einem relevanten Feld. Ein Beispiel ist der Prozentsatz der Nachrichten, die für jeden Lauf einer Kampagne an die Empfänger zugestellt wurden, gruppiert nach Kampagnenlauf.

Sie können schnell feststellen, ob es sich bei einer Metrik um eine einwertige oder eine mehrwertige Metrik handelt, indem Sie Bezug auf den Namen der Metrik nehmen. Wenn `grouped-by` im Namen nicht enthalten ist, handelt es sich um eine einwertige Metrik. Falls doch, handelt es sich um eine mehrwertige Metrik. Eine vollständige Liste der Metriken, die Sie programmgesteuert abfragen können, finden Sie unter Standardmetriken von Amazon Pinpoint Analytics.

Einzelwert-Metriken

Bei einer einwertigen Metrik enthält das `Rows`-Objekt ein `Values`-Objekt, das:

- Den Anzeigenamen der Metrik angibt, die abgefragt wurde.
- Den Wert für die abgefragte Metrik angibt.
- Den Datentyp des zurückgegebenen Wertes angibt.

Die folgende JSON-Antwort enthält beispielsweise die Abfrageergebnisse für eine Einzelwertmetrik. Diese Metrik gibt die Anzahl der eindeutigen Endpunkte an, an die Nachrichten von allen Kampagnen, die mit einem Projekt verbunden sind, vom 1. August 2019 bis zum 31. August 2019 gesendet wurden:

```
{
    "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 diesem Beispiel zeigt die Antwort, dass alle Kampagnen des Projekts vom 1. August 2019 bis 31. August 2019 Nachrichten an 1.368 eindeutige Endpunkte lieferten, wobei folgendes gilt:

- `Key` ist der Anzeigename der Metrik, deren Wert im `Value`-Feld (`UniqueDeliveries`) angegeben ist.
- `Type` ist der Datentyp des im `Value`-Feld (`Double`) angegebenen Wertes.
- `Value` ist der tatsächliche Wert für die Metrik, die abgefragt wurde, einschließlich aller angewendeten Filter (`1368.0`).

Wenn die Abfrageergebnisse für eine einwertige Metrik Null sind (nicht größer oder gleich Null), ist das `Rows`-Objekt leer. Amazon Pinpoint gibt einen Nullwert für eine Metrik zurück, wenn keine Daten für die Metrik zurückgegeben werden können. Beispiele:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"234567890123456789012345example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[

            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Mehrwertige Metriken

Struktur und Inhalt des `Rows`-Objekts für eine mehrwertige Metrik sind meist identisch mit einer einwertigen Metrik. Das `Rows`-Objekt für eine Mehrwertmetrik enthält auch ein `Values`-Objekt. Das `Values`-Objekt gibt den Anzeigenamen der abgefragten Metrik an, liefert den Wert für diese Metrik und identifiziert den Datentyp dieses Wertes.

Das `Rows`-Objekt für eine mehrwertige Metrik enthält jedoch auch ein oder mehrere `GroupedBy`-Objekte. Es gibt ein `GroupedBy`-Objekt für jedes `Values`-Objekt in den Abfrageergebnissen. Das `GroupedBy`-Objekt gibt an, in welchem Feld die Daten in den Ergebnissen gruppiert wurden und den Datentyp dieses Feldes. Es zeigt auch den Gruppierungswert für dieses Feld an (für das zugehörige `Values`-Objekt).

Die folgende JSON-Antwort enthält beispielsweise die Abfrageergebnisse für eine mehrwertige Metrik, die die Anzahl der eindeutigen Endpunkte angibt, an die Nachrichten gesendet wurden, für jede Kampagne, die einem Projekt zugeordnet ist, vom 1. August 2019 bis zum 31. August 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 diesem Beispiel zeigt die Antwort an, dass drei der Kampagnen des Projekts vom 1. August 2019 bis zum 31. August 2019 Nachrichten an eindeutige Endpunkte übermittelt haben. Für jede dieser Kampagnen lautet die Aufschlüsselung der Zustellanzahl:

- Kampagne `80b8efd84042ff8d9c96ce2f8example` hat Nachrichten an 123 eindeutige Endpunkte übermittelt.

- Kampagne `810c7aab86d42fb2b56c8c966example` hat Nachrichten an 456 eindeutige Endpunkte übermittelt.
- Kampagne `42d8c7eb0990a57ba1d5476a3example` hat Nachrichten an 789 eindeutige Endpunkte übermittelt.

Wobei dies die allgemeine Struktur der Objekte und Felder ist:

- `GroupedBys.Key`: Der Name der Eigenschaft oder des Feldes, in dem der im `GroupedBys.Value`-Feld (`CampaignId`) angegebene Gruppierungswert gespeichert wird.
- `GroupedBys.Type`: Der Datentyp des im `GroupedBys.Value`-Feld (`String`) angegebenen Wertes.
- `GroupedBys.Value`: Der tatsächliche Wert für das Feld, das zur Gruppierung der Daten verwendet wurde, wie im `GroupedBys.Key`-Feld angegeben (Kampagnen-ID).
- `Values.Key`: Der Anzeigename der Metrik, deren Wert im `Values.Value`-Feld (`UniqueDeliveries`) angegeben ist.
- `Values.Type`: Der Datentyp des im `Values.Value`-Feld (`Double`) angegebenen Wertes.
- `Values.Value`: Der tatsächliche Wert für die abgefragte Metrik, einschließlich aller angewendeten Filter.

Wenn die Abfrageergebnisse für eine mehrwertige Metrik für ein bestimmtes Projekt, eine Kampagne oder eine andere Ressource null (nicht größer als oder gleich Null) sind, werden von Amazon Pinpoint keine Objekte oder Felder für die Ressource zurückgegeben. Wenn die Abfrageergebnisse für eine mehrwertige Metrik für alle Ressourcen null sind, wird ein leeres `Rows`-Objekt von Amazon Pinpoint zurückgegeben.

JSON-Objekte und -Felder

Zusätzlich zur Angabe der Werte, die eine Abfrage definiert haben, wie z. B. die Projekt-ID (`ApplicationId`), beinhaltet jede JSON-Antwort auf eine Abfrage für eine Anwendungsmetrik, einer Kampagnenmetrik oder einer Journey-Engagement-Metrik ein `KpiResult`-Objekt. Dieses Objekt enthält die Gesamtergebnismenge für eine Abfrage, die Sie analysieren können, um Analysedaten an einen anderen Service oder eine andere Anwendung zu senden. Jedes `KpiResult`-Objekt enthält einige oder alle der folgenden Standardobjekte und -felder, abhängig von der Metrik.

Objekt oder Feld	Beschreibung
`Rows`	Ein Array von Objekten, das die Ergebnism enge für eine Abfrage enthält.
`Rows.GroupedBys`	Bei einer mehrwertigen Metrik ein Array von Feldern, das das Feld und die Werte definiert , die zum Gruppieren von Daten in Abfrageer gebnissen verwendet wurden.
`Rows.GroupedBys.Key`	Bei einer mehrwertigen Metrik der Name der Eigenschaft oder des Feldes, in dem der im `GroupedBys.Value` -Feld angegebene Wert gespeichert wird.
`Rows.GroupedBys.Type`	Bei einer mehrwertigen Metrik der Datentyp des im `GroupedBys.Value` -Feld angegebenen Wertes.
`Rows.GroupedBys.Value`	Bei einer mehrwertigen Metrik der tatsächliche Wert für das Feld, das zum Gruppieren von Daten in Abfrageergebnissen verwendet wurde. Dieser Wert korreliert mit einem zugeordneten `Values`-Objekt.
`Rows.Values`	Ein Array von Feldern, das Abfrageergebnisse enthält.
`Rows.Values.Key`	Der Anzeigename der Metrik, die abgefragt wurde. Der Wert der Metrik wird im Feld `Values.Value` angegeben.
`Rows.Values.Type`	Der Datentyp des im `Values.Value` -Feld angegebenen Wertes.
`Rows.Values.Value`	Der tatsächliche Wert für die abgefragte Metrik, einschließlich aller angewendeten Filter.

Informationen zu den Feldern in einer JSON-Antwort auf eine Abfrage für eine Journey-Ausführungs- oder Journey-Aktivitätsausführungsmetrik finden Sie unter Standardmetriken von Amazon Pinpoint Analytics.

Protokollieren von Amazon Pinpoint API-Aufrufen mit AWS CloudTrail

Amazon Pinpoint ist integriert. Dabei handelt es sich um einen Service AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon Pinpoint ausgeführt wurden. CloudTrail erfasst API-Aufrufe für Amazon Pinpoint als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon-Pinpoint-Konsole und Code-Aufrufe der Amazon-Pinpoint-API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon Simple Storage Service (Amazon S3) -Bucket aktivieren, einschließlich Ereignissen für Amazon Pinpoint. Wenn Sie keinen Trail konfigurieren, können Sie trotzdem die neuesten Ereignisse mithilfe des Ereignisverlaufs auf der CloudTrail Konsole anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon Pinpoint gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen darüber CloudTrail, einschließlich der Konfiguration und Aktivierung, finden Sie im AWS CloudTrail Benutzerhandbuch.

Informationen zu Amazon Pinpoint in CloudTrail

CloudTrail ist für Ihr AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn unterstützte Ereignisaktivitäten in Amazon Pinpoint auftreten, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS Konto, einschließlich Ereignissen für Amazon Pinpoint, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen helfen Ihnen beim Bestimmen der Folgenden Elemente:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail -Element userIdentity.

Sie können einen Trail erstellen und Ihre Protokolldateien beliebig lange in Ihrem Amazon-S3-Bucket speichern. Außerdem können Sie Amazon-S3-Lebenszyklusregeln definieren, um Protokolldateien automatisch zu archivieren oder zu löschen. Standardmäßig werden die Protokolldateien mit serverseitiger Amazon-S3-Verschlüsselung (SSE) verschlüsselt.

Um über die Übermittlung von Protokolldateien informiert zu werden, konfigurieren Sie die Konfiguration so, CloudTrail dass Amazon SNS SNS-Benachrichtigungen veröffentlicht werden, wenn neue Protokolldateien zugestellt werden. Weitere Informationen finden Sie unter Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail.

Sie können auch Amazon Pinpoint Pinpoint-Protokolldateien aus mehreren AWS Regionen und mehreren AWS Konten in einem einzigen Amazon S3 S3-Bucket zusammenfassen. Weitere Informationen finden Sie unter Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten.

Sie können CloudTrail damit Aktionen für die folgenden Amazon Pinpoint Pinpoint-APIs protokollieren:

- Amazon-Pinpoint-API
- Amazon-Pinpoint-SMS- und -Sprachnachrichten-API

Amazon Pinpoint API-Aktionen, die protokolliert werden können von CloudTrail

Die Amazon Pinpoint API unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien:

- CreateApp
- CreateCampaign
- CreateEmailTemplate
- CreateExportJob
- CreateImportJob
- CreateJourney
- CreatePushTemplate
- CreateRecommenderConfiguration
- CreateSegment
- CreateSmsTemplate
- CreateVoiceTemplate
- DeleteAdmChannel
- DeleteApnsChannel
- DeleteApnsSandboxChannel
- DeleteApnsVoipChannel
- DeleteApnsVoipSandboxChannel
- DeleteApp
- DeleteBaiduChannel
- DeleteCampaign
- DeleteEmailChannel
- DeleteEmailTemplate
- DeleteEndpoint
- DeleteEventStream
- DeleteGcmChannel
- DeleteJourney

- DeletePushTemplate
- DeleteRecommenderConfiguration
- DeleteSegment
- DeleteSmsChannel
- DeleteSmsTemplate
- DeleteUserEndpoints
- DeleteVoiceChannel
- DeleteVoiceTemplate
- GetAdmChannel
- GetApnsChannel
- GetApnsSandboxChannel
- GetApnsVoipChannel
- GetApnsVoipSandboxChannel
- GetApp
- GetApplicationDateRangeKpi
- GetApplicationSettings
- GetApps
- GetBaiduChannel
- GetCampaign
- GetCampaignActivities
- GetCampaignDateRangeKpi
- GetCampaignVersion
- GetCampaignVersions
- GetCampaigns
- GetChannels
- GetEmailChannel
- GetEmailTemplate
- GetEndpoint
- GetEventStream

- GetExportJob
- GetExportJobs
- GetGcmChannel
- GetImportJob
- GetImportJobs
- GetJourney
- GetJourneyDateRangeKpi
- GetJourneyExecutionActivityMetrics
- GetJourneyExecutionMetrics
- GetPushTemplate
- GetRecommenderConfiguration
- GetRecommenderConfigurations
- GetSegment
- GetSegmentExportJobs
- GetSegmentImportJobs
- GetSegmentVersion
- GetSegmentVersions
- GetSegments
- GetSmsChannel
- GetSmsTemplate
- GetUserEndpoints
- GetVoiceChannel
- GetVoiceTemplate
- ListJourneys
- ListTagsForResource
- ListTemplates
- ListTemplateVersions
- PhoneNumberValidate
- PutEventStream

- RemoveAttributes
- TagResource
- UntagResource
- UpdateAdmChannel
- UpdateApnsChannel
- UpdateApnsSandboxChannel
- UpdateApnsVoipChannel
- UpdateApnsVoipSandboxChannel
- UpdateApplicationSettings
- UpdateBaiduChannel
- UpdateCampaign
- UpdateEmailChannel
- UpdateEmailTemplate
- UpdateEndpoint
- UpdateEndpointsBatch
- UpdateGcmChannel
- UpdateJourney
- UpdateJourneyState
- UpdatePushTemplate
- UpdateRecommenderConfiguration
- UpdateSegment
- UpdateSmsChannel
- UpdateSmsTemplate
- UpdateTemplateActiveVersion
- UpdateVoiceChannel
- UpdateVoiceTemplate

Die folgenden Amazon Pinpoint API-Aktionen sind nicht angemeldet CloudTrail:

- PutEvents

- SendMessages
- SendUsersMessages

Amazon Pinpoint E-Mail-API-Aktionen, die protokolliert werden können von CloudTrail

Die Amazon Pinpoint Email API unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien:

- 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

Die folgende Amazon Pinpoint Email API-Aktion ist nicht angemeldet CloudTrail:

- SendEmail

Amazon Pinpoint SMS- und Sprach-API-Aktionen, Version 1, die protokolliert werden können von CloudTrail

Die Amazon Pinpoint SMS and Voice Version 1 API unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien:

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- GetConfigurationSetEventDestinations
- UpdateConfigurationSetEventDestination

Die folgende Amazon Pinpoint SMS and Voice Version 1-API-Aktion ist nicht angemeldet CloudTrail:

- SendVoiceMessage

Beispiele: Einträge in der Amazon-Pinpoint-Protokolldatei

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen

oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar. Sie enthalten Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateCampaign` Aktionen `GetCampaigns` und Funktionen der Amazon Pinpoint Pinpoint-API demonstriert.


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

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateConfigurationSetEventDestination` Aktionen `CreateConfigurationSet` und in der Amazon Pinpoint SMS and Voice API demonstriert.

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

Markieren von Amazon-Pinpoint-Ressourcen

Ein Tag ist ein Label, das Sie definieren und mit AWS-Ressourcen, darunter bestimmten Arten von Amazon-Pinpoint-Ressourcen, verknüpfen können. Tags können Ihnen dabei helfen, Ressourcen auf verschiedene Arten zu kategorisieren und zu verwalten (z. B. nach Zweck, Besitzer, Umgebung oder anderen Kriterien). Sie können Tags zum Beispiel verwenden, um Richtlinien oder eine Automatisierung anzuwenden oder um Ressourcen zu identifizieren, für die bestimmte Compliance-Anforderungen erfüllt werden müssen. Sie können den folgenden Amazon-Pinpoint-Ressourcentypen Tags hinzufügen:

- Kampagnen
- Nachrichtenvorlagen
- Projekte (Anwendungen)
- Segmente

Eine Ressource kann bis zu 50 Tags enthalten.

Verwalten von Tags

Jedes Tag besteht aus einem Schlüssel und einem optionalen Wert. Beides können Sie definieren. Ein Tag-Schlüssel ist eine allgemeine Markierung, die wie eine Kategorie für spezifischere Tag-Werte fungiert. Ein Tag-Wert dient als Bezeichnung für einen Tag-Schlüssel.

Ein Tag-Schlüssel kann bis zu 128 Zeichen enthalten. Ein Tag-Wert kann bis zu 256 Zeichen enthalten. Die Zeichen können Unicode-Buchstaben, Zahlen, Leerzeichen oder eines der folgenden Symbole sein: _ . : / = + -. Für Tags gelten die folgenden zusätzlichen Einschränkungen:

- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden.
- Für jede zugeordnete Ressource muss jeder Tag-Schlüssel eindeutig sein und darf nur einen Wert haben.
- Das Präfix `aws:` ist für die Verwendung durch AWS reserviert. Sie können es nicht in von Ihnen definierten Tag-Schlüsseln oder -Werten verwenden. Sie können auch keine Tag-Schlüssel oder -Werte mit diesem Präfix bearbeiten oder entfernen. Tags mit diesem Präfix werden beim Kontingent von 50 Tags pro Ressource nicht eingerechnet.

- Es ist nicht möglich, eine Ressource nur anhand ihrer Tags zu aktualisieren oder zu löschen. Darüber hinaus müssen Sie den Amazon-Ressourcennamen (ARN) oder die Ressourcen-ID angeben – je nach Operation, die Sie verwenden.
- Sie können Tags öffentlichen oder freigegebenen Ressourcen zuordnen. Die Tags sind jedoch nur für Ihr AWS-Konto verfügbar, nicht jedoch für andere Konten, die die Ressource gemeinsam nutzen. Außerdem sind Tags nur für Ressourcen in der angegebenen AWS-Region für Ihr AWS-Konto verfügbar.

Zum Hinzufügen, Anzeigen, Aktualisieren und Entfernen von Tag-Schlüsseln und -Werten von Amazon-Pinpoint-Ressourcen können Sie die AWS CLI (AWS Resource Groups), die Amazon-Pinpoint-API, die AWS Command Line Interface-Tagging-API oder ein AWS-SDK verwenden. Um Tag-Schlüssel und -Werte über alle AWS-Ressourcen in der angegebenen AWS-Region für Ihr AWS-Konto zu verwalten, einschließlich Amazon-Pinpoint-Ressourcen, können Sie die AWS Resource Groups-Tagging-API verwenden.

Verwenden von Tags in IAM-Richtlinien

Nachdem Sie mit der Implementierung von Tags begonnen haben, können Sie tagbasierte Berechtigungen auf Ressourcenebene auf AWS Identity and Access Management (IAM)-Richtlinien und API-Vorgänge anwenden. Dies umfasst Vorgänge, die das Hinzufügen von Tags zu Ressourcen beim Erstellen von Ressourcen unterstützen. Wenn Sie Tags in dieser Weise verwenden, können Sie genauer steuern, welche Gruppen und Benutzer in Ihrem AWS-Konto die Berechtigung zum Erstellen und Markieren von Ressourcen haben, und welche Gruppen und Benutzer über die Berechtigung zum allgemeineren Erstellen, Aktualisieren und Entfernen von Tags verfügen.

Beispielsweise können Sie eine Richtlinie erstellen, die es einem Benutzer ermöglicht, vollen Zugriff auf alle Amazon-Pinpoint-Ressourcen zu haben, deren Name ein Wert im `Owner`-Tag für die Ressource ist:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "ModifyResourceIfOwner",
         "Effect": "Allow",
         "Action": "mobiletargeting:*",
         "Resource": "*",
         "Condition": {
```

```
            "StringEqualsIgnoreCase": {
               "aws:ResourceTag/Owner": "${aws:username}"
            }
         }
      }
   ]
}
```

Wenn Sie Tag-basierte Berechtigungen auf Ressourcenebene definieren, werden die Berechtigungen sofort wirksam. Dies bedeutet, dass Ihre Ressourcen besser geschützt sind, sobald sie erstellt wurden, und Sie schnell damit beginnen können, die Verwendung von Tags für neue Ressourcen zu erzwingen. Mithilfe von Berechtigungen auf Ressourcenebene können Sie auch steuern, welche Tag-Schlüssel und -Werte können mit neuen und vorhandenen Ressourcen verknüpft werden können. Weitere Informationen finden Sie unter Zugriffssteuerung mit Tags im AWS-IAM-Benutzerhandbuch.

Hinzufügen von Tags zu Ressourcen

In den folgenden Beispielen wird gezeigt, wie Sie einer Amazon-Pinpoint-Ressource mithilfe der AWS CLI und der Amazon-Pinpoint-REST-API ein Tag hinzufügen. Sie können auch ein beliebiges unterstütztes AWS-SDK verwenden, um einer Ressource ein Tag hinzuzufügen.

Um ein Tag für mehrere Amazon-Pinpoint-Ressourcen in einem einzigen Vorgang hinzuzufügen, verwenden Sie die Tagging-Operationen für Ressourcengruppen der AWS CLI oder der AWS Resource Groups-Tagging-API.

Hinzufügen von Tags mithilfe der API

Um eine neue Ressource zu erstellen und ihr mithilfe der Amazon-Pinpoint-REST-API ein Tag hinzuzufügen, senden Sie eine POST-Anforderung an den entsprechenden Ressourcen-URI. Geben Sie im Text der Anforderung den `tags`-Parameter und die Werte an. Das folgende Beispiel zeigt, wie Sie ein Tag angeben, wenn Sie ein neues Projekt erstellen.

```
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"
   }
}
```

Um einer vorhandenen Ressource ein Tag hinzuzufügen, senden Sie eine POST-Anforderung an den Tags-URI. Fügen Sie den Amazon-Ressourcennamen (ARN) der Ressource in den URI ein. Der ARN sollte URL-codiert sein. Geben Sie im Anforderungstext den `tags`-Parameter und die Werte an, wie im folgenden Beispiel gezeigt.

```
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"
   }
}
```

Hinzufügen von Tags mithilfe der AWS CLI

Um eine neue Ressource zu erstellen und dieser ein Tag mit der AWS CLI hinzuzufügen, verwenden Sie den entsprechenden `create`-Befehl für die Ressource. Fügen Sie die `tags`-Parameter und -Werte hinzu. Das folgende Beispiel zeigt, wie Sie Tags angeben, wenn Sie ein neues Projekt erstellen.

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}
```

Gehen Sie im vorhergehenden Beispiel wie folgt vor:

- Ersetzen Sie *myProject* durch den Namen, den Sie dem Projekt zuweisen möchten.
- Ersetzen Sie *key1* und *key2* durch die Schlüssel der Tags, die Sie der Ressource hinzufügen möchten.
- Ersetzen Sie *value1* und *value2* durch die Werte der Tags, die Sie für die jeweiligen Schlüssel hinzufügen möchten.

Weitere Informationen zu den Befehlen, die Sie zum Erstellen einer Amazon-Pinpoint-Ressource verwenden können, finden Sie in der AWS CLI-Befehl-Referenz.

Um einer vorhandenen Ressource ein Tag hinzuzufügen, verwenden Sie den `tag-resource`-Befehl, und geben Sie die entsprechenden Werte für die erforderlichen Parameter an:

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}
```

Gehen Sie im vorhergehenden Beispiel wie folgt vor:

- Ersetzen Sie *resource-arn* durch den Amazon-Ressourcennamen (ARN) der Ressource, der Sie ein Tag hinzufügen möchten.
- Ersetzen Sie *key1* und *key2* durch die Schlüssel der Tags, die Sie der Ressource hinzufügen möchten.
- Ersetzen Sie *value1* und *value2* durch die Werte der Tags, die Sie für die jeweiligen Schlüssel hinzufügen möchten.

Anzeigen von Tags für Ressourcen

In den folgenden Beispielen wird gezeigt, wie die AWS CLI und die Amazon-Pinpoint-REST-API verwendet wird, um eine Liste aller Tags (Schlüssel und Werte) anzuzeigen, die einer Amazon-Pinpoint-Ressource zugeordnet sind. Sie können auch jedes unterstützte AWS-SDK verwenden, um die Tags anzuzeigen, die einer Ressource zugeordnet sind.

Anzeigen von Tags mithilfe der API

Um mit der Amazon-Pinpoint-REST-API alle Tags anzuzeigen, die einer bestimmten Ressource zugeordnet sind, senden Sie eine GET-Anforderung an den Tags-URI, einschließlich des Amazon-Ressourcennamens (ARN) der Ressource. Der ARN sollte URL-codiert sein. Die folgende Anforderung ruft beispielsweise alle Tags ab, die einer angegebenen Kampagne (*resource-arn*) zugeordnet sind:

```
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
```

Die JSON-Antwort auf die Anforderung enthält ein `tags`-Objekt. Das `tags`-Objekt listet alle Tag-Schlüssel und -Werte auf, die der Kampagne zugeordnet sind.

Um alle Tags anzuzeigen, die mehr als einer Ressource desselben Typs zugeordnet sind, senden Sie eine GET-Anforderung an den entsprechenden Ressourcen-URI. Beispiel: Die folgende Anforderung ruft Informationen zu allen Kampagnen im angegebenen Projekt (*application-id*) ab:

```
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
```

In der JSON-Antwort auf die Anforderung sind alle Kampagnen im Projekt aufgeführt. Das `tags`-Objekt der einzelnen Kampagnen listet alle Tag-Schlüssel und -Werte auf, die mit der Kampagne verknüpft sind.

Anzeigen von Tags mithilfe der AWS CLI

Um mit AWS CLI eine Liste der Tags anzuzeigen, die einer bestimmten Ressource zugeordnet sind, führen Sie den `list-tags-for-resource`-Befehl aus, und geben Sie den Amazon-Ressourcenname (ARN) der Ressource für den `resource-arn`-Parameter an, wie im folgenden Beispiel gezeigt:

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
```

Um eine Liste aller Amazon-Pinpoint-Ressourcen anzuzeigen, die über Tags verfügen, sowie aller Tags, die den einzelnen Ressourcen zugeordnet sind, verwenden Sie den get-resources-Befehl der AWS Resource Groups-Markieren API. Legen Sie den `resource-type-filters`-Parameter wie im folgenden Beispiel gezeigt fest.

Linux, macOS, or Unix

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

Windows Command Prompt

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

Die Ausgabe des Befehls ist eine Liste der ARNs für alle Amazon-Pinpoint-Ressourcen, die über Tags verfügen. Die Liste enthält alle Tag-Schlüssel und -Werte, die den einzelnen Ressourcen zugeordnet sind.

Aktualisieren von Tags für Ressourcen

Zum Aktualisieren (Überschreiben) eines Tags für eine Amazon-Pinpoint-Ressource haben Sie mehrere Möglichkeiten. Welche die zum Aktualisieren eines Tags beste Methode ist, hängt von folgenden Faktoren ab:

- Dem Typ der Ressource, für die Sie Tags aktualisieren möchten
- Ob Sie ein Tag für eine Ressource oder mehrere Ressourcen gleichzeitig aktualisieren möchten
- Ob Sie einen Tag-Schlüssel, einen Tag-Wert oder beides aktualisieren möchten

Um einen Tag-Schlüssel für ein Amazon-Pinpoint-Projekt oder mehrere Ressourcen gleichzeitig zu aktualisieren, können Sie die Tagging-Operationen für Ressourcengruppen der AWS CLI oder der AWS Resource Groups-Tagging-API verwenden. Die Amazon-Pinpoint-API bietet derzeit keine direkte Unterstützung für eine dieser Aufgaben.

Um einen Tag für eine Ressource zu aktualisieren, entfernen Sie das aktuelle Tag und fügen Sie ein neues Tag hinzu, indem Sie die Amazon-Pinpoint-API verwenden.

Entfernen von Tags von Ressourcen

In den folgenden Beispielen wird gezeigt, wie ein Tag (sowohl der Schlüssel als auch der Wert) aus einer Amazon-Pinpoint-Ressource mithilfe der AWS CLI und der Amazon-Pinpoint-REST-API entfernt wird. Sie können auch ein beliebiges AWS-unterstütztes SDK verwenden, um ein Tag aus einer Ressource zu entfernen.

Um ein Tag von mehreren Amazon-Pinpoint-Ressourcen in einem einzigen Vorgang zu entfernen, verwenden Sie die Tagging-Operationen für Ressourcengruppen der AWS CLI oder der AWS Resource Groups-Tagging-API. Um nur einen bestimmten Tag-Wert, also nicht den Tag-Schlüssel, von einer Ressource zu entfernen, können Sie das Tag für die Ressource aktualisieren.

Entfernen von Tags mithilfe der API

Um ein Tag aus einer Ressource mithilfe der Amazon-Pinpoint-REST-API zu entfernen, senden Sie eine DELETE-Anforderung an den Tags-URI. Fügen Sie im URI eine Abfragezeichenfolge an, die den

Amazon-Ressourcennamen (ARN) der Ressource enthält, von der Sie ein Tag entfernen möchten, gefolgt vom `tagKeys`-Parameter und dem zu entfernenden Tag. Beispiele:

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

Wobei gilt:

- *`Endpunkt`* ist der Amazon-Pinpoint-Endpunkt für die AWS-Region, die die Ressource hostet.
- *`resource-arn`* ist der ARN der Ressource, von der Sie ein Tag entfernen möchten.
- *`key`* ist der Tag, den Sie von der Ressource entfernen möchten.

Alle Parameter sollten URL-codiert sein.

Um mehrere Tag-Schlüssel als auch die zugehörigen Werte von einer Ressource zu entfernen, fügen Sie den `tagKeys`-Parameter und das Argument für jedes zusätzliche zu entfernende Tag an, getrennt durch ein kaufmännisches Und-Zeichen (&). Beispiele:

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

Alle Parameter sollten URL-codiert sein.

Entfernen von Tags mithilfe der AWS CLI

Um ein Tag aus einer Ressource mithilfe der AWS CLI zu entfernen, führen Sie den `untag-resource`-Befehl aus. Geben Sie den `tag-keys`-Parameter und das -Argument an, wie im folgenden Beispiel gezeigt.

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
```

Nehmen Sie im vorherigen Beispiel Sie die folgenden Änderungen vor:

- Ersetzen Sie *resource-arn* durch den ARN der Ressource, von der Sie Tags entfernen möchten.
- Ersetzen Sie *key1* und *key2* durch die Schlüssel der Tags, die Sie aus der Ressource entfernen möchten.

Ähnliche Informationen

Weitere Informationen zu den CLI-Befehlen, die Sie für die Verwaltung von Amazon-Pinpoint-Ressourcen verwenden können, finden Sie im Amazon-Pinpoint-Abschnitt der AWS CLI-Befehlsreferenz.

Weitere Informationen zu den Ressourcen in der Amazon-Pinpoint-API, einschließlich unterstützter HTTP(S)-Methoden, Parameter und Schemata, finden Sie in der Amazon-Pinpoint-API-Referenz.

Anpassen von Empfehlungen mit AWS Lambda

In Amazon Pinpoint können Sie personalisierte Empfehlungen aus einem Empfehlungsmodell abrufen und zu Nachrichten hinzufügen, die Sie von Kampagnen und Journeys aus senden. Ein Empfehlungsmodell ist ein Art Machine-Learning-(ML-)Modell, das Muster in Daten findet und Vorhersagen sowie Empfehlungen basierend auf den gefundenen Mustern generiert. Es sagt vorher, welche Produkte oder Elemente aus einer Gruppe von Produkten oder Elementen ein bestimmter Benutzer bevorzugen wird, und stellt diese Informationen in Form einer Reihe von Empfehlungen für den Benutzer bereit.

Wenn Sie Empfehlungsmodelle mit Amazon Pinpoint verwenden, können Sie basierend auf den Attributen und Verhaltensweisen der einzelnen Empfänger personalisierte Empfehlungen an die Nachrichtenempfänger senden. Mithilfe von AWS Lambda können Sie diese Empfehlungen zudem anpassen und verbessern. Beispielsweise können Sie eine Empfehlung dynamisch von einem einzelnen Textwert (z. B. einem Produktnamen oder einer ID) in komplexere Inhalte (z. B. einen Produktnamen, eine Beschreibung und ein Bild) umwandeln. Dies ist in Echtzeit möglich, wenn Amazon Pinpoint die Nachricht sendet.

Dieses Feature ist in den folgenden AWS-Regionen verfügbar: USA Ost (Nord-Virginia), USA West (Oregon), Asien-Pazifik (Mumbai), Asien-Pazifik (Sydney) und Europa (Irland).

Themen

Verwenden von Empfehlungen in Nachrichten

Um ein Empfehlungsmodell mit Amazon Pinpoint zu verwenden, erstellen Sie zunächst eine Amazon-Personalize-Lösung und stellen diese Lösung als Amazon-Personalize-Kampagne bereit. Anschließend erstellen Sie eine Konfiguration für das Empfehlungsmodell in Amazon Pinpoint. In der Konfiguration geben Sie Einstellungen an, die bestimmen, wie Empfehlungsdaten aus der Amazon-Personalize-Kampagne abgerufen und verarbeitet werden sollen. Diese Einstellungen legen

auch fest, ob eine AWS Lambda-Funktion für die zusätzliche Verarbeitung der abgerufenen Daten aufgerufen werden soll.

Amazon Personalize ist ein AWS-Service, der Sie bei der Erstellung von ML-Modellen unterstützt, die personalisierte Empfehlungen in Echtzeit für Kunden bereitstellen, die Ihre Anwendungen verwenden. Amazon Personalize führt Sie durch den Prozess der Erstellung und Schulung eines ML-Modells und der anschließenden Vorbereitung und Bereitstellung des Modells als Amazon-Personalize-Kampagne. Sie können dann personalisierte Empfehlungen in Echtzeit aus der Kampagne abrufen. Weitere Informationen zu Amazon Personalize finden Sie im Amazon-Personalize-Entwicklerhandbuch.

AWS Lambda ist ein Datenverarbeitungsservice, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Sie packen Ihren Code und laden ihn auf AWS Lambda als Lambda-Funktion hoch. AWS Lambda führt die Funktion aus, wenn die Funktion aufgerufen wird. Eine Funktion kann manuell von Ihnen, automatisch als Reaktion auf Ereignisse oder als Reaktion auf Anforderungen von Anwendungen oder Diensten, einschließlich Amazon Pinpoint, aufgerufen werden. Informationen zum Erstellen und Abrufen und Lambda-Funktionen finden Sie im AWS Lambda-Entwicklerhandbuch.

Nachdem Sie eine Amazon-Pinpoint-Konfiguration für ein Empfehlungsmodell erstellt haben, können Sie den Nachrichten, die Sie von Kampagnen und Journeys aus senden, Empfehlungen aus dem Modell hinzufügen. Hierfür verwenden Sie Nachrichtenvorlagen, die Nachrichtenvariablen für empfohlene Attribute enthalten. Ein empfohlenes Attribut ist ein dynamisches Endpunkt- oder Benutzerattribut zum Speichern von Empfehlungsdaten. Diese Attribute definieren Sie, wenn Sie die Konfiguration für ein Empfehlungsmodell erstellen.

In den folgenden Arten von Nachrichtenvorlagen können Sie Variablen für empfohlene Attribute verwenden:

- E-Mail-Vorlagen für E-Mail-Nachrichten, die Sie von Kampagnen oder Journeys senden.
- Push-Benachrichtigungsvorlagen für Push-Benachrichtigungen, die Sie von Kampagnen senden.
- SMS-Vorlagen, für SMS-Textnachrichten, die Sie von Kampagnen senden.

Weitere Informationen zur Verwendung von Empfehlungsmodellen mit Amazon Pinpoint, finden Sie unter Machine-Learning-Modelle im Amazon-Pinpoint-Benutzerhandbuch.

Wenn Sie Amazon Pinpoint so konfigurieren, dass eine Lambda-Funktion aufgerufen wird, die Empfehlungsdaten verarbeitet, führt Amazon Pinpoint jedes Mal die folgenden allgemeinen Aufgaben

durch, wenn personalisierte Empfehlungen in einer Nachricht für eine Kampagne oder Journey gesendet werden:

1. Evaluierung und Verarbeitung der Konfigurationseinstellungen und der Inhalte der Nachricht sowie der Nachrichtenvorlage.
2. Festlegung, dass die Nachrichtenvorlage mit einem Empfehlungsmodell verbunden ist.
3. Evaluierung der Konfigurationseinstellungen für die Verbindung mit dem Modell und die Verwendung des Modells. Diese werden von der Empfehlungsmodell-Ressource für das Modell definiert.
4. Erkennen einer oder mehrerer Nachrichtenvariablen für empfohlene Attribute, die über die Konfigurationseinstellungen für das Modell definiert werden.
5. Abrufen von Empfehlungsdaten aus der Amazon-Personalize-Kampagne, die in den Konfigurationseinstellungen für das Modell angegeben ist. Für diese Aufgabe wird die Operation GetRecommendations der Amazon Personalize-Laufzeit-API verwendet.
6. Hinzufügen der entsprechenden Empfehlungsdaten zu einem dynamischen empfohlenen Attribut (`RecommendationItems`) für jeden Nachrichtenempfänger.
7. Aufruf Ihrer Lambda-Funktion und Senden der Empfehlungsdaten für die einzelnen Empfänger zur Verarbeitung an diese Funktion.

   Die Daten werden als JSON-Objekt gesendet, das die Endpunktdefinition für jeden Empfänger enthält. Jede Endpunktdefinition enthält ein Feld `RecommendationItems`, das ein geordnetes Array von 1–5 Werten enthält. Die Anzahl der Werte in dem Array hängt von den Konfigurationseinstellungen für das Modell ab.
8. Warten, bis Ihre Lambda-Funktion die Daten verarbeitet und die Ergebnisse zurückgibt.

   Die Ergebnisse sind ein JSON-Objekt, das eine aktualisierte Endpunktdefinition für jeden Empfänger enthält. Jede aktualisierte Endpunktdefinition enthält ein neues `Recommendations`-Objekt. Dieses Objekt enthält 1–10 Felder, eines für jedes benutzerdefinierte empfohlene Attribut, das Sie in den Konfigurationseinstellungen für das Modell definiert haben. In jedem dieser Felder werden erweiterte Empfehlungsdaten für den Endpunkt gespeichert.
9. Verwendung der aktualisierten Endpunktdefinition für die einzelnen Empfänger, um die einzelnen Nachrichtenvariablen durch den entsprechenden Wert für die betreffenden Empfänger zu ersetzen.
10. Sendet eine Version der Nachricht, die die personalisierten Empfehlungen für jeden Nachrichtenempfänger enthält.

Um Empfehlungen auf diese Weise anzupassen und zu verbessern, erstellen Sie zunächst eine Lambda-Funktion, die die von Amazon Pinpoint, gesendeten Endpunktdefinitionen verarbeitet und aktualisierte Endpunktdefinitionen zurückgibt. Weisen Sie dann der Funktion eine Lambda-Funktionsrichtlinie zu und autorisieren Sie Amazon Pinpoint, die Funktion aufzurufen. Konfigurieren Sie anschließend das Empfehlungsmodell in Amazon Pinpoint. Wenn Sie das Modell konfigurieren, geben Sie die aufzurufende Funktion an, und definieren Sie die zu verwendenden empfohlenen Attribute.

Erstellen der Lambda-Funktion

Informationen zum Erstellen einer Lambda-Funktion finden Sie unter Erste Schritte im AWS Lambda-Entwicklerhandbuch. Beachten Sie beim Entwerfen und Entwickeln der Funktion die folgenden Anforderungen und Richtlinien.

Eingabeereignisdaten

Wenn Amazon Pinpoint eine Lambda-Funktion für ein Empfehlungsmodell aufruft, sendet es eine Nutzlast, die die Konfiguration und andere Einstellungen für die Kampagne oder Journey enthält, von der die Nachricht gesendet wird. Die Nutzlast enthält ein `Endpoints`-Objekt. Hierbei handelt es sich um eine Zuordnung, die Endpunkt-IDs mit Endpunktdefinitionen für Nachrichtenempfänger verknüpft.

Die Endpunktdefinitionen verwenden die Struktur, die von der Endpunkt-Ressource der Amazon-Pinpoint-API definiert wird. Sie enthalten jedoch auch ein Feld für ein dynamisches empfohlenes Attribut namens `RecommendationItems`. Das Feld `RecommendationItems` enthält mindestens ein empfohlenes Element für den Endpunkt, das von der Amazon-Personalize-Kampagne zurückgegeben wird. Der Wert für dieses Feld ist ein geordnetes Array von 1–5 empfohlenen Elementen (als Zeichenfolgen). Die Anzahl der Elemente in dem Array hängt davon ab, wie viele Elemente Amazon Pinpoint laut Ihrer Konfiguration für jeden Endpunkt oder Benutzer abrufen soll.

Beispiele:

```
"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"
    }
}
```

Im vorherigen Beispiel lauten die relevanten Amazon-Pinpoint-Einstellungen wie folgt:

- Das Empfehlungsmodell ist so konfiguriert, dass drei empfohlene Elemente für jeden Endpunkt oder Benutzer abgerufen werden. (Der Wert für die Eigenschaft `RecommendationsPerMessage` ist auf `3` festgelegt.) Mit dieser Einstellung werden nur die ersten, zweiten und dritten empfohlenen Elemente für jeden Endpunkt oder Benutzer von Amazon Pinpoint abgerufen und hinzugefügt.
- Das Projekt ist für die Verwendung von benutzerdefinierten Benutzerattributen konfiguriert, in denen der Vorname, Nachname und das Umfeld der Benutzer gespeichert werden. (Im Objekt `UserAttributes` sind die Werte für diese Attribute enthalten.)
- Das Projekt ist für die Verwendung eines benutzerdefinierten Endpunktattributs (`AddressType`) konfiguriert, das angibt, ob der Endpunkt die bevorzugte Adresse (Kanal) des Benutzers für den Empfang von Nachrichten aus dem Projekt ist. (Im Objekt `Attributes` ist der Wert für dieses Attribut enthalten.)

Wenn Amazon Pinpoint die Lambda-Funktion aufruft und diese Nutzlast als Ereignisdaten sendet, übergibt AWS Lambda die Daten zur Verarbeitung an die Lambda-Funktion.

Jede Nutzlast kann Daten für bis zu 50 Endpunkte enthalten. Wenn ein Segment mehr als 50 Endpunkte enthält, ruft Amazon Pinpoint die Funktion wiederholt für bis zu 50 Endpunkte gleichzeitig auf, bis die Funktion alle Daten verarbeitet.

Antwortdaten und Anforderungen

Berücksichtigen Sie beim Entwerfen und Entwickeln Ihrer Lambda-Funktion die Kontingente für Machine-Learning-Modelle. Wenn die Funktion die durch diese Kontingente definierten Bedingungen nicht erfüllt, kann Amazon Pinpoint die Nachricht nicht verarbeiten und senden.

Beachten Sie auch die folgenden Anforderungen:

- Die Funktion muss aktualisierte Endpunktdefinitionen in dem Format zurückgeben, das von den Eingabeereignisdaten bereitgestellt wurde.
- Jede aktualisierte Endpunktdefinition kann 1–10 benutzerdefinierte empfohlene Attribute für den Endpunkt oder Benutzer enthalten. Die Namen dieser Attribute müssen mit den Attributnamen übereinstimmen, die Sie beim Konfigurieren des Empfehlungsmodells in Amazon Pinpoint angeben.
- Alle benutzerdefinierten empfohlenen Attribute müssen in einem einzelnen `Recommendations`-Objekt für jeden Endpunkt oder Benutzer zurückgegeben werden. Mithilfe dieser Anforderung wird sichergestellt, dass keine Namenskonflikte auftreten. Sie können das `Recommendations`-Objekt einem beliebigen Speicherort in einer Endpunktdefinition hinzufügen.
- Der Wert für jedes benutzerdefinierte empfohlene Attribut muss eine Zeichenfolge (Einzelwert) oder ein Array von Zeichenfolgen (mehrere Werte) sein. Wenn es sich bei dem Wert um ein Array von Zeichenfolgen handelt, empfehlen wir, die Reihenfolge der empfohlenen Elemente beizubehalten, die von Amazon Personalize zurückgegeben wurden, wie im Feld `RecommendationItems` angegeben. Andernfalls könnte es sein, dass Ihre Inhalte die Vorhersagen des Modells für einen Endpunkt oder Benutzer nicht widerspiegeln.
- Die Funktion sollte keine anderen Elemente in den Ereignisdaten ändern. Dies gilt auch für andere Attributwerte für einen Endpunkt oder Benutzer. Es sollten nur Werte für benutzerdefinierte empfohlene Attribute hinzugefügt und zurückgegeben werden. Amazon Pinpoint akzeptiert keine Aktualisierungen anderer Werte in der Antwort der Funktion.
- Die Funktion muss in derselben AWS-Region gehostet werden wie das Amazon-Pinpoint-Projekt, das die Funktion aufruft. Wenn sich die Funktion und das Projekt nicht in derselben Region befinden, kann Amazon Pinpoint keine Ereignisdaten an die Funktion senden.

Wenn eine der oben genannten Anforderungen nicht erfüllt ist, kann Amazon Pinpoint die Nachricht nicht verarbeiten und an einen oder mehrere Endpunkte senden. Dies kann zum Fehlschlagen einer Kampagnen- oder Journey-Aktivität führen.

Schließlich empfehlen wir, 256 gleichzeitige Ausführungen für die Funktion zu reservieren.

Insgesamt sollte Ihre Lambda-Funktion die Ereignisdaten verarbeiten, die von Amazon Pinpoint gesendet werden, und geänderte Endpunktdefinitionen zurückgeben. Zu diesem Zweck kann die Funktion jeden Endpunkt im `Endpoints`-Objekt durchlaufen und für jeden Endpunkt Werte für die zu verwendenden benutzerdefinierten empfohlenen Attribute erstellen und festlegen. Der folgende in Python geschriebene Beispiel-Handler, der mit dem vorherigen Beispiel für Eingabeereignisdaten fortfährt, zeigt folgende Ausgabe:

```
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
```

Im vorherigen Beispiel übergibt AWS Lambda die Ereignisdaten als `event`-Parameter an den Handler. Der Handler durchläuft jeden Endpunkt im `Endpoints`-Objekt und legt Werte für benutzerdefinierte empfohlene Attribute mit den Namen `Recommendations.Title` und `Recommendations.Genre` fest. Die `return`-Anweisung gibt jede aktualisierte Endpunktdefinition an Amazon Pinpoint zurück.

Bei Fortführung des früheren Beispiels für Eingabeereignisdaten lauten die aktualisierten Endpunktdefinitionen wie folgt:

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

Im vorherigen Beispiel hat die Funktion das erhaltene `Endpoints`-Objekt geändert und die Ergebnisse zurückgegeben. Das `Endpoint`-Objekt für jeden Endpunkt enthält nun ein neues `Recommendations`-Objekt mit den Feldern `Genre` und `Title`. In jedem dieser Felder wird ein geordnetes Array von drei Werten (als Zeichenfolgen) gespeichert, wobei jeder Wert erweiterte Inhalte für ein entsprechendes empfohlenes Element im Feld `RecommendationItems` bereitstellt.

Zuweisen einer Lambda-Funktionsrichtlinie

Bevor Sie die Lambda-Funktion zum Verarbeiten von Empfehlungsdaten verwenden können, müssen Sie Amazon Pinpoint zum Aufruf der Funktion berechtigen. Um eine Aufrufberechtigung zu erteilen, weisen Sie der Funktion eine Lambda-Funktionsrichtlinie zu. Eine Lambda-Funktionsrichtlinie ist eine ressourcenbasierte Berechtigungsrichtlinie, die bestimmt, welche Entitäten eine Funktion verwenden dürfen und welche Aktionen diese Entitäten durchführen können. Weitere Informationen finden

Sie unter Verwenden von ressourcenbasierten Richtlinien für AWS Lambda im AWS Lambda-Entwicklerhandbuch.

Die folgende Beispielrichtlinie ermöglicht es dem Amazon-Pinpoint-Service-Prinzipal, die `lambda:InvokeFunction`-Aktion für eine bestimmte Amazon-Pinpoint-Kampagne (*`campaignId`*) in einem bestimmten Amazon-Pinpoint-Projekt (*`projectId`*) zu verwenden:

```
{
  "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/*"
    }
  }
}
```

Die Funktionsrichtlinie erfordert einen `Condition`-Block, der einen `AWS:SourceArn`-Schlüssel beinhaltet. Dieser Schlüssel gibt an, welche Ressource die Funktion aufrufen darf. Im vorherigen Beispiel ermöglicht es die Richtlinie einer bestimmten Kampagne, die Funktion aufzurufen.

Sie können auch eine Richtlinie schreiben, die es einem Amazon-Pinpoint-Service-Prinzipal ermöglicht, die Aktion für alle Kampagnen und Journeys in einem bestimmten Amazon-Pinpoint-Projekt (*`projectId`*) zu verwenden. Die folgende Beispielrichtlinie zeigt dies:

```
{
  "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/*"
```

```
    }
  }
}
```

Im Gegensatz zum ersten Beispiel erlaubt es der `AWS:SourceArn`-Schlüssel im `Condition`-Block dieses Beispiels einem bestimmten Projekt, die Funktion aufzurufen. Diese Berechtigung gilt für alle Kampagnen und Journeys im Projekt.

Um eine allgemeinere Richtlinie zu schreiben, können Sie Wildcards (*), die mehrere Zeichen darstellen, verwenden. So können Sie beispielsweise den folgenden `Condition`-Block verwenden, um jedem Amazon-Pinpoint-Projekt den Aufruf der Funktion zu gestatten:

```
"Condition": {
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}
```

Wenn Sie die Lambda-Funktion mit allen Projekten für Ihr Amazon-Pinpoint-Konto verwenden möchten, empfehlen wir, den `Condition`-Block der Richtlinie auf die vorherige Weise zu konfigurieren. Als bewährte Methode sollten Sie jedoch Richtlinien erstellen, die nur die zum Ausführen einer bestimmten Aktion für eine bestimmte Ressource erforderlichen Berechtigungen enthalten.

Erteilen der Berechtigung für Amazon Pinpoint zum Aufrufen der Funktion

Nachdem Sie der Funktion eine Lambda-Funktionsrichtlinie zugewiesen haben, können Sie Berechtigungen hinzufügen, die es Amazon Pinpoint ermöglichen, die Funktion für ein bestimmtes Projekt, eine bestimmte Kampagne oder Journey aufzurufen. Sie können hierfür die AWS Command Line Interface (AWS CLI) und den Lambda-Befehl `add-permission` verwenden. Das folgende Beispiel zeigt die Vorgehensweise für ein bestimmtes Projekt (*`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/*
```

Das vorherige Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Microsoft Windows das Zeilenfortsetzungszeichen, also den umgekehrten Schrägstrich (\), durch ein Caret-Zeichen (^).

Wird der Befehl erfolgreich ausgeführt, sehen Sie eine Ausgabe ähnlich der folgenden:

```
{
  "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/*\"}}}"
}
```

Der `Statement`-Wert ist eine JSON-Zeichenfolgenversion der Anweisung, die der Lambda-Funktionsrichtlinie hinzugefügt wurde.

Konfigurieren des Empfehlungsmodells

Um Amazon Pinpoint für den Aufruf der Lambda-Funktion für ein Empfehlungsmodell zu konfigurieren, geben Sie die folgenden Lambda-spezifischen Konfigurationseinstellungen für das Modell an:

- `RecommendationTransformerUri`: Diese Eigenschaft gibt den Namen oder den Amazon-Ressourcennamen (ARN) der Lambda-Funktion an.
- `Attributes`: Dieses Objekt ist eine Zuordnung, die die benutzerdefinierten empfohlenen Attribute definiert, die die Funktion jeder Endpunktdefinition hinzufügt. Jedes dieser Attribute kann als Nachrichtenvariable in einer Nachrichtenvorlage verwendet werden.

Sie können diese Einstellungen mithilfe der Ressource Empfehlungsmodelle der Amazon-Pinpoint-API (wenn Sie die Konfiguration für ein Modell erstellen) oder mithilfe der Ressource Empfängermodell der Amazon-Pinpoint-API (wenn Sie die Konfiguration für ein Modell aktualisieren) angeben. Sie können diese Einstellungen auch über die Amazon-Pinpoint-Konsole definieren.

Weitere Informationen zur Verwendung von Empfehlungsmodellen mit Amazon Pinpoint, finden Sie unter Machine-Learning-Modelle im Amazon-Pinpoint-Benutzerhandbuch.

Löschen von Daten aus Amazon Pinpoint

Abhängig davon, wie Sie Amazon Pinpoint verwenden, werden möglicherweise Daten gespeichert, die als personenbezogen angesehen werden könnten. Ein Endpunkt in Amazon Pinpoint enthält beispielsweise Kontaktinformationen für einen Endbenutzer, z. B. die E-Mail-Adresse oder Mobilnummer.

Sie können diese personenbezogenen Daten über die Konsole oder die Amazon-Pinpoint-API dauerhaft löschen. In diesem Thema finden Sie Verfahren zum Löschen verschiedener Arten von Daten, die als personenbezogen angesehen werden könnten.

Löschen von Endpunkten

Ein Endpunkt stellt eine einzelne Methode zur Kontaktaufnahme mit einem Kunden dar. Jeder Endpunkt kann sich auf eine E-Mail-Adresse eines Kunden, die ID eines Mobilgeräts, eine Telefonnummer oder einen anderen Zieltyp beziehen, an den Sie Nachrichten senden können. In vielen Zuständigkeiten gelten diese Informationen als persönlich.

Um alle Daten für einen bestimmten Endpunkt zu löschen, können Sie mit der Amazon-Pinpoint-API den Endpunkt löschen. Das folgende Verfahren veranschaulicht, wie Sie einen Endpunkt mithilfe der AWS CLI in Verbindung mit der Amazon-Pinpoint-API löschen. Bei diesem Verfahren wird vorausgesetzt, dass Sie die AWS CLI bereits installiert und konfiguriert haben. Weitere Informationen finden Sie unter Installieren der AWS CLI im AWS Command Line Interface -Benutzerhandbuch.

Um einen Endpunkt mit dem zu löschen AWS CLI

- Geben Sie in der Befehlszeile folgenden Befehl ein:

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

  Ersetzen Sie im vorherigen Befehl *810c7aab86d42fb2b56c8c966example* durch die ID des Projekts, dem der Endpunkt zugeordnet ist. Ersetzen Sie zudem *ad015a3bf4f1b2b0b82example* durch die eindeutige ID des Endpunkts selbst.

Um die Endpunkt-ID für einen bestimmten Endpunkt zu finden, bestimmen Sie, zu welchem Segment der Endpunkt gehört, und exportieren Sie dann das Segment aus Amazon Pinpoint. Die exportierten

Daten enthalten die Endpunkt-ID für jeden Endpunkt. Sie können ein Segment mithilfe der Amazon-Pinpoint-Konsole in eine Datei exportieren. Wie das geht, erfahren Sie unter Exportieren von Segmenten im Amazon-Pinpoint-Benutzerhandbuch. Sie können ein Segment mithilfe der Amazon-Pinpoint-API auch in einen Amazon Simple Storage Service (Amazon S3)-Bucket exportieren. Weitere Informationen hierzu finden Sie unter Exportieren von Endpunkten in diesem Handbuch.

Löschen der in Amazon S3 gespeicherten Segment- und Endpunktdaten

Sie können Segmente mithilfe der Amazon-Pinpoint-Konsole oder Amazon-Pinpoint-API aus einer Datei importieren, die in einem Amazon-S3-Bucket gespeichert ist. Sie können auch Anwendungs-, Segment- oder Endpunktdaten aus Amazon Pinpoint in einen Amazon-S3-Bucket exportieren. Die importierten und exportierten Dateien können personenbezogene Daten enthalten, z. B. E-Mail-Adressen, Mobilnummern und Informationen über den physischen Standort eines Endpunkts. Sie können diese Dateien aus Amazon S3 löschen.

Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter Wie entleere ich einen S3 Bucket? oder Wie lösche ich einen S3 Bucket?.

Löschen aller Projektdaten

Es ist möglich, alle Daten, die Sie für ein Amazon-Pinpoint-Projekt gespeichert haben, dauerhaft zu löschen. Löschen Sie hierfür das Projekt.

> **Warning**
>
> Wenn Sie ein Projekt löschen, löscht Amazon Pinpoint alle projektspezifischen Einstellungen und Daten für das Projekt. Die Informationen können nicht wiederhergestellt werden.

Wenn Sie ein Projekt löschen, löscht Amazon Pinpoint alle projektspezifischen Einstellungen für die Push-Benachrichtigungs- und Zwei-Wege-SMS-Messaging-Kanäle sowie alle Segmente, Kampagnen, Journeys und projektspezifischen Analysedaten, die in Amazon Pinpoint gespeichert sind, wie z. B.:

- Segmente – Alle Segmenteinstellungen und -daten. Bei dynamischen Segmenten umfasst dies die von Ihnen definierten Segmentgruppen und -filter. Bei importierten Segmenten umfasst dies

Endpunkte, Benutzer-IDs und andere Daten, die Sie importiert haben, sowie alle Filter, die Sie angewendet haben.

- Kampagnen – Alle Nachrichten, Nachrichtenverarbeitungen und -variablen, Analysedaten, Zeitpläne und anderen Einstellungen.
- Journeys – Alle Aktivitäten, Analysedaten, Zeitpläne und anderen Einstellungen.
- Analysen – Daten für alle Interaktionsmetriken, z. B. die Anzahl der gesendeten und zugestellten Nachrichten für Kampagnen und Journeys sowie alle Metriken für die Journey-Ausführung. Für mobile Apps und Web-Apps alle Eventdaten, die nicht an einen anderen AWS Service wie Amazon Kinesis gestreamt wurden, alle Funnels und Daten für Anwendungsnutzung, Umsatz und demografische Kennzahlen. Bevor Sie ein Projekt löschen, sollten Sie diese Daten nach einem anderen Speicherort exportieren.

Sie können ein Projekt mithilfe der Amazon-Pinpoint-Konsole löschen. Weitere Informationen finden Sie unter Löschen eines Projekts im Amazon-Pinpoint-Benutzerhandbuch. Sie können ein Projekt auch programmgesteuert mit der App-Ressource der Amazon-Pinpoint-API löschen.

Löschen aller AWS Daten durch Schließung Ihres Kontos AWS

Es ist auch möglich, alle Daten, die Sie in Amazon Pinpoint gespeichert haben, durch Schließen Ihres AWS -Kontos zu löschen. Durch diese Aktion werden jedoch auch alle anderen Daten — persönliche oder nicht personenbezogene — gelöscht, die Sie in allen anderen Diensten gespeichert haben. AWS Nach Ablauf der Frist nach der Schließung wird Ihr AWS Konto AWS dauerhaft geschlossen und Sie können es nicht mehr erneut öffnen. Alle Inhalte, die Sie nicht gelöscht haben, werden dauerhaft gelöscht, und alle AWS Dienste, die Sie nicht beendet haben, werden eingestellt. Weitere Informationen finden Sie im AWS Account Management Referenzhandbuch AWS unter Konto schließen.

> **Warning**
>
> Mit dem folgenden Verfahren werden alle Daten, die in Ihrem AWS Konto gespeichert sind, für alle AWS Dienste und AWS Regionen vollständig entfernt.

Sie können Ihr AWS Konto schließen, indem Sie die verwenden AWS Management Console.

Um dein AWS Konto zu schließen

1. Öffnen Sie das AWS Management Console unter https://console.aws.amazon.com.
2. Rufen Sie die Seite mit den Kontoeinstellungen unter https://console.aws.amazon.com/billing/home?#/account auf.

   > **Warning**
   >
   > Mit den folgenden Schritten werden alle Daten, die Sie in allen AWS Diensten in allen AWS Regionen gespeichert haben, dauerhaft gelöscht.

3. Lesen Sie unter Konto schließen den Haftungsausschluss, in dem die Folgen der Schließung Ihres AWS Kontos beschrieben werden. Wenn Sie den Bedingungen zustimmen, aktivieren Sie das Kontrollkästchen und wählen Sie dann Konto schließen.
4. Klicken Sie im Bestätigungsdialogfeld auf Konto schließen.

Codebeispiele für Amazon Pinpoint unter Verwendung von AWS-SDKs

Die folgenden Code-Beispiele zeigen, wie Sie Amazon Pinpoint mit einem AWS-SDK (Software Development Kit) verwenden.

Eine vollständige Liste der AWS-SDK-Entwicklerhandbücher und Code-Beispiele finden Sie unter Amazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Codebeispiele

Codebeispiele für Amazon Pinpoint mit SDKs AWS

Die folgenden Codebeispiele zeigen, wie Amazon Pinpoint mit einem AWS Software Development Kit (SDK) verwendet wird.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Servicefunktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien und serviceübergreifenden Beispiele sehen.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Codebeispiele

- Aktionen für Amazon Pinpoint mithilfe von SDKs AWS
  - Verwendung CreateApp mit einem AWS SDK oder CLI
  - Verwendung CreateCampaign mit einem AWS SDK oder CLI
  - Verwendung CreateExportJob mit einem AWS SDK oder CLI
  - Verwendung CreateImportJob mit einem AWS SDK oder CLI
  - Verwendung CreateSegment mit einem AWS SDK oder CLI
  - Verwendung DeleteApp mit einem AWS SDK oder CLI
  - Verwendung DeleteEndpoint mit einem AWS SDK oder CLI
  - Verwendung GetEndpoint mit einem AWS SDK oder CLI
  - Verwendung GetSegments mit einem AWS SDK oder CLI
  - Verwendung GetSmsChannel mit einem AWS SDK oder CLI
  - Verwendung GetUserEndpoints mit einem AWS SDK oder CLI
  - Verwendung SendMessages mit einem AWS SDK oder CLI
  - Verwendung UpdateEndpoint mit einem AWS SDK oder CLI

Aktionen für Amazon Pinpoint mithilfe von SDKs AWS

Die folgenden Codebeispiele zeigen, wie einzelne Amazon Pinpoint Pinpoint-Aktionen mit AWS SDKs durchgeführt werden. Diese Auszüge rufen die Amazon-Pinpoint-API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der Amazon-Pinpoint-API-Referenz.

Beispiele

- Verwendung CreateApp mit einem AWS SDK oder CLI
- Verwendung CreateCampaign mit einem AWS SDK oder CLI
- Verwendung CreateExportJob mit einem AWS SDK oder CLI
- Verwendung CreateImportJob mit einem AWS SDK oder CLI
- Verwendung CreateSegment mit einem AWS SDK oder CLI
- Verwendung DeleteApp mit einem AWS SDK oder CLI
- Verwendung DeleteEndpoint mit einem AWS SDK oder CLI
- Verwendung GetEndpoint mit einem AWS SDK oder CLI
- Verwendung GetSegments mit einem AWS SDK oder CLI
- Verwendung GetSmsChannel mit einem AWS SDK oder CLI
- Verwendung GetUserEndpoints mit einem AWS SDK oder CLI
- Verwendung SendMessages mit einem AWS SDK oder CLI
- Verwendung UpdateEndpoint mit einem AWS SDK oder CLI

Verwendung **`CreateApp`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateApp`.

CLI

AWS CLI

Beispiel 1: Erstellen einer Anwendung

Im folgenden `create-app`-Beispiel wird eine neue Anwendung (Projekt) erstellt.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

Ausgabe:

```
{
    "ApplicationResponse": {
```

```
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

Beispiel 2: Erstellen einer mit Tags versehenen Anwendung

Im folgenden `create-app`-Beispiel wird eine neue Anwendung (Projekt) erstellt und der Anwendung ein Tag (Schlüssel und Wert) zugeordnet.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

Ausgabe:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
            "Stack": "Test"
        }
    }
}
```

- Einzelheiten zur API finden Sie CreateAppin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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 "";
    }
}
```

- Einzelheiten zur API finden Sie CreateAppin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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
    }
}
```

- API-Details finden Sie CreateAppin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateCampaign`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateCampaign`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Erstellen Sie eine Kampagne.

```
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;
    }
```

```
}
```

- Einzelheiten zur API finden Sie CreateCampaignin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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}")
    }
}
```

- API-Details finden Sie CreateCampaignin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateExportJob`** mit einem AWS SDK oder CLI

Das folgende Codebeispiel zeigt, wie es verwendet wird`CreateExportJob`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Exportieren Sie einen Endpunkt.

```
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);
        }
    }
}
```

- Einzelheiten zur API finden Sie CreateExportJobin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateImportJob`** mit einem AWS SDK oder CLI

Das folgende Codebeispiel zeigt, wie es verwendet wird`CreateImportJob`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Importieren Sie ein Segment.

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

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

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

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

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

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

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

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

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket,
 key, roleArn);
```

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

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

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

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

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

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

- Einzelheiten zur API finden Sie CreateImportJobin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateSegment`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`CreateSegment`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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;
        }
}
```

- Einzelheiten zur API finden Sie CreateSegmentin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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
    }
}
```

- API-Details finden Sie CreateSegmentin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteApp`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteApp`.

CLI

AWS CLI

So löschen Sie eine Anwendung

Im folgenden `delete-app`-Beispiel wird eine Anwendung (Projekt) gelöscht.

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

Ausgabe:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- Einzelheiten zur API finden Sie DeleteAppin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Löschen Sie eine Anwendung.

```
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);
        }
    }
}
```

- Einzelheiten zur API finden Sie DeleteAppin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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.")
    }
}
```

- API-Details finden Sie DeleteAppin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteEndpoint`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteEndpoint`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Löschen eines Endpunktes

```
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");
    }
}
```

- Einzelheiten zur API finden Sie DeleteEndpointin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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")
    }
}
```

- API-Details finden Sie DeleteEndpointin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetEndpoint`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetEndpoint`.

CLI

AWS CLI

So rufen Sie Informationen über die Einstellungen und Attribute eines bestimmten Endpunkts für eine Anwendung ab

Das folgende `get-endpoint`-Beispiel ruft Informationen über die Einstellungen und Attribute eines bestimmten Endpunkts für eine Anwendung ab.

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

Ausgabe:

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

- Einzelheiten zur API finden Sie GetEndpointin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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");
    }
}
```

- Einzelheiten zur API finden Sie GetEndpointin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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)
    }
}
```

- API-Details finden Sie GetEndpointin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetSegments`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetSegments`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Listen Sie Segmente auf.

```
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);
        }
    }
}
```

- Einzelheiten zur API finden Sie GetSegmentsin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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}")
        }
    }
}
```

- API-Details finden Sie GetSegmentsin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetSmsChannel`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetSmsChannel`.

CLI

AWS CLI

So rufen Sie Informationen über den Status und die Einstellungen jedes Sprachkanals für eine Anwendung ab

Im folgenden `get-sms-channel`-Beispiel werden Status und Einstellungen des SMS-Kanals für eine Anwendung abgerufen.

```
aws pinpoint get-sms-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Ausgabe:

```
{
    "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
    }
}
```

- Einzelheiten zur API finden Sie GetSmsChannelin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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);
       }
   }
```

```
}
```

- Einzelheiten zur API finden Sie GetSmsChannelin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetUserEndpoints`** mit einem AWS SDK oder CLI

Das folgende Codebeispiel zeigt, wie es verwendet wird`GetUserEndpoints`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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);
        }
    }
}
```

- Einzelheiten zur API finden Sie GetUserEndpointsin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **SendMessages** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`SendMessages`.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Senden Sie eine E-Mail-Nachricht.

```
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;
    }
}
```

Senden Sie eine SMS-Nachricht.

```
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;
    }
}
```

- Einzelheiten zur API finden Sie SendMessagesin der AWS SDK for .NET API-Referenz.

CLI

AWS CLI

So senden Sie eine SMS-Nachricht über den Endpunkt einer Anwendung

Im folgenden `send-messages`-Beispiel wird eine Direktnachricht für eine Anwendung mit einem Endpunkt gesendet.

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

Inhalt von `myfile.json`:

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Endpoints": {
        "testendpoint": {}
    }
}
```

Ausgabe:

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
```

```
    }
}
```

Weitere Informationen finden Sie unter Amazon-Pinpoint-SMS-Kanal im Amazon-Pinpoint-Benutzerhandbuch.

- Einzelheiten zur API finden Sie SendMessagesin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Senden Sie eine E-Mail-Nachricht.

```
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);
        }
    }
}
```

Senden einer E-Mail-Nachricht mit CC-Werten.

```
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();
        }
    }
}
```

Senden Sie eine SMS-Nachricht.

```
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);
                }
        }
}
```

Senden Sie Batch-SMS-Nachrichten.

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

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

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

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

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

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

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

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber> <destinationNumber1>\s

                                Where:
                                  message - The body of the message to send.
```

```
                                appId - The Amazon Pinpoint project/application
ID to use when you send this message.
                                originationNumber - The phone number or
short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format (for example,
+1-555-555-5654).
                                destinationNumber - The recipient's phone
number.  For best results, you should specify the phone number in E.164 format
(for example, +1-555-555-5654).
                                destinationNumber1 - The second recipient's
phone number.  For best results, you should specify the phone number in E.164
format (for example, +1-555-555-5654).\s
                                """;

                if (args.length != 5) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                String destinationNumber1 = args[4];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber, destinationNumber1);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
                        String originationNumber,
                        String destinationNumber, String destinationNumber1) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
```

```
                                        .build();

                        // Add an entry to the Map object for each number to whom
you want to send a
                        // message.
                        addressMap.put(destinationNumber, addConfig);
                        addressMap.put(destinationNumber1, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

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

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

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

                        SendMessagesResponse response =
pinpoint.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);
                }
        }
}
```

- Einzelheiten zur API finden Sie SendMessagesin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn.

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

Senden Sie eine E-Mail-Nachricht.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

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

// The email body for recipients with non-HTML email clients.
```

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

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

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

const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
```

```
            Data: body_text,
          },
        },
      },
    },
  },
};

const run = async () => {
  try {
    const { MessageResponse } = await pinClient.send(
      new SendMessagesCommand(params),
    );

    if (!MessageResponse) {
      throw new Error("No message response.");
    }

    if (!MessageResponse.Result) {
      throw new Error("No message result.");
    }

    const recipientResult = MessageResponse.Result[toAddress];

    if (recipientResult.StatusCode !== 200) {
      throw new Error(recipientResult.StatusMessage);
    } else {
      console.log(recipientResult.MessageId);
    }
  } catch (err) {
    console.log(err.message);
  }
};

run();
```

Senden Sie eine SMS-Nachricht.

```
// 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.*/
var messageType = "TRANSACTIONAL";

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

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

var senderId = "MySenderID";

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
```

```
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

const run = async () => {
  try {
    const data = await pinClient.send(new SendMessagesCommand(params));
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"],
    );
  } catch (err) {
    console.log(err);
  }
};
run();
```

- Einzelheiten zur API finden Sie SendMessagesin der AWS SDK for JavaScript API-Referenz.

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Senden Sie eine E-Mail-Nachricht.

```
"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"]
    );
  }
});
```

Senden Sie eine SMS-Nachricht.

```
"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"]
    );
  }
});
```

- Einzelheiten zur API finden Sie SendMessagesin der AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
/**
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")
    }
}
```

- API-Details finden Sie SendMessagesin der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Senden Sie eine E-Mail-Nachricht.

```
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()
```

Senden Sie eine SMS-Nachricht.

```
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()
```

Senden Sie eine E-Mail-Nachricht mit einer vorhandenen E-Mail-Vorlage.

```
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()
```

Senden Sie eine Textnachricht mit einer vorhandenen SMS-Vorlage.

```
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()
```

- Einzelheiten zur API finden Sie SendMessagesin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Amazon Pinpoint mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`UpdateEndpoint`** mit einem AWS SDK oder CLI

Das folgende Codebeispiel zeigt, wie es verwendet wird`UpdateEndpoint`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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;
    }
}
```

- Einzelheiten zur API finden Sie UpdateEndpointin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Codebeispiele für Amazon Pinpoint SMS und Voice API mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie Amazon Pinpoint SMS and Voice API mit einem AWS Software Development Kit (SDK) verwenden.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Servicefunktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien und serviceübergreifenden Beispiele sehen.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterAmazon Pinpoint mit einem AWS SDK verwenden. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Codebeispiele

- Aktionen für Amazon Pinpoint SMS und Voice API mithilfe von SDKs AWS
  - Verwendung SendVoiceMessage mit einem AWS SDK oder CLI

Aktionen für Amazon Pinpoint SMS und Voice API mithilfe von SDKs AWS

Die folgenden Codebeispiele zeigen, wie einzelne Amazon Pinpoint SMS- und Voice API-Aktionen mit AWS SDKs durchgeführt werden. Diese Auszüge rufen die Amazon-Pinpoint-SMS- und -Sprachnachrichten-API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der Referenz zur Amazon-Pinpoint-SMS- und -Sprachnachrichten-API.

Beispiele

- Verwendung SendVoiceMessage mit einem AWS SDK oder CLI

Verwendung **SendVoiceMessage** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`SendVoiceMessage`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

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

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

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

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

        public static void main(String[] args) {

                final String usage = """

                                Usage:   <originationNumber> <destinationNumber>
\s

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

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

```
                String originationNumber = args[0];
                String destinationNumber = args[1];
                System.out.println("Sending a voice message");

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

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

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

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

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

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

                        SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()
```

```
            .destinationPhoneNumber(destinationNumber)

            .originationPhoneNumber(originationNumber)
                                                .content(content)
                                                .build();

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

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

- Einzelheiten zur API finden Sie SendVoiceMessagein der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
"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"]);
  }
});
```

- Einzelheiten zur API finden Sie SendVoiceMessagein der AWS SDK for JavaScript API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
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()
```

- Einzelheiten zur API finden Sie SendVoiceMessagein AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Amazon Pinpoint mit einem AWS SDK verwenden Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Sicherheit in Amazon Pinpoint

Cloud-Sicherheit hat AWS höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für Amazon Pinpoint gelten, finden Sie unter AWS Services in Scope by Compliance Program AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation zeigt Ihnen, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Amazon Pinpoint einsetzen können. Die folgenden Themen zeigen Ihnen, wie Sie Amazon Pinpoint konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Amazon Pinpoint Pinpoint-Ressourcen überwachen und sichern können.

Weitere Informationen zu Referenzarchitekturen finden Sie im Handbuch zur Amazon-Pinpoint-resistenten Architektur.

Themen

- Bewährte Methoden für die Sicherheit in Amazon Pinpoint

Datenschutz bei Amazon Pinpoint

Das AWS Modell der gilt für den Datenschutz in Amazon Pinpoint. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der die AWS Cloud gesamte Infrastruktur läuft. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon Pinpoint oder anderen AWS-Services über die Konsole AWS CLI, API oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen

externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Je nachdem, wie Sie den Service konfigurieren und verwenden, kann Amazon Pinpoint die folgenden Arten von persönlichen Daten für Sie oder über Ihre Kunden speichern:

Konfigurationsdaten

Dazu gehören Projektkonfigurationsdaten wie Anmeldeinformationen und Einstellungen, die definieren, wie und wann Amazon Pinpoint Nachrichten über unterstützte Kanäle sendet, sowie die Benutzersegmente, an die Nachrichten gesendet werden. Zum Senden von Nachrichten können diese Daten dedizierte IP-Adressen für E-Mail-Nachrichten, Kurzcodes und Absender-IDs für SMS-Textnachrichten sowie Anmeldeinformationen für die Kommunikation mit Push-Benachrichtigungsdiensten wie Apple Push Notification service (APNs) und Firebase Cloud Messaging (FCM) enthalten.

Benutzer- und Endpunktdaten

Dazu gehören Standard- und benutzerdefinierte Attribute, die Sie zum Speichern und Verwalten von Daten über Benutzer und Endpunkte für ein Amazon-Pinpoint-Projekt verwenden. Ein Attribut kann Informationen über einen bestimmten Benutzer (z. B. den Namen eines Benutzers) oder einen bestimmten Endpunkt für einen Benutzer (z. B. die E-Mail-Adresse eines Benutzers, die Mobiltelefonnummer oder das Token eines Mobilgeräts) speichern. Diese Daten können auch externe Benutzer-IDs enthalten, die Benutzer für ein Amazon-Pinpoint-Projekt mit Benutzern in einem externen System, z. B. einem Kundenbeziehungsmanagement-System, korrelieren. Weitere Informationen dazu, was diese Daten enthalten können, finden Sie in den Schemata für Benutzer und Endpunkte in der Amazon-Pinpoint-API-Referenz.

Analytics-Daten

Dazu gehören Daten für Metriken, die auch als Key Performance Indicators (KPIs), bezeichnet werden, die Einblicke in die Leistung eines Amazon-Pinpoint-Projekts für Bereiche wie Benutzereinbindung und Einkaufsaktivitäten geben. Dazu gehören auch Daten für Metriken, die Einblicke in die Benutzerdemographie für ein Projekt bieten. Die Daten können aus Standard- und benutzerdefinierten Attributen für Benutzer und Endpunkte abgeleitet werden, z. B. aus der Stadt, in der ein Benutzer lebt. Sie können auch von Ereignissen abgeleitet werden, z. B. von Öffnen- und Klickereignissen für die E-Mail-Nachrichten, die Sie für ein Projekt senden.

Importierte Daten

Dies schließt alle Benutzer-, Segmentierungs- und Analysedaten ein, die Sie aus externen Quellen hinzufügen oder importieren und in Amazon Pinpoint verwenden. Ein Beispiel ist etwa

eine JSON-Datei, die Sie in Amazon Pinpoint importieren (direkt über die Konsole oder aus einem Amazon-S3-Bucket), um ein statisches Segment zu erstellen. Weitere Beispiele sind Endpunktdaten, die Sie programmgesteuert hinzufügen, um ein dynamisches Segment zu erstellen, Endpunktadressen, an die Sie direkte Nachrichten senden, und Ereignisse, an die Sie eine App für die Berichterstellung zu Amazon Pinpoint konfigurieren.

Themen

Datenverschlüsselung

Amazon-Pinpoint-Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Wenn Sie Daten an Amazon Pinpoint senden, werden die Daten beim Empfang verschlüsselt und gespeichert. Wenn Sie Daten aus Amazon Pinpoint abrufen, werden die Daten mithilfe der aktuellen Sicherheitsprotokolle an Sie übertragen.

Verschlüsselung im Ruhezustand

Amazon Pinpoint verschlüsselt alle Daten, die für Sie gespeichert werden. Dazu gehören Konfigurationsdaten, Benutzer- und Endpunktdaten, Analysedaten und alle Daten, die Sie in Amazon Pinpoint hinzufügen oder importieren. Um Ihre Daten zu verschlüsseln, verwendet Amazon Pinpoint interne AWS Key Management Service (AWS KMS) Schlüssel, die dem Service gehören und in Ihrem Namen verwaltet werden. Diese Schlüssel werden regelmäßig rotiert. Informationen dazu AWS KMS finden Sie im AWS Key Management Service Entwicklerhandbuch.

Verschlüsselung während der Übertragung

Amazon Pinpoint verwendet HTTPS und Transport Layer Security (TLS) 1.2 oder höher, um mit Ihren Clients und Anwendungen zu kommunizieren. Für die Kommunikation mit anderen AWS Diensten verwendet Amazon Pinpoint HTTPS und TLS 1.2. Wenn Sie Amazon Pinpoint Pinpoint-Ressourcen mithilfe der Konsole, eines AWS SDK oder des erstellen und verwalten, ist die AWS Command Line Interface gesamte Kommunikation außerdem mit HTTPS und TLS 1.2 gesichert.

Schlüsselverwaltung

Um Ihre Amazon Pinpoint-Daten zu verschlüsseln, verwendet Amazon Pinpoint interne AWS KMS Schlüssel, die dem Service gehören und in Ihrem Namen verwaltet werden. Diese Schlüssel werden regelmäßig rotiert. Sie können Ihre eigenen AWS KMS oder andere Schlüssel nicht bereitstellen und verwenden, um Daten zu verschlüsseln, die Sie in Amazon Pinpoint speichern.

Richtlinie für den Datenverkehr zwischen Netzwerken

Datenschutz im Netzwerkverkehr bezieht sich auf die Sicherung von Verbindungen und Datenverkehr zwischen Amazon Pinpoint und Ihren lokalen Clients und Anwendungen sowie zwischen Amazon Pinpoint und anderen AWS Ressourcen in derselben Region. AWS Die folgenden Features und Praktiken können Ihnen dabei helfen, den Schutz des Netzwerkverkehrs für Amazon Pinpoint sicherzustellen.

Datenverkehr zwischen Amazon Pinpoint und On-Premises-Clients und -Anwendungen

Um eine private Verbindung zwischen Amazon Pinpoint und On-Premises-Clients und -Anwendungen in Ihrem Netzwerk herzustellen, können Sie AWS Direct Connect verwenden. Auf diese Weise können Sie Ihr Netzwerk mit einem AWS Direct Connect -Standort verbinden, indem Sie ein Standard-Glasfaser-Ethernet-Kabel verwenden. Ein Ende des Kabels ist mit Ihrem Router verbunden. Das andere Ende ist mit einem Router verbunden. AWS Direct Connect Weitere Informationen finden Sie unter Was ist AWS Direct Connect? im AWS Direct Connect -Benutzerhandbuch.

Um den Zugriff auf Amazon Pinpoint über veröffentlichte APIs zu sichern, empfehlen wir Ihnen, die Amazon-Pinpoint-Anforderungen für API-Aufrufe einzuhalten. Amazon Pinpoint verlangt von Kunden die Verwendung von Transport Layer Security (TLS) 1.2 oder höher. Clients müssen außerdem Cipher Suites mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Darüber hinaus müssen Anfragen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert werden, der einem AWS Identity and Access Management (IAM-) Prinzipal für Ihr AWS Konto zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Verkehr zwischen Amazon Pinpoint und anderen Ressourcen AWS

Um die Kommunikation zwischen Amazon Pinpoint und anderen AWS Ressourcen in derselben AWS Region zu sichern, verwendet Amazon Pinpoint standardmäßig HTTPS und TLS 1.2.

Erstellen eines Schnittstellen-VPC-Endpunkts für Amazon Pinpoint

Sie können eine private Verbindung zwischen Ihrer Virtual Private Cloud (VPC) und einem Endpunkt in Amazon Pinpoint herstellen, indem Sie einen Schnittstellen-VPC-Endpunkt erstellen.

Schnittstellenendpunkte werden von einer Technologie unterstützt AWS PrivateLink, mit der Sie privat auf Amazon Pinpoint Pinpoint-APIs zugreifen können, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder. AWS Direct Connect Die Instances in VPC benötigen für die Kommunikation mit Amazon-Pinpoint-APIs keine öffentlichen IP-Adressen, die mit AWS PrivateLink integriert sind.

Weitere Informationen finden Sie im AWS PrivateLink -Handbuch.

Erstellen von Schnittstellen-VPC-Endpunkten

Sie können einen Schnittstellenendpunkt entweder mit der Amazon VPC-Konsole oder mit AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie im AWS PrivateLink Handbuch unter Erstellen eines Schnittstellenendpunkts.

Amazon Pinpoint unterstützt die folgenden Servicenamen:

- `com.amazonaws.region.pinpoint`
- `com.amazonaws.region.pinpoint-sms-voice-v2`

Wenn Sie privates DNS für einen Schnittstellenendpunkt aktivieren, können Sie API-Anfragen an Amazon Pinpoint stellen, indem Sie den Standard-DNS-Namen für verwenden AWS-Region, `com.amazonaws.us-east-1.pinpoint` zum Beispiel. Weitere Informationen finden Sie unter DNS-Hostnamen im AWS PrivateLink -Benutzerhandbuch.

Eine Liste aller Regionen und Endpunkte, in denen Amazon Pinpoint derzeit verfügbar ist, finden Sie unter AWS -Service-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

Erstellen einer VPC-Endpunktrichtlinie

Sie können eine Endpunktrichtlinie an Ihren VPC-Endpunkt anhängen, der den Zugriff steuert. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter Steuern des Zugriffs auf Services mit Endpunktrichtlinien im AWS PrivateLink -Leitfaden.

Beispiele für VPC-Endpunktrichtlinien

Die folgende VPC-Endpunktrichtlinie gewährt Zugriff auf die aufgelisteten Aktionen in Amazon Pinpoint für alle Prinzipale auf allen Ressourcen.

```
{
"Statement": [
    {
      "Principal": "*",
      "Action": [
        "mobiletargeting:CreateCampaign",
        "mobiletargeting:CreateApp",
        "mobiletargeting:DeleteApp",
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Identitäts- und Zugriffsverwaltung für Amazon Pinpoint

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Amazon-Pinpoint-Ressourcen zu nutzen. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien

- Funktionsweise von Amazon Pinpoint mit IAM
- Amazon-Pinpoint-Aktionen für IAM-Richtlinien
- Beispiele für identitätsbasierte Amazon-Pinpoint-Richtlinien
- IAM-Rollen für allgemeine Amazon-Pinpoint-Aufgaben
- Fehlerbehebung der Identitäts- und Zugriffsverwaltung für Amazon Pinpoint

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon Pinpoint ausführen.

Service-Benutzer – wenn Sie den Amazon-Pinpoint-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie zur Ausführung von Aufgaben weitere Amazon-Pinpoint-Features verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Fuktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie auf ein Feature in Amazon Pinpoint nicht zugreifen können, siehe Fehlerbehebung der Identitäts- und Zugriffsverwaltung für Amazon Pinpoint.

Service-Administrator – wenn Sie in Ihrem Unternehmen für die Amazon-Pinpoint-Ressourcen zuständig sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon Pinpoint. Ihre Aufgabe besteht darin, zu bestimmen, auf welche Amazon-Pinpoint-Features und -Ressourcen Ihre Service-Nutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit Amazon Pinpoint verwenden kann, finden Sie unter Funktionsweise von Amazon Pinpoint mit IAM.

IAM-Administrator – wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht Details darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon Pinpoint erstellen können. Beispiele für identitätsbasierte Amazon-Pinpoint-Richtlinien, die Sie in IAM verwenden können, finden Sie unter Beispiele für identitätsbasierte Amazon-Pinpoint-Richtlinien.

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center - Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-

Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb von Ihnen AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM

erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

- Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

- Auf Amazon EC2 ausgeführte Anwendungen — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Amazon Pinpoint unterstützt die Verwendung identitätsbasierter Richtlinien zur Steuerung des Zugriffs auf Amazon-Pinpoint-Ressourcen.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen

in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Amazon Pinpoint unterstützt die Verwendung identitätsbasierter Richtlinien zur Steuerung des Zugriffs auf Amazon-Pinpoint-Ressourcen.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Amazon Pinpoint unterstützt die Verwendung von ACLs zur Steuerung des Zugriffs auf Amazon-Pinpoint-Ressourcen nicht.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service Control Policies (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS

Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations -Benutzerhandbuch.

- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Amazon Pinpoint unterstützt die Verwendung identitätsbasierter Richtlinien zur Steuerung des Zugriffs auf Amazon-Pinpoint-Ressourcen.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

Funktionsweise von Amazon Pinpoint mit IAM

Um Amazon Pinpoint verwenden zu können, benötigen Benutzer in Ihrem AWS Konto Berechtigungen, die es ihnen ermöglichen, Analysedaten einzusehen, Projekte zu erstellen, Benutzersegmente zu definieren, Kampagnen bereitzustellen und vieles mehr. Wenn Sie eine Mobil- oder Web-App in Amazon Pinpoint integrieren, benötigen Benutzer Ihrer App auch Zugriff auf Amazon Pinpoint. Dieser Zugriff ermöglicht es Ihrer App, Endpunkte zu registrieren und Nutzungsdaten an Amazon Pinpoint zu melden. Um Zugriff auf Amazon Pinpoint Pinpoint-Funktionen zu gewähren, erstellen Sie AWS Identity and Access Management (IAM-) Richtlinien, die Amazon Pinpoint Pinpoint-Aktionen für IAM-Identitäten oder Amazon Pinpoint Pinpoint-Ressourcen zulassen.

IAM ist ein Service, der Administratoren hilft, den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Richtlinien beinhalten Anweisungen, die bestimmte Aktionen, die Benutzer auf bestimmten

Ressourcen durchführen können, zulassen oder verweigern. Amazon Pinpoint stellt eine Reihe von Aktionen bereit, die Sie in IAM-Richtlinien verwenden können, um abgestufte Berechtigungen für Amazon-Pinpoint-Benutzer festzulegen. Sie können den angemessenen Zugriffsgrad für Amazon Pinpoint gewähren, ohne übermäßig lockere Richtlinien zu erstellen, die möglicherweise wichtige Daten gefährden oder Ihre Ressourcen beeinträchtigen könnten. Beispielsweise können Sie einem Amazon-Pinpoint-Administrator uneingeschränkten Zugriff und Personen, die nur Zugriff auf ein spezifisches Projekt benötigen, Lesezugriff erteilen.

Bevor Sie mit IAM den Zugriff auf Amazon Pinpoint verwalten können, sollten Sie sich darüber informieren, welche IAM-Features Sie mit Amazon Pinpoint verwenden können. Einen allgemeinen Überblick darüber, wie Amazon Pinpoint und andere AWS Services mit IAM zusammenarbeiten, finden Sie im AWS IAM-Benutzerhandbuch unter Services, die mit IAM funktionieren.

Themen

Identitätsbasierte Amazon-Pinpoint-Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Amazon Pinpoint unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Aktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen,

die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Dies bedeutet, dass Richtlinienaktionen steuern, was Benutzer auf der Amazon-Pinpoint-Konsole tun können. Sie kontrollieren auch, was Benutzer programmgesteuert tun können, indem sie die AWS SDKs, die AWS Command Line Interface (AWS CLI) oder die Amazon Pinpoint Pinpoint-APIs direkt verwenden.

Richtlinienaktionen in Amazon Pinpoint verwenden die folgenden Präfixe:

- **`mobiletargeting`**: Für Aktionen, die von der Amazon-Pinpoint-API abgeleitet werden, die die primäre API für Amazon Pinpoint ist.
- **`sms-voice`**: Für Aktionen, die von der Amazon Pinpoint-SMS und -Sprachnachricht-API abgeleitet werden, einer ergänzenden API, die erweiterte Optionen für die Verwendung und Verwaltung der SMS- und Sprachkanäle in Amazon Pinpoint bietet.

Um beispielsweise jemandem die Berechtigung zum Anzeigen von Informationen über alle Segmente für ein Projekt zu erteilen, wobei es sich um eine Aktion handelt, die der `GetSegments`-Operation in der Amazon-Pinpoint-API entspricht, schließen Sie die `mobiletargeting:GetSegments`-Aktion in ihre Richtlinie ein. Richtlinienanweisungen müssen entweder ein – `Action`oder ein `NotAction`-Element enthalten. Amazon Pinpoint definiert eine eigene Gruppe von Aktionen, die Aufgaben beschreiben, die Benutzer damit durchführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "mobiletargeting:action1",
      "mobiletargeting:action2"
```

Sie können auch mehrere Aktionen mittels Platzhaltern (*) angeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Get` beginnen, einschließlich der folgenden Aktion:

```
"Action": "mobiletargeting:Get*"
```

Als bewährte Methode sollten Sie jedoch Richtlinien erstellen, die dem Prinzip der geringsten Rechte folgen. Mit anderen Worten: Sie sollten Richtlinien erstellen, die nur die Berechtigungen enthalten, die zum Ausführen einer bestimmten Aktion erforderlich sind.

Eine vollständige Liste der Amazon-Pinpoint-Aktionen, die Sie in IAM-Richtlinien verwenden können, finden Sie unter Amazon-Pinpoint-Aktionen für IAM-Richtlinien.

Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Beispielsweise ruft die `mobiletargeting:GetSegments`-Aktion Informationen über alle Segmente ab, die einem bestimmten Amazon-Pinpoint-Projekt zugeordnet sind. Sie identifizieren ein Projekt mit einem ARN im folgenden Format:

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

Weitere Informationen zum Format von ARNs finden Sie unter Amazon-Ressourcennamen (ARNs) im Allgemeine AWS-Referenz.

In IAM-Richtlinien können Sie ARNs für die folgenden Amazon-Pinpoint-Ressourcentypen angeben:

- Kampagnen
- Journeys

- Nachrichtenvorlagen (in manchen Kontexten als Vorlagen bezeichnet)
- Projekte (in manchen Kontexten als Apps oder Anwendungen bezeichnet)
- Empfehlungsmodelle (in manchen Kontexten als Empfehlungsgeber bezeichnet)
- Segmente

Um beispielsweise eine Richtlinienanweisung für das Projekt mit der Projekt-ID `810c7aab86d42fb2b56c8c966example` zu erstellen, verwenden Sie den folgenden ARN:

```
"Resource": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

Um alle Projekte anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

Einige Amazon-Pinpoint-Aktionen, z. B. zum Erstellen von Ressourcen, können auf bestimmten Ressourcen nicht ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden:

```
"Resource": "*"
```

In IAM-Richtlinien können Sie auch ARNs für die folgenden Amazon-Pinpoint-SMS- und Sprachnachricht-Ressourcentypen angeben:

- Konfigurationssatz
- Abmeldeliste
- Telefonnummer
- Pool
- Sender-ID

Um beispielsweise eine Richtlinienerklärung für eine Telefonnummer mit der Rufnummern-ID zu erstellen, verwendet `phone-12345678901234567890123456789012` den folgenden ARN:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-12345678901234567890123456789012"
```

Um alle Telefonnummern anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie einen Platzhalter (*) anstelle der Rufnummer-ID:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

Einige SMS- und Sprachnachricht-Aktionen von Amazon Pinpoint werden nicht für eine bestimmte Ressource ausgeführt, z. B. für die Verwaltung von Einstellungen auf Kontoebene, wie Ausgabenlimits. In diesen Fällen müssen Sie den Platzhalter (*) verwenden:

```
"Resource": "*"
```

Bei einigen Amazon-Pinpoint-API-Aktionen sind mehrere Ressourcen beteiligt. Beispielsweise kann die `TagResource`-Aktion mehreren Projekten ein Tag hinzufügen. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie die ARNs durch Kommata voneinander:

```
"Resource": [
      "resource1",
      "resource2"
```

Eine Liste der Amazon-Pinpoint-Ressourcentypen und ihrer ARNs finden Sie unter Von Amazon Pinpoint definierte Ressourcen im IAM-Benutzerhandbuch. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von Amazon Pinpoint definierte Aktionen im IAM-Benutzerhandbuch.

Bedingungsschlüssel

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation

aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Amazon Pinpoint definiert einen eigenen Satz von Bedingungsschlüsseln und unterstützt auch einige globale Bedingungsschlüssel. Eine Liste aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch. Eine Liste der Amazon-Pinpoint-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Pinpoint im IAM-Benutzerhandbuch. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon Pinpoint definierte Aktionen im IAM-Benutzerhandbuch.

Beispiele

Beispiele für identitätsbasierte Amazon-Pinpoint-Richtlinien finden Sie unter Beispiele für identitätsbasierte Amazon-Pinpoint-Richtlinien.

Amazon-Pinpoint-Richtlinien auf Basis von Ressourcenberechtigungen

Richtlinien auf Basis von Ressourcenberechtigungen sind JSON-Richtliniendokumente, die angeben, welche Aktionen ein bestimmter Auftraggeber auf einer Amazon-Pinpoint-Ressource durchführen kann und unter welchen Bedingungen. Amazon Pinpoint unterstützt Richtlinien auf Basis von Ressourcenberechtigungen für Kampagnen, Journeys, Nachrichtenvorlagen (Vorlagen), Empfehlungsmodelle (Empfehlungen), Projekte (Apps) und Segmente.

Beispiele

Beispiele für ressourcenbasierte Amazon-Pinpoint-Richtlinien finden Sie unter the section called “Beispiele für identitätsbasierte Richtlinien”,

Autorisierung basierend auf Amazon-Pinpoint-Tags

Sie können Tags bestimmten Arten von Amazon-Pinpoint-Ressourcen zuordnen oder Tags in einer Anforderung an Amazon Pinpoint übergeben. Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/${TagKey}`, `aws:RequestTag/${TagKey}`, oder Bedingung `aws:TagKeys` verwenden.

Hinweise zum Markieren von Amazon-Pinpoint-Ressourcen, einschließlich einer IAM-Beispielrichtlinie, finden Sie unter Markieren von Amazon-Pinpoint-Ressourcen.

Amazon-Pinpoint-IAM-Rollen

Eine IAM-Rolle ist eine Entität in Ihrem AWS -Konto mit spezifischen Berechtigungen.

Verwenden temporärer Anmeldeinformationen mit Amazon Pinpoint

Sie können temporäre Anmeldeinformationen verwenden, um sich mit dem Verbund anzumelden, eine IAM-Rolle zu übernehmen oder eine kontoübergreifende Rolle zu übernehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie API-Operationen AWS Security Token Service (AWS STS) wie AssumeRoleoder GetFederationTokenaufrufen.

Amazon Pinpoint unterstützt die Verwendung temporärer Anmeldeinformationen.

Service-verknüpfte Rollen

Mit Diensten verknüpfte Rollen ermöglichen es AWS Diensten, auf Ressourcen in anderen Diensten zuzugreifen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Amazon Pinpoint verwendet keine serviceverknüpften Rollen.

Servicerollen

Dieses Feature ermöglicht einem Service das Annehmen einer Servicerolle in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem IAM-Konto angezeigt und gehören zum Konto. Dies bedeutet, dass ein IAM-Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Dienstes beeinträchtigen.

Amazon Pinpoint unterstützt die Verwendung von Servicerollen.

Amazon-Pinpoint-Aktionen für IAM-Richtlinien

Um den Zugriff auf Amazon Pinpoint Pinpoint-Ressourcen in Ihrem AWS Konto zu verwalten, können Sie Amazon Pinpoint Pinpoint-Aktionen zu AWS Identity and Access Management (IAM-) Richtlinien hinzufügen. Mithilfe von Aktionen in Richtlinien können Sie steuern, was Benutzer auf der Amazon-Pinpoint-Konsole tun können. Sie können auch steuern, was Benutzer programmgesteuert tun können, indem Sie die AWS SDKs, die AWS Command Line Interface (AWS CLI) oder die Amazon Pinpoint Pinpoint-APIs direkt verwenden.

In einer Richtlinie geben Sie jede Aktion mit dem richtigen Amazon-Pinpoint-Namespace gefolgt von einem Doppelpunkt und dem Namen der Aktion an, z. B. `GetSegments`. Die meisten Aktionen entsprechen einer Anforderung bei der Amazon-Pinpoint-API unter Verwendung einer bestimmten URI und HTTP-Methode. Wenn Sie beispielsweise die `mobiletargeting:GetSegments`-Aktion in der Richtlinie eines Benutzers zulassen, kann der Benutzer Informationen über alle Segmente eines Projekts abrufen, indem er eine HTTP-GET-Anforderung an den `/apps/projectId/segments`-URI sendet. Diese Richtlinie ermöglicht es dem Benutzer auch, diese Informationen auf der Konsole anzuzeigen und diese Informationen mithilfe eines AWS SDK oder des abzurufen. AWS CLI

Jede Aktion wird auf einer bestimmten Amazon-Pinpoint-Ressource ausgeführt, die Sie in einer Richtlinienanweisung mit dem entsprechenden Amazon-Ressourcennamen (ARN) identifizieren. Beispiel: Die Aktion `mobiletargeting:GetSegments` wird für ein bestimmtes Projekt ausgeführt, das Sie mit dem ARN `arn:aws:mobiletargeting:region:accountId:apps/projectId` identifizieren.

In diesem Thema werden Amazon-Pinpoint-Aktionen identifiziert, die Sie IAM-Richtlinien für Ihr AWS -Konto hinzufügen können. Beispiele, die veranschaulichen, wie Sie Aktionen in Richtlinien zum Verwalten des Zugriffs auf Amazon-Pinpoint-Ressourcen verwenden können, finden Sie unter Beispiele für identitätsbasierte Amazon-Pinpoint-Richtlinien.

Themen

- API-Aktionen für Amazon Pinpoint
- Amazon-Pinpoint-SMS- und -Sprachnachrichten-API-Aktionen, Version 1

API-Aktionen für Amazon Pinpoint

In diesem Abschnitt werden Aktionen für Features identifiziert, die über die Amazon-Pinpoint-API verfügbar sind, bei der es sich um die primäre API für Amazon Pinpoint handelt. Weitere Informationen zur Verwendung dieser API finden Sie in der API-Referenz zu Amazon Pinpoint.

Kategorien:

- Analysen und Metriken
- Kampagnen
- Kanäle
- Endpunkte
- Ereignis-Streams
- Ereignisse
- Exportaufträge
- Importaufträge
- Journeys
- Nachrichtenvorlagen
- Nachrichten
- Einmalpasswörter
- Telefonnummernüberprüfung
- Projekte
- Empfehlungsmodelle
- Segmente
- Tags
- Benutzer

Analysen und Metriken

Die folgenden Berechtigungen beziehen sich auf die Anzeige von Analysedaten in der Amazon-Pinpoint-Konsole. Sie beziehen sich auch auf das Abrufen (Abfragen) aggregierter Daten für Standardmetriken, auch als Key Performance Indicators (KPIs) bezeichnet, die für Projekte, Kampagnen und Journeys gelten.

`mobiletargeting:GetReports`

Zeigen Sie Analysedaten in der Amazon-Pinpoint-Konsole an. Diese Berechtigung ist auch erforderlich, um Segmente mit benutzerdefinierten Attributen mithilfe der Amazon-Pinpoint-Konsole zu erstellen. Es ist auch erforderlich, eine Schätzung der Größe eines Segments in der Amazon-Pinpoint-Konsole zu erhalten.

- URI – nicht zutreffend
- Methode – Nicht zutreffend
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:GetApplicationDateRangeKpi`

Abrufen (Abfragen) aggregierter Daten für eine Standardanwendungsmetrik. Dies ist eine Metrik, die für alle Kampagnen oder transaktionalen Nachrichten gilt, die mit einem Projekt verbunden sind.

- URI – `/apps/projectId/kpis/daterange/kpi-name`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/kpis/daterange/kpi-name`

`mobiletargeting:GetCampaignDateRangeKpi`

Abrufen (Abfragen) aggregierter Daten für eine Standardkampagnenmetrik. Dies ist eine Metrik, die für eine einzelne Kampagne gilt.

- URI – `/apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name`

`mobiletargeting:GetJourneyDateRangeKpi`

Abrufen (Abfragen) aggregierter Daten für eine Standard-Journey-Engagement-Metrik. Dabei handelt es sich um eine Interaktionsmetrik, die für eine einzelne Journey gilt – zum Beispiel die Anzahl der Nachrichten, die von Teilnehmern für alle Aktivitäten einer Journey geöffnet wurden.

- URI – `/apps/projectId/journeys/journeyId/kpis/daterange/kpi-name`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/kpis/daterange/kpi-name`

`mobiletargeting:GetJourneyExecutionMetrics`

Rufen Sie aggregierte Daten für Standardausführungsmetriken ab, die für eine individuelle Journey gelten, z. B. die Anzahl der Teilnehmer, die aktiv alle Aktivitäten einer Journey durchlaufen.

- URI – /apps/*projectId*/journeys/*journeyId*/execution-metrics
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/execution-metrics

mobiletargeting:GetJourneyExecutionActivityMetrics

Rufen Sie aggregierte Daten für Standardausführungsmetriken ab, die für eine individuelle Journey gelten, z. B. die Anzahl der Teilnehmer, die eine Aktivität begonnen oder abgeschlossen haben.

- URI – /apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics

Kampagnen

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Kampagnen in Ihrem Amazon-Pinpoint-Konto.

mobiletargeting:CreateCampaign

Erstellen einer Kampagne für ein Projekt

- URI – /apps/*projectId*/campaigns
- Methode – POST
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns

mobiletargeting:DeleteCampaign

Löschen einer bestimmten Kampagne

- URI – /apps/*projectId*/campaigns/*campaignId*
- Methode – DELETE
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaign

Abrufen von Informationen zu einer bestimmten Kampagne

- URI – /apps/*projectId*/campaigns/*campaignId*
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignActivities

Abrufen von Informationen zu den Aktivitäten, die von einer Kampagne durchgeführt werden

- URI – /apps/*projectId*/campaigns/*campaignId*/activities
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaigns

Abrufen von Informationen zu allen Kampagnen für ein Projekt

- URI – /apps/*projectId*/campaigns
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetCampaignVersion

Abrufen von Informationen zu einer bestimmten Kampagnenversion

- URI – /apps/*projectId*/campaigns/*campaignId*/versions/*versionId*
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignVersions

Abrufen von Informationen zu den aktuellen und vorherigen Versionen einer Kampagne

- URI – /apps/*projectId*/campaigns/*campaignId*/versions
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

`mobiletargeting:UpdateCampaign`

Aktualisieren einer bestimmten Kampagne

- URI – `/apps/projectId/campaigns/campaignId`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId`

Kanäle

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Kanälen in Ihrem Amazon-Pinpoint-Konto. In Amazon Pinpoint beziehen sich Kanäle auf die Methoden, mit denen Sie Ihre Kunden kontaktieren, z. B. durch Senden von E-Mails, SMS-Nachrichten oder Push-Benachrichtigungen.

`mobiletargeting:DeleteAdmChannel`

Deaktivieren des Amazon Device Messaging-Kanals (ADM) für ein Projekt

- URI – `/apps/projectId/channels/adm`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

`mobiletargeting:GetAdmChannel`

Abrufen von Informationen zum ADM-Kanal für ein Projekt

- URI – `/apps/projectId/channels/adm`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

`mobiletargeting:UpdateAdmChannel`

Aktivieren oder Aktualisieren des ADM-Kanals für ein Projekt

- URI – `/apps/projectId/channels/adm`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/adm`

mobiletargeting:DeleteApnsChannel

Deaktivieren des APNs-Kanals (Apple Push Notification-Service) für ein Projekt

- URI – /apps/*projectId*/channels/apns
- Methode – DELETE
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:GetApnsChannel

Abrufen von Informationen zum APNs-Kanal für ein Projekt

- URI – /apps/*projectId*/channels/apns
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:UpdateApnsChannel

Aktivieren oder Aktualisieren des APNs-Kanals für ein Projekt

- URI – /apps/*projectId*/channels/apns
- Methode – PUT
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:DeleteApnsSandboxChannel

Deaktivieren des APNs-Sandbox-Kanals für ein Projekt

- URI – /apps/*projectId*/channels/apns_sandbox
- Methode – DELETE
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:GetApnsSandboxChannel

Abrufen von Informationen zum APNs-Sandbox-Kanal für ein Projekt

- URI – /apps/*projectId*/channels/apns_sandbox
- Methode – GET

- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:UpdateApnsSandboxChannel`

Aktivieren oder Aktualisieren des APNs-Sandbox-Kanals für ein Projekt

- URI – `/apps/projectId/channels/apns_sandbox`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:DeleteApnsVoipChannel`

Deaktivieren des APNs-VoIP-Kanals für ein Projekt

- URI – `/apps/projectId/channels/apns_voip`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:GetApnsVoipChannel`

Abrufen von Informationen zum APNs-VoIP-Kanal für ein Projekt

- URI – `/apps/projectId/channels/apns_voip`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:UpdateApnsVoipChannel`

Aktivieren oder Aktualisieren des APNs-VoIP-Kanals für ein Projekt

- URI – `/apps/projectId/channels/apns_voip`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:DeleteApnsVoipSandboxChannel`

Deaktivieren des APNs-VoIP-Sandbox-Kanals für ein Projekt

- URI – `/apps/projectId/channels/apns_voip_sandbox`

- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

`mobiletargeting:GetApnsVoipSandboxChannel`

Abrufen von Informationen zum APNs-VoIP-Sandbox-Kanal für ein Projekt

- URI – `/apps/projectId/channels/apns_voip_sandbox`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

`mobiletargeting:UpdateApnsVoipSandboxChannel`

Aktivieren oder Aktualisieren des APNs-VoIP-Sandbox-Kanals für ein Projekt

- URI – `/apps/projectId/channels/apns_voip_sandbox`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip_sandbox`

`mobiletargeting:DeleteBaiduChannel`

Deaktivieren des Baidu Cloud Push-Kanals für ein Projekt

- URI – `/apps/projectId/channels/baidu`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/baidu`

`mobiletargeting:GetBaiduChannel`

Abrufen von Informationen zum Baidu Cloud Push-Kanal für ein Projekt

- URI – `/apps/projectId/channels/baidu`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/baidu`

`mobiletargeting:UpdateBaiduChannel`

Aktivieren oder Aktualisieren des Baidu Cloud Push-Kanals für ein Projekt

- URI – /apps/*projectId*/channels/baidu
- Methode – PUT
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:DeleteEmailChannel

Deaktivieren des E-Mail-Kanals für ein Projekt

- URI – /apps/*projectId*/channels/email
- Methode – DELETE
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:GetEmailChannel

Abrufen von Informationen zum E-Mail-Kanal für ein Projekt

- URI – /apps/*projectId*/channels/email
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:UpdateEmailChannel

Aktivieren oder Aktualisieren des E-Mail-Kanals für ein Projekt

- URI – /apps/*projectId*/channels/email
- Methode – PUT
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:DeleteGcmChannel

Deaktivieren des FCM-Kanals (Firebase Cloud Messaging) für ein Projekt Dieser Kanal ermöglicht es Amazon Pinpoint, Push-Benachrichtigungen über den FCM-Service, der den Google Cloud Messaging (GCM)-Service ersetzt, an eine Android-App zu senden.

- URI – /apps/*projectId*/channels/gcm
- Methode – DELETE
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:GetGcmChannel

Abrufen von Informationen zum FCM-Kanal für ein Projekt Dieser Kanal ermöglicht es Amazon Pinpoint, Push-Benachrichtigungen über den FCM-Service, der den Google Cloud Messaging (GCM)-Service ersetzt, an eine Android-App zu senden.

- URI – /apps/*projectId*/channels/gcm
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/gcm`

mobiletargeting:UpdateGcmChannel

Aktivieren oder Aktualisieren des FCM-Kanals für ein Projekt Dieser Kanal ermöglicht es Amazon Pinpoint, Push-Benachrichtigungen über den FCM-Service, der den Google Cloud Messaging (GCM)-Service ersetzt, an eine Android-App zu senden.

- URI – /apps/*projectId*/channels/gcm
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/gcm`

mobiletargeting:DeleteSmsChannel

Deaktivieren des SMS-Kanals für ein Projekt

- URI – /apps/*projectId*/channels/sms
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/sms`

mobiletargeting:GetSmsChannel

Abrufen von Informationen zum SMS-Kanal für ein Projekt

- URI – /apps/*projectId*/channels/sms
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/sms`

mobiletargeting:UpdateSmsChannel

Aktivieren oder Aktualisieren des SMS-Kanals für ein Projekt

- URI – /apps/*projectId*/channels/sms
- Methode – PUT
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetChannels

Ruft Informationen über den Verlauf und den Status jedes Kanals für eine Anwendung ab.

- URI – /apps/*application-id*/channels
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/channels

mobiletargeting:DeleteVoiceChannel

Deaktiviert den Sprachkanal für eine Anwendung und löscht alle vorhandenen Einstellungen für den Kanal.

- URI – /apps/*application-id*/channels/voice
- Methode – DELETE
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:GetVoiceChannel

Ruft Informationen über den Status und die Einstellungen jedes Sprachkanals für eine Anwendung ab.

- URI – /apps/*application-id*/channels/voice
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:UpdateVoiceChannel

Aktiviert den Sprachkanal für eine Anwendung oder aktualisiert den Status und die Einstellungen des Sprachkanals für eine Anwendung.

- URI – /apps/*application-id*/channels/voice
- Methode – PUT

- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

Endpunkte

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Endpunkten in Ihrem Amazon-Pinpoint-Konto. In Amazon Pinpoint ist ein Endpunkt ein einzelnes Ziel für Ihre Nachrichten. Ein Endpunkt kann beispielsweise die E-Mail-Adresse, Telefonnummer oder ein Token für ein mobiles Gerät eines Kunden sein.

`mobiletargeting:DeleteEndpoint`

Löschen eines Endpunktes

- URI – `/apps/projectId/endpoints/endpointId`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

`mobiletargeting:GetEndpoint`

Abrufen von Informationen zu einem bestimmten Endpunkt

- URI – `/apps/projectId/endpoints/endpointId`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

`mobiletargeting:RemoveAttributes`

Entfernt ein oder mehrere Attribute desselben Attributtyps von allen Endpunkten, die einer Anwendung zugeordnet sind.

- URI – `apps/application-id/attributes/attribute-type`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/attributes/attribute-type`

`mobiletargeting:UpdateEndpoint`

Erstellen eines Endpunkts oder Aktualisieren der Informationen für einen Endpunkt

- URI – /apps/*projectId*/endpoints/*endpointId*
- Methode – PUT
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:UpdateEndpointsBatch

Erstellen oder Aktualisieren von Endpunkten als Batchvorgang

- URI – /apps/*projectId*/endpoints
- Methode – PUT
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

Ereignis-Streams

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Ereignisstreams für Ihr Amazon-Pinpoint-Konto.

mobiletargeting:DeleteEventStream

Löschen des Ereignis-Streams für ein Projekt

- URI – /apps/*projectId*/eventstream/
- Methode – DELETE
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:GetEventStream

Abrufen von Informationen zum Ereignis-Stream für ein Projekt

- URI – /apps/*projectId*/eventstream/
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:PutEventStream

Erstellen oder Aktualisieren eines Ereignis-Streams für ein Projekt

- URI – /apps/*projectId*/eventstream/

- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

Ereignisse

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Ereignisaufträgen in Ihrem Amazon-Pinpoint-Konto. In Amazon Pinpoint erstellen Sie Importaufträge, um Segmente auf Basis von Endpunktdefinitionen zu schaffen, die in einem Amazon-S3-Bucket gespeichert sind.

`mobiletargeting:PutEvents`

Erstellt ein neues Ereignis, das für Endpunkte aufgezeichnet wird, oder erstellt oder aktualisiert Endpunktdaten, denen bestehende Ereignisse zugeordnet sind.

- URI – `/apps/application-id/events`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/events`

Exportaufträge

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Exportaufträgen in Ihrem Amazon-Pinpoint-Konto. In Amazon Pinpoint erstellen Sie Exportaufträge zum Senden von Informationen über Endpunkte an einen Amazon-S3-Bucket zur Speicherung oder Analyse.

`mobiletargeting:CreateExportJob`

Erstellen Sie einen Exportauftrag für das Exportieren von Endpunktdefinitionen nach Amazon S3.

- URI – `/apps/projectId/jobs/export`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

`mobiletargeting:GetExportJob`

Abrufen von Informationen zu einem bestimmten Exportauftrag für ein Projekt

- URI – `/apps/projectId/jobs/export/jobId`

- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export/jobId`

`mobiletargeting:GetExportJobs`

Abrufen einer Liste aller Exportaufträge für ein Projekt

- URI – `/apps/projectId/jobs/export`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

Importaufträge

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Importaufträgen in Ihrem Amazon-Pinpoint-Konto. In Amazon Pinpoint erstellen Sie Importaufträge, um Segmente auf Basis von Endpunktdefinitionen zu schaffen, die in einem Amazon-S3-Bucket gespeichert sind.

`mobiletargeting:CreateImportJob`

Importieren Sie Endpunktdefinitionen aus Amazon S3 zum Erstellen eines Segments.

- URI – `/apps/projectId/jobs/import`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetImportJob`

Abrufen von Informationen zu einem bestimmten Importauftrag für ein Projekt

- URI – `/apps/projectId/jobs/import/jobId`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/import/jobId`

`mobiletargeting:GetImportJobs`

Abrufen von Informationen zu allen Importaufträgen für ein Projekt

- URI – `/apps/projectId/jobs/import`
- Methode – GET

- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Journeys

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Journeys in Ihrem Amazon-Pinpoint-Konto.

`mobiletargeting:CreateJourney`

Erstellen Sie eine Journey für ein Projekt.

- URI – `/apps/projectId/journeys`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

`mobiletargeting:GetJourney`

Rufen Sie Informationen zu einem bestimmten Ablauf ab.

- URI – `/apps/projectId/journeys/journeyId`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

`mobiletargeting:ListJourneys`

Informationen über alle Journey zu einem Projekt abrufen.

- URI – `/apps/projectId/journeys`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

`mobiletargeting:UpdateJourney`

Aktualisieren Sie die Konfiguration und andere Einstellungen für eine bestimmte Journey.

- URI – `/apps/projectId/journeys/journeyId`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

`mobiletargeting:UpdateJourneyState`

Eine aktive Journey abbrechen.

- URI – `/apps/projectId/journeys/journeyId/state`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/state`

`mobiletargeting:DeleteJourney`

Löschen Sie einen spezifischen Ablauf.

- URI – `/apps/projectId/journeys/journeyId`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

Nachrichtenvorlagen

Die folgenden Berechtigungen beziehen sich auf das Erstellen und Verwalten von Nachrichtenvorlagen für Ihr Amazon-Pinpoint-Konto. Eine Nachrichtenvorlage ist eine Gruppe von Inhalten, die Sie in Nachrichten erstellen, speichern und dann wiederverwenden können, die Sie für jedes Ihrer Amazon-Pinpoint-Projekte senden.

`mobiletargeting:ListTemplates`

Abrufen von Informationen zu allen Nachrichtenvorlagen, die Ihrem Amazon-Pinpoint-Konto zugeordnet sind

- URI – `/templates`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates`

`mobiletargeting:ListTemplateVersions`

Abrufen von Informationen über alle Versionen einer bestimmten Nachrichtenvorlage

- URI – `/templates/template-name/template-type/versions`
- Methode – GET

- Ressourcen-ARN – nicht zutreffend

mobiletargeting:UpdateTemplateActiveVersion

Bestimmen einer bestimmten Version einer Nachrichtenvorlage als aktive Version der Vorlage

- URI – /templates/*template-name*/*template-type*/active-version
- Methode – GET
- Ressourcen-ARN – nicht zutreffend

mobiletargeting:GetEmailTemplate

Abrufen von Informationen zu einer Nachrichtenvorlage für Nachrichten, die über den E-Mail-Kanal gesendet werden

- URI – /templates/*template-name*/email
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:CreateEmailTemplate

Erstellen einer Nachrichtenvorlage für Nachrichten, die über den E-Mail-Kanal gesendet werden

- URI – /templates/*template-name*/email
- Methode – POST
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:UpdateEmailTemplate

Aktualisieren einer vorhandenen Nachrichtenvorlage für Nachrichten, die über den E-Mail-Kanal gesendet werden

- URI – /templates/*template-name*/email
- Methode – PUT
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

`mobiletargeting:DeleteEmailTemplate`

Löschen einer Nachrichtenvorlage für Nachrichten, die über den E-Mail-Kanal gesendet wurden

- URI – `/templates/template-name/email`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:GetPushTemplate`

Abrufen von Informationen zu einer Nachrichtenvorlage für Nachrichten, die über einen Push-Benachrichtigungskanal gesendet werden

- URI – `/templates/template-name/push`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:CreatePushTemplate`

Erstellen einer Nachrichtenvorlage für Nachrichten, die über einen Push-Benachrichtigungskanal gesendet werden

- URI – `/templates/template-name/push`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:UpdatePushTemplate`

Aktualisieren einer vorhandenen Nachrichtenvorlage für Nachrichten, die über einen Push-Benachrichtigungskanal gesendet werden

- URI – `/templates/template-name/push`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:DeletePushTemplate`

Löschen einer Nachrichtenvorlage für Nachrichten, die über einen Push-Benachrichtigungskanal gesendet wurden

- URI – `/templates/template-name/push`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:GetSmsTemplate`

Abrufen von Informationen über eine Nachrichtenvorlage für Nachrichten, die über den SMS-Kanal gesendet werden

- URI – `/templates/template-name/sms`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:CreateSmsTemplate`

Erstellen einer Nachrichtenvorlage für Nachrichten, die über den SMS-Kanal gesendet werden

- URI – `/templates/template-name/sms`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:UpdateSmsTemplate`

Aktualisieren einer vorhandenen Nachrichtenvorlage für Nachrichten, die über den SMS-Kanal gesendet werden

- URI – `/templates/template-name/sms`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

mobiletargeting:DeleteSmsTemplate

Löschen einer Nachrichtenvorlage für Nachrichten, die über den SMS-Kanal gesendet wurden

- URI – /templates/*template-name*/sms
- Methode – DELETE
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:GetVoiceTemplate

Abrufen von Informationen über eine Nachrichtenvorlage für Nachrichten, die über den Sprach-Kanal gesendet werden

- URI – /templates/*template-name*/voice
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:CreateVoiceTemplate

Erstellen einer Nachrichtenvorlage für Nachrichten, die über den Sprach-Kanal gesendet werden

- URI – /templates/*template-name*/voice
- Methode – POST
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:UpdateVoiceTemplate

Aktualisieren einer vorhandenen Nachrichtenvorlage für Nachrichten, die über den Sprach-Kanal gesendet werden

- URI – /templates/*template-name*/voice
- Methode – PUT
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:DeleteVoiceTemplate

Löschen einer Nachrichtenvorlage für Nachrichten, die über den Sprach-Kanal gesendet wurden

- URI – /templates/*template-name*/voice

- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

Nachrichten

Die folgenden Berechtigungen beziehen sich auf das Senden von Nachrichten und Push-Benachrichtigungen von Ihrem Amazon-Pinpoint-Konto aus. Sie können die Operationen `SendMessages` und `SendUsersMessages` zum Senden von Nachrichten an bestimmte Endpunkte verwenden, ohne zunächst Segmente und Kampagnen zu erstellen.

`mobiletargeting:SendMessages`

Senden einer Nachricht oder Push-Benachrichtigung an bestimmte Endpunkte

- URI – `/apps/projectId/messages`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

`mobiletargeting:SendUsersMessages`

Senden einer Nachricht oder Push-Benachrichtigung an alle Endpunkte, die mit einer bestimmten Benutzer-ID verknüpft sind

- URI – `/apps/projectId/users-messages`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/messages`

Einmalpasswörter

Die folgenden Berechtigungen beziehen sich auf das Senden und Überprüfen von Einmalpasswörtern (one-time passwords, OTPs) in Amazon Pinpoint.

`mobiletargeting:SendOTPMessage`

Senden Sie eine Textnachricht, die ein Einmalpasswort enthält.

- URI – `/apps/projectId/otp`

- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/otp`

`mobiletargeting:VerifyOTPMessage`

Überprüfen Sie die Gültigkeit eines Einmalpassworts (OTP), das mit der SendOTPMessage-Operation generiert wurde.

- URI – `/apps/projectId/verify-otp`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/verify-otp`

Telefonnummernüberprüfung

Die folgenden Berechtigungen stehen im Zusammenhang mit der Verwendung des Services zur Telefonnummernüberprüfung in Amazon Pinpoint.

`mobiletargeting:PhoneNumberValidate`

Abrufen von Informationen zu einer Telefonnummer

- URI – `/phone/number/validate`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:phone/number/validate`

Projekte

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Projekten in Ihrem Amazon-Pinpoint-Konto. Ursprünglich wurden Projekte als Anwendungen bezeichnet. Bei diesen Operationen ist eine Amazon-Pinpoint-Anwendung dasselbe wie ein Amazon-Pinpoint-Projekt.

`mobiletargeting:CreateApp`

Erstellen Sie ein Amazon-Pinpoint-Projekt.

- URI – `/apps`
- Methode – POST

- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:DeleteApp`

Löschen Sie ein Amazon-Pinpoint-Projekt.

- URI – `/apps/projectId`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApp`

Rufen Sie Informationen zu einem Amazon-Pinpoint-Projekt ab.

- URI – `/apps/projectId`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApps`

Rufen Sie Informationen über alle Projekte ab, die Ihrem Amazon-Pinpoint-Konto zugeordnet sindx.

- URI – `/apps`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:GetApplicationSettings`

Rufen Sie die Standardeinstellungen für ein Amazon-Pinpoint-Projekt ab.

- URI – `/apps/projectId/settings`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:UpdateApplicationSettings`

Aktualisieren Sie die Standardeinstellungen für ein Amazon-Pinpoint-Projekt.

- URI – `/apps/projectId/settings`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Empfehlungsmodelle

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Amazon-Pinpoint-Konfigurationen zum Abrufen und Verarbeiten von Empfehlungsdaten aus Empfehlungsmodellen. Ein Empfehlungsmodell ist eine Art Machine-Learning-Modell, das personalisierte Empfehlungen vorhersagt und generiert, indem es Muster in Daten findet.

`mobiletargeting:CreateRecommenderConfiguration`

Erstellen Sie eine Amazon Pinpoint-Konfiguration für ein Empfehlungsmodell.

- URI – `/recommenders`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:recommenders`

`mobiletargeting:GetRecommenderConfigurations`

Rufen Sie Informationen zu allen Konfigurationen des Empfehlungsmodells, die Ihrem Amazon Pinpoint-Konto zugeordnet sind.

- URI – `/recommenders`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:recommenders`

`mobiletargeting:GetRecommenderConfiguration`

Rufen Sie Informationen über eine individuelle Amazon-Pinpoint-Konfiguration für ein Empfehlungsmodell ab.

- URI – `/recommenders/recommenderId`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

`mobiletargeting:UpdateRecommenderConfiguration`

Aktualisieren Sie eine Amazon Pinpoint-Konfiguration für ein Empfehlungsmodell.

- URI – `/recommenders/recommenderId`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

`mobiletargeting:DeleteRecommenderConfiguration`

Löschen Sie eine Amazon Pinpoint-Konfiguration für ein Empfehlungsmodell.

- URI – `/recommenders/recommenderId`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:recommenders/recommenderId`

Segmente

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Segmenten in Ihrem Amazon-Pinpoint-Konto. In Amazon Pinpoint sind Segmente Gruppen von Empfängern für Ihre Kampagnen, die bestimmte, von Ihnen definierte Attribute gemeinsam haben.

`mobiletargeting:CreateSegment`

Erstellen eines Segments Wenn Sie einem Benutzer erlauben möchten, ein Segment durch Importieren von Endpunkt-Daten außerhalb von Amazon Pinpoint zu erstellen, lassen Sie die Aktion `mobiletargeting:CreateImportJob` zu.

- URI – `/apps/projectId/segments`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:DeleteSegment`

Löschen eines Segments

- URI – `/apps/projectId/segments/segmentId`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

`mobiletargeting:GetSegment`

Abrufen von Informationen zu einem bestimmten Segment

- URI – `/apps/projectId/segments/segmentId`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

mobiletargeting:GetSegmentExportJobs

Abrufen von Informationen zu Aufträgen, die Endpunktdefinitionen für ein Segment exportieren

- URI – /apps/*projectId*/segments/*segmentId*/jobs/export
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*/jobs/export

mobiletargeting:GetSegments

Abrufen von Informationen zu allen Segmenten für ein Projekt

- URI – /apps/*projectId*/segments
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetSegmentImportJobs

Rufen Sie Informationen zu Aufträgen ab, die Segmente durch Importieren von Endpunktdefinitionen aus Amazon S3 erstellen.

- URI – /apps/*projectId*/segments/*segmentId*/jobs/import
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersion

Abrufen von Informationen zu einer bestimmten Segmentversion

- URI – /apps/*projectId*/segments/*segmentId*/versions/*versionId*
- Methode – GET
- Ressourcen-ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersions

Abrufen von Informationen zu den aktuellen und vorherigen Versionen eines Segments

- URI – /apps/*projectId*/segments/*segmentId*/versions
- Methode – GET

- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

`mobiletargeting:UpdateSegment`

Aktualisieren eines bestimmten Segments

- URI – `/apps/projectId/segments/segmentId`
- Methode – PUT
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

Tags

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Tags für Amazon-Pinpoint-Ressourcen.

`mobiletargeting:ListTagsForResource`

Abrufen von Informationen zu den Tags, die einem Projekt, einer Kampagne, einer Nachrichtenvorlage oder einem Segment zugeordnet sind

- URI – `/tags/resource-arn`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:TagResource`

Hinzufügen eines oder mehrerer Tags zu einem Projekt, einer Kampagne, einer Nachrichtenvorlage oder einem Segment

- URI – `/tags/resource-arn`
- Methode – POST
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:UntagResource`

Entfernen eines oder mehrerer Tags von einem Projekt, einer Kampagne, einer Nachrichtenvorlage oder einem Segment

- URI – `/tags/resource-arn`

- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:*`

Benutzer

Die folgenden Berechtigungen beziehen sich auf die Verwaltung von Benutzern. In Amazon Pinpoint sind Benutzer die Personen, die Nachrichten von Ihnen erhalten. Ein einzelner Benutzer kann mit mehr als einem Endpunkt verknüpft sein.

`mobiletargeting:DeleteUserEndpoints`

Löschen aller Endpunkte, die einer Benutzer-ID zugeordnet sind

- URI – `/apps/projectId/users/userId`
- Methode – DELETE
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

`mobiletargeting:GetUserEndpoints`

Abrufen von Informationen zu allen Endpunkten, die einer Benutzer-ID zugeordnet sind

- URI – `/apps/projectId/users/userId`
- Methode – GET
- Ressourcen-ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

Amazon-Pinpoint-SMS- und -Sprachnachrichten-API-Aktionen, Version 1

In diesem Abschnitt werden Aktionen für Features identifiziert, die von der Amazon-Pinpoint-SMS- und -Sprachnachrichten-API verfügbar sind. Dies ist eine ergänzende API, die erweiterte Optionen für die Verwendung und Verwaltung der SMS- und Sprachkanäle in Amazon Pinpoint bietet. Weitere Informationen zu dieser API finden Sie in der Amazon-Pinpoint-SMS- und Sprachnachricht-API-Referenz.

`sms-voice:CreateConfigurationSet`

Erstellen Sie einen Konfigurationssatz zum Senden von Sprachnachrichten.

- URI – `/sms-voice/configuration-sets`
- Methode – POST
- Ressourcen-ARN – nicht verfügbar. Verwenden Sie *.

`sms-voice:DeleteConfigurationSet`

Löschen eines Konfigurationssatzes zum Senden von Sprachnachrichten

- URI — /sms-voice/configuration-sets/ *`ConfigurationSetName`*
- Methode – DELETE
- Ressourcen-ARN – nicht verfügbar. Verwenden Sie *.

`sms-voice:GetConfigurationSetEventDestinations`

Abrufen von Informationen zu einem Konfigurationssatz und den darin enthaltenen Ereigniszielen

- URI — /sms-voice/configuration-sets/ *`ConfigurationSetName`*/event-destinations
- Methode – GET
- Ressourcen-ARN – nicht verfügbar. Verwenden Sie *.

`sms-voice:CreateConfigurationSetEventDestination`

Erstellen eines Ereignisziels für Sprachereignisse.

- URI — /sms-voice/configuration-sets/ /event-destinations *`ConfigurationSetName`*
- Methode – POST
- Ressourcen-ARN – nicht verfügbar. Verwenden Sie *.

`sms-voice:UpdateConfigurationSetEventDestination`

Aktualisieren eines Ereignisziels für Sprachereignisse.

- URI — *`ConfigurationSetName`*/sms-voice/configuration-sets/ /event-destinations/ *`EventDestinationName`*
- Methode – PUT
- Ressourcen-ARN – nicht verfügbar. Verwenden Sie *.

`sms-voice:DeleteConfigurationSetEventDestination`

Löschen eines Ereignisziels für Sprachereignisse.

- URI — *`ConfigurationSetName`*/sms-voice/configuration-sets/ /event-destinations/ *`EventDestinationName`*
- Methode – DELETE
- Ressourcen-ARN – nicht verfügbar. Verwenden Sie *.

`sms-voice:SendVoiceMessage`

Erstellen und Senden von Sprachnachrichten.

- URI – /sms-voice/voice/message
- Methode – POST
- Ressourcen-ARN – nicht verfügbar. Verwenden Sie *.

Beispiele für identitätsbasierte Amazon-Pinpoint-Richtlinien

Benutzer und Rollen besitzen standardmäßig keine Berechtigungen zum Erstellen oder Ändern von Amazon-Pinpoint-Ressourcen. Sie können auch keine Aufgaben mit der, oder einer API ausführen. AWS Management Console AWS CLI AWS Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den -Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

Themen

- Beispiele: Gewähren von Zugriff auf Amazon-Pinpoint-API-Aktionen
- Beispiele: Gewähren von Zugriff auf Amazon-Pinpoint-SMS- und Sprachnachricht-API-Aktionen
- Beispiel: Beschränken des Amazon-Pinpoint-Projektzugriffs auf bestimmte IP-Adressen
- Beispiel: Beschränken des Amazon-Pinpoint-Zugriffs auf der Grundlage von Tags
- Beispiel: Amazon Pinpoint erlauben, E-Mails mit Identitäten zu senden, die in Amazon SES verifiziert wurden

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien können festlegen, ob jemand Amazon-Pinpoint-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder daraus löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation

B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Amazon-Pinpoint-Konsole

Um auf die Amazon-Pinpoint-Konsole zugreifen zu können, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon Pinpoint Pinpoint-Ressourcen in Ihrem AWS Konto aufzulisten und einzusehen. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die Berechtigungen anwendet, die strenger als die mindestens erforderlichen Berechtigungen sind, funktioniert die Konsole für Entitäten mit dieser Richtlinie (Benutzer oder Rollen) nicht wie vorgesehen. Um sicherzustellen, dass diese Entitäten die Amazon-Pinpoint-Konsole verwenden können, fügen Sie den Entitäten eine Richtlinie an. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Die folgende Beispielrichtlinie bietet Lesezugriff auf die Amazon Pinpoint Pinpoint-Konsole in einer bestimmten Region. AWS Sie beinhaltet den schreibgeschützten Zugriff auf andere Services, von denen die Amazon-Pinpoint-Konsole abhängt, wie Amazon Simple Email Service (Amazon SES), IAM und 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"
                }
            }
        }
    ]
}
```

Ersetzen Sie im vorherigen Richtlinienbeispiel *Region* durch den Namen einer AWS Region und AccountID durch Ihre *AWS accountId*.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Beispiel: Zugriff auf ein einzelnes Amazon-Pinpoint-Projekt

Sie können auch Richtlinien für den schreibgeschützten Zugriff erstellen, die Zugriff nur auf bestimmte Projekte gewähren. Mit der folgenden Beispielrichtlinie können sich Benutzer bei der

Konsole anmelden und eine Projektliste anzeigen. Außerdem können Benutzer Informationen zu verwandten Ressourcen für andere AWS -Services anzeigen, von denen die Amazon-Pinpoint-Konsole abhängig ist, wie etwa Amazon SES, IAM und Amazon Kinesis. Allerdings können Benutzer nur zusätzliche Informationen zu dem Projekt anzeigen, die in der Richtlinie angegeben sind. Sie können diese Richtlinie ändern, um den Zugriff auf zusätzliche Projekte oder AWS Regionen zu ermöglichen.

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

Ersetzen Sie im vorherigen Beispiel *Region* durch den Namen einer AWS Region, ersetzen Sie *AccountID durch Ihre AWS accountId* und *ProjectID* durch die ID des Amazon Pinpoint Pinpoint-Projekts, für das Sie Zugriff gewähren möchten.

Ebenso können Sie Richtlinien erstellen, die einem Benutzer in Ihrem AWS Konto eingeschränkten Schreibzugriff auf eines Ihrer Amazon Pinpoint Pinpoint-Projekte gewähren, beispielsweise auf das Projekt mit der `810c7aab86d42fb2b56c8c966example` Projekt-ID. In diesem Fall wollen Sie dem Benutzer erlauben, Projektkomponenten wie Segmente und Kampagnen anzuzeigen, hinzuzufügen und zu aktualisieren, aber keine Komponenten zu löschen.

Erstellen Sie nicht nur Berechtigungen für `mobiletargeting:Get`-und `mobiletargeting:List`-Aktionen, sondern auch eine Richtlinie, die Berechtigungen für die folgenden Aktionen erteilt: `mobiletargeting:Create`, `mobiletargeting:Update` und `mobiletargeting:Put`. Dies sind die zusätzlichen Berechtigungen, die zum Erstellen und Verwalten der meisten Projektkomponenten erforderlich sind. Beispielsweise:

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

Beispiel: Anzeigen von Amazon-Pinpoint-Ressourcen basierend auf Tags

Sie können in Ihrer identitätsbasierten Richtlinie Bedingungen für die Steuerung des Zugriffs auf Amazon-Pinpoint-Ressourcen auf der Basis von Tags verwenden. Diese Beispielrichtlinie zeigt, wie Sie diese Art von Richtlinie erstellen können, um die Anzeige von Amazon-Pinpoint-Ressourcen zu ermöglichen. Die Berechtigung wird jedoch nur gewährt, wenn der Wert des Ressourcen-Tags `0wner` der Name des Benutzers ist. Diese Richtlinie gewährt auch die Berechtigungen, die für die Ausführung dieser Aktion auf der Konsole erforderlich sind.

```
{
    "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:*"
                }
            }
        }
    ]
}
```

Sie können diese Richtlinie den -Benutzern in Ihrem Konto anfügen. Wenn ein Benutzer mit dem Namen `richard-roe` versucht, eine Amazon-Pinpoint-Ressource anzuzeigen, muss die Ressource mit dem Tag `0wner=richard-roe` oder `owner=richard-roe` versehen sein. Andernfalls wird der Zugriff abgelehnt. Der Tag-Schlüssel `0wner` der Bedingung stimmt sowohl mit `0wner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

Beispiel: Benutzern die Berechtigung zur Anzeige eigener Berechtigungen erteilen

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie beinhaltet Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI API oder. 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": "*"
        }
    ]
}
```

Beispiele: Gewähren von Zugriff auf Amazon-Pinpoint-API-Aktionen

Dieser Abschnitt enthält Beispielrichtlinien, die den Zugriff auf Features ermöglichen, die über die Amazon-Pinpoint-API verfügbar sind, bei der es sich um die primäre API für Amazon Pinpoint handelt. Weitere Informationen zur Verwendung dieser API finden Sie in der API-Referenz zu Amazon Pinpoint.

Schreibgeschützter Zugriff

Die folgende Beispielrichtlinie ermöglicht den schreibgeschützten Zugriff auf alle Ressourcen in Ihrem Amazon Pinpoint Pinpoint-Konto in einer bestimmten Region. AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewAllResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        }
    ]
}
```

Ersetzen Sie im vorherigen Beispiel *Region* durch den Namen einer AWS Region und AccountID durch Ihre *AWS accountId*.

Administratorzugriff

Die folgende Beispielrichtlinie ermöglicht vollen Zugriff auf alle Amazon-Pinpoint-Aktionen und Ressourcen in Ihrem Amazon-Pinpoint-Konto:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
```

```
            "Action": [
                "mobiletargeting:*"
            ],
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        }
    ]
}
```

Ersetzen Sie im vorhergehenden Beispiel *accountId* durch Ihre AWS -Konto-ID.

Beispiele: Gewähren von Zugriff auf Amazon-Pinpoint-SMS- und Sprachnachricht-API-Aktionen

Dieser Abschnitt enthält Beispielrichtlinien, die den Zugriff auf Features ermöglichen, die über die Amazon-Pinpoint-SMS- und Sprachnachricht-API verfügbar sind. Dies ist eine ergänzende API, die erweiterte Optionen für die Verwendung und Verwaltung der SMS- und Sprachkanäle in Amazon Pinpoint bietet. Weitere Informationen zu dieser API finden Sie in der Amazon-Pinpoint-SMS- und Sprachnachricht-API-Referenz.

Schreibgeschützter Zugriff

Die folgende Beispielrichtlinie ermöglicht den schreibgeschützten Zugriff auf alle Amazon Pinpoint SMS- und Voice-API-Aktionen und -Ressourcen in Ihrem Konto: 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:*"
                }
```

```
            }
        }
    ]
}
```

Administratorzugriff

Die folgende Beispielrichtlinie ermöglicht vollen Zugriff auf alle Amazon Pinpoint SMS- und Voice-API-Aktionen und -Ressourcen in Ihrem AWS Konto:

```
{
    "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:*"
                }
            }
        }
    ]
}
```

Beispiel: Beschränken des Amazon-Pinpoint-Projektzugriffs auf bestimmte IP-Adressen

Die folgende Beispielrichtlinie gewährt jedem Benutzer Berechtigungen zum Ausführen einer Amazon-Pinpoint-Aktion für ein bestimmtes Projekt (*`projectId`*). Die Anforderung muss jedoch aus dem in der Bedingung angegebenen IP-Adressbereich stammen.

Die Bedingung in dieser Anweisung identifiziert den Bereich `54.240.143.*` als zulässigen Bereich für Internetprotokoll 4-Adressen (IPv4), mit einer Ausnahme: `54.240.143.188`. Der `Condition`

Block verwendet die `NotIpAddress` Bedingungen `IpAddress` und und den `aws:SourceIp` Bedingungsschlüssel, der ein AWS-weiter Bedingungsschlüssel ist. Weitere Informationen zu diesen Bedingungsschlüsseln finden Sie unter Angeben von Bedingungen in einer Richtlinie im IAM-Benutzerhandbuch. Die `aws:SourceIp`-IPv4-Werte verwenden die CIDR-Standardnotation. Weitere Informationen finden Sie unter IP-Adressen-Bedingungsoperatoren im IAM-Benutzerhandbuch.

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

Beispiel: Beschränken des Amazon-Pinpoint-Zugriffs auf der Grundlage von Tags

Die folgende Beispielrichtlinie gewährt Berechtigungen zum Ausführen einer Amazon-Pinpoint-Aktion für ein bestimmtes Projekt (*`projectId`*). Berechtigungen werden jedoch nur erteilt, wenn die Anforderung von einem Benutzer stammt, dessen Name ein Wert im `Owner`-Ressourcen-Tag für das Projekt ist, wie in der Bedingung angegeben.

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

Beispiel: Amazon Pinpoint erlauben, E-Mails mit Identitäten zu senden, die in Amazon SES verifiziert wurden

Wenn Sie eine E-Mail-Identität (z. B. eine E-Mail-Adresse oder Domain) über die Amazon-Pinpoint-Konsole verifizieren, wird diese Identität automatisch so konfiguriert, dass sie sowohl von Amazon Pinpoint als auch von Amazon SES verwendet werden kann. Wenn Sie jedoch eine E-Mail-Identität über Amazon SES verifizieren und diese Identität mit Amazon Pinpoint verwenden möchten, müssen Sie eine Richtlinie auf diese Identität anwenden.

Die folgende Beispielrichtlinie erteilt Amazon Pinpoint die Erlaubnis, E-Mails mit einer E-Mail-Identität zu senden, die über Amazon SES verifiziert wurde.

```
{
    "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/*"
                }
            }
        }
    ]
}
```

Wenn Sie Amazon Pinpoint in der Region AWS GovCloud (USA West) verwenden, verwenden Sie stattdessen das folgende Richtlinienbeispiel:

```
{
    "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/*"
                }
            }
        }
    ]
}
```

IAM-Rollen für allgemeine Amazon-Pinpoint-Aufgaben

Eine IAM-Rolle ist eine AWS Identity and Access Management (IAM-) Identität, die Sie in Ihrem AWS Konto erstellen und bestimmte Berechtigungen gewähren können. Eine IAM-Rolle ist eine AWS Identität mit Berechtigungsrichtlinien, die festlegen, wofür die Identität zuständig ist und welche nicht.

AWS Eine Rolle ist jedoch nicht einer einzigen Person zugeordnet, sondern kann von allen Personen angenommen werden, die diese Rolle benötigen.

Einer Rolle sind außerdem keine standardmäßigen, langfristigen Anmeldeinformationen zugeordnet. Stattdessen werden temporäre Anmeldeinformationen für eine Sitzung bereitgestellt. Sie können IAM-Rollen verwenden, um den Zugriff an Benutzer, Apps, Anwendungen oder Dienste zu delegieren, die normalerweise keinen Zugriff auf Ihre Ressourcen haben. AWS

Aus diesen Gründen können Sie mithilfe von IAM-Rollen Amazon Pinpoint in bestimmte AWS -Services und -Ressourcen für Ihr Konto integrieren. Beispielsweise möchten Sie Amazon Pinpoint den Zugriff auf Endpunktdefinitionen gewähren, die Sie in einem Amazon Simple Storage Service (Amazon S3)-Bucket speichern und für Segmente verwenden möchten. Oder Sie möchten Amazon Pinpoint erlauben, Ereignisdaten in einen Amazon-Kinesis-Stream für Ihr Konto zu streamen. In ähnlicher Weise möchten Sie möglicherweise IAM-Rollen verwenden, um Web- oder mobilen Apps die Registrierung von Endpunkten oder die Meldung von Nutzungsdaten für Amazon Pinpoint Pinpoint-Projekte zu ermöglichen, ohne AWS Schlüssel in die Apps einzubetten (wo sie sich nur schwer rotieren lassen und Benutzer sie möglicherweise extrahieren können).

Für diese Szenarien können Sie den Zugriff auf Amazon Pinpoint mithilfe von IAM-Rollen delegieren. In diesem Abschnitt werden Beispiele allgemeiner Amazon-Pinpoint-Aufgaben erläutert, die IAM-Rollen für die Arbeit mit anderen AWS -Services verwenden. Weitere Informationen zur Verwendung von IAM-Rollen mit Web- und mobilen Apps finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.

Themen

IAM-Rolle für das Importieren von Endpunkten oder Segmenten

Mit Amazon Pinpoint können Sie ein Benutzersegment definieren, indem Sie Endpunktdefinitionen aus einem Amazon Simple Storage Service (Amazon S3) -Bucket in Ihr AWS Konto importieren. Vor dem Importieren müssen Sie die erforderlichen Berechtigungen auf Amazon Pinpoint übertragen.

Dazu erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle und fügen der Rolle die folgenden Richtlinien hinzu:

- Die von `AmazonS3ReadOnlyAccess` AWS verwaltete Richtlinie. Diese Richtlinie wird von erstellt und verwaltet AWS und gewährt schreibgeschützten Zugriff auf Ihren Amazon S3 S3-Bucket.
- Eine Vertrauensrichtlinie, die Amazon Pinpoint ermöglicht, die Rolle zu übernehmen.

Nach dem Erstellen der Rolle können Sie mit Amazon Pinpoint Segmente aus einem Amazon-S3-Bucket importieren. Informationen zum Erstellen des -Buckets, Erstellen von Endpunkt-Dateien und Importieren eines Segments mit der Konsole finden Sie unter Importieren von Segmenten im Amazon-Pinpoint-Benutzerhandbuch. Ein Beispiel dafür, wie Sie ein Segment programmgesteuert mithilfe von importieren AWS SDK for Java, finden Importieren von Segmenten Sie in diesem Handbuch.

Erstellen der IAM-Rolle (AWS CLI)

Führen Sie die folgenden Schritte aus, um die IAM-Rolle mithilfe von () zu erstellen. AWS Command Line Interface AWS CLI Falls Sie das nicht installiert haben AWS CLI, finden Sie weitere Informationen unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Um die IAM-Rolle mit dem zu erstellen AWS CLI

1. Erstellen Sie eine JSON-Datei, die die Vertrauensrichtlinie für Ihre Rolle enthält, und speichern Sie die Datei lokal. Sie können die folgende Vertrauensrichtlinie verwenden.

```
{
     "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"
```

```
                    }
                }
            }
        ]
}
```

Gehen Sie im vorhergehenden Beispiel wie folgt vor:

- Ersetzen Sie *Region* durch die AWS Region, in der Sie Amazon Pinpoint verwenden.
- Ersetzen Sie *accountId* durch die eindeutige ID für Ihr AWS Konto.
- Ersetzen Sie die *Anwendungs-ID* durch die eindeutige ID des Projekts.

2. Geben Sie an der Befehlszeile den Befehl `create-role` ein, um die Rolle zu erstellen und die Vertrauensrichtlinie anzufügen:

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document
 file://PinpointImportTrustPolicy.json
```

Geben Sie nach dem Präfix `file://` den Pfad zur JSON-Datei an, die die Vertrauensrichtlinie enthält.

Nachdem Sie diesen Befehl ausgeführt haben, sehen Sie eine Ausgabe in Ihrem Terminal, die der folgenden ähnelt:

```
{
    "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. Verwenden Sie den attach-role-policyBefehl, um die `AmazonS3ReadOnlyAccess` AWS verwaltete Richtlinie an die Rolle anzuhängen:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport
```

IAM-Rolle für das Exportieren von Endpunkten oder Segmenten

Sie können eine Liste der Endpunkte abrufen, indem Sie einen Exportauftrag erstellen. Wenn Sie einen Exportauftrag erstellen, müssen Sie eine Projekt-ID angeben. Optional können Sie auch eine Segment-ID festlegen. Amazon Pinpoint exportiert dann eine Liste der mit dem Projekt oder Segment verknüpften Endpunkte in einen Amazon Simple Storage Service (Amazon S3)-Bucket. Die resultierende Datei enthält eine JSON-formatierte Liste von Endpunkten und deren Attribute (wie z. B. Channel, Adresse, An-/Abmeldungsstatus, Erstellungsdatum und Endpunkt-ID).

Zum Erstellen eines Exportauftrags müssen Sie eine IAM-Rolle konfigurieren, mit der Amazon Pinpoint Daten in einen Amazon-S3-Bucket schreiben darf. Die Konfiguration der Rolle besteht aus zwei Schritten:

1. Erstellen Sie eine IAM-Richtlinie, mit der eine Entity (in diesem Fall Amazon Pinpoint) Daten in einen bestimmten Amazon-S3-Bucket schreiben darf.
2. Erstellen Sie eine IAM-Rolle und fügen Sie sie der Richtlinie an.

In diesem Abschnitt werden Verfahren für die Durchführung dieser beiden Schritte beschrieben. In diesem Verfahren wird davon ausgegangen, dass Sie bereits einen Amazon-S3-Bucket sowie einen Ordner innerhalb dieses Buckets zum Speichern von exportierten Segmenten erstellt haben. Weitere Informationen zum Erstellen eines Buckets finden Sie unter Erstellen von Buckets im Benutzerhandbuch für Amazon Simple Storage Service.

Bei diesen Verfahren wird vorausgesetzt, dass Sie die AWS Command Line Interface (AWS CLI) bereits installiert und konfiguriert haben. Informationen zur Einrichtung von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Schritt 1: Erstellen der IAM-Richtlinie

Eine IAM-Richtlinie definiert die Berechtigungen für eine Entity, wie z. B. eine Identität oder Ressource. Zum Erstellen einer Rolle für das Exportieren von Amazon-Pinpoint-Endpunkten müssen Sie eine Richtlinie erstellen, die die Berechtigung zum Schreiben von Daten in einen bestimmten Ordner eines spezifischen Amazon-S3-Buckets erteilt. Das folgende Richtlinienbeispiel befolgt die Sicherheitsmaßnahme, die für das Erteilen von geringsten Rechten gilt, d. h., es werden nur die Berechtigungen erteilt, die zum Durchführen einer einzelnen Aufgabe erforderlich sind.

So erstellen Sie die -IAM-Richtlinie

1. Erstellen Sie in einem Texteditor eine neue Datei. Fügen Sie folgenden Code in die Datei ein:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::*" ]
        },
        {
            "Sid": "AllowRootAndHomeListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::example-bucket" ],
```

```
            "Condition": {
                "StringEquals": {
                    "s3:delimiter": [ "/" ],
                    "s3:prefix": [
                        "",
                        "Exports/"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::example-bucket" ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "Exports/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
            "Action": [ "s3:*" ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::example-bucket/Exports/*" ]
        }
    ]
}
```

Ersetzen Sie im vorangegangenen Code alle Instances von *example-bucket* durch den Namen des Amazon-S3-Buckets, in dem sich der Ordner befindet, in den Sie die Segmentinformationen exportieren möchten. Ersetzen Sie außerdem alle Instances von *Exports* durch den Namen des Ordners.

Wenn Sie fertig sind, speichern Sie die Datei unter `s3policy.json`.

2. Navigieren Sie mithilfe von zu dem Verzeichnis AWS CLI, in dem sich die `s3policy.json` Datei befindet. Geben Sie dann den folgenden Befehl ein, um die Richtlinie zu erstellen:

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
 file://s3policy.json
```

Wenn die Richtlinie erfolgreich ausgeführt wurde, sehen Sie eine Ausgabe ähnlich der folgenden:

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

Kopieren Sie den Amazon-Ressourcennamen (ARN) der Richtlinie (`arn:aws:iam::123456789012:policy/s3ExportPolicy` im vorherigen Beispiel). Im nächsten Abschnitt müssen Sie diesen ARN angeben, wenn Sie die Rolle erstellen.

> **Note**
>
> Wenn eine Meldung angezeigt wird, die besagt, dass Ihr Konto für die `CreatePolicy`-Operation nicht autorisiert ist, müssen Sie Ihrem Benutzerkonto eine Richtlinie anfügen, mit der Sie neue IAM-Richtlinien und -Rollen erstellen können. Informationen finden Sie im Abschnitt Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im -IAM-Benutzerhandbuch.

Schritt 2: Erstellen der IAM-Rolle

Nachdem Sie eine IAM-Richtlinie erstellt haben, können Sie eine Rolle erstellen und dieser die Richtlinie anfügen. Jede IAM-Rolle enthält eine Vertrauensrichtlinie. Hierbei handelt es sich um eine Reihe von Regeln, die angibt, welche Entitys die Rolle übernehmen dürfen. In diesem Abschnitt erstellen Sie eine Vertrauensrichtlinie, die Amazon Pinpoint ermöglicht, die Rolle zu übernehmen.

Als Nächstes erstellen Sie die eigentliche Rolle und fügen die Richtlinie an, die Sie im vorherigen Abschnitt erstellt haben.

So erstellen Sie die IAM-Rolle

1. Erstellen Sie in einem Texteditor eine neue Datei. Fügen Sie folgenden Code in die Datei ein:

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

   Speichern Sie die Datei als `trustpolicy.json`.

2. Navigieren Sie mithilfe von zu dem Verzeichnis AWS CLI, in dem sich die `trustpolicy.json` Datei befindet. Geben Sie den folgenden Befehl ein, um eine neue Rolle zu erstellen.

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
 file://trustpolicy.json
```

3. Geben Sie in der Befehlszeile den folgenden Befehl ein, um die Richtlinie, die Sie im vorherigen Abschnitt erstellt haben, der Rolle anzufügen, die Sie gerade erstellt haben:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
s3ExportPolicy --role-name s3ExportRole
```

Ersetzen Sie im vorherigen Befehl *arn:aws:iam: :123456789012:policy/s3 ExportPolicy* durch den ARN der Richtlinie, die Sie im vorherigen Abschnitt erstellt haben.

IAM-Rolle zum Abrufen von Empfehlungen von Amazon Personalize

Sie können Amazon Pinpoint so konfigurieren, dass Empfehlungsdaten aus einer Amazon-Personalize-Lösung abgerufen werden, die als Amazon-Personalize-Kampagne bereitgestellt wurde. Mithilfe dieser Daten können Sie personalisierte Empfehlungen an Nachrichtenempfänger basierend auf den Attributen und Verhaltensweisen der einzelnen Empfänger senden. Weitere Informationen finden Sie unter Machine-Learning-Modelle im Amazon-Pinpoint-Benutzerhandbuch.

Bevor Sie Empfehlungsdaten aus einer Amazon-Personalize-Kampagne abrufen können, müssen Sie eine AWS Identity and Access Management (IAM)-Rolle erstellen, mit der Amazon Pinpoint die Daten aus der Kampagne abrufen kann. Amazon Pinpoint kann diese Rolle automatisch für Sie erstellen, wenn Sie die Konsole verwenden, um ein Empfehlungsmodell in Amazon Pinpoint einzurichten. Sie können diese Rolle auch manuell erstellen.

Um die Rolle manuell zu erstellen, führen Sie mithilfe der IAM-API folgende Schritte aus:

1. Erstellen Sie eine IAM-Richtlinie, die es einer Entity (in diesem Fall Amazon Pinpoint) gestattet, Empfehlungsdaten aus einer Amazon-Personalize-Kampagne abzurufen.
2. Erstellen Sie eine IAM-Rolle und fügen Sie ihr die IAM-Richtlinie an.

In diesem Thema wird erklärt, wie Sie diese Schritte mithilfe von () ausführen. AWS Command Line Interface AWS CLI Es wird davon ausgegangen, dass Sie die Amazon-Personalize-Lösung bereits erstellt und als Amazon-Personalize-Kampagne bereitgestellt haben. Informationen zum Erstellen und Bereitstellen einer Kampagne finden Sie unter Erstellen einer Kampagne im Amazon-Personalize-Entwicklerhandbuch.

In diesem Abschnitt wird auch vorausgesetzt, dass Sie die AWS CLI bereits installiert und konfiguriert haben. Informationen zur Einrichtung von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Schritt 1: Erstellen der IAM-Richtlinie

Eine IAM-Richtlinie definiert Berechtigungen für eine Entity, wie z. B. eine Identität oder Ressource. Um eine Rolle zu erstellen, mit der Amazon Pinpoint Empfehlungsdaten aus einer Amazon-

Personalize-Kampagne abrufen kann, müssen Sie zunächst eine IAM-Richtlinie für die Rolle erstellen. Diese Richtlinie muss Amazon Pinpoint Folgendes ermöglichen:

- Abrufen von Konfigurationsinformationen für die Lösung, die von der Kampagne bereitgestellt wird (`DescribeSolution`)
- Überprüfen des Status der Kampagne (`DescribeCampaign`)
- Abrufen von Empfehlungsdaten aus der Kampagne (`GetRecommendations`)

Im folgenden Verfahren ermöglicht die Beispielrichtlinie diesen Zugriff für eine bestimmte Amazon-Personalize-Lösung, die von einer bestimmten Amazon-Personalize-Kampagne bereitgestellt wurde.

So erstellen Sie die -IAM-Richtlinie

1. Erstellen Sie in einem Texteditor eine neue Datei. Fügen Sie folgenden Code in die Datei ein:

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

   Ersetzen Sie im vorangegangenen Beispiel den *kursiv formatierten* Text durch Ihre Informationen:

   - *Region*: Der Name der AWS -Region, in der die Amazon-Personalize-Lösung und -Kampagne gehostet werden.
   - *accountId*: Ihre AWS-Konto -ID.

- *solutionId*: Die eindeutige Ressourcen-ID für die Amazon-Personalize-Lösung, die im Rahmen der Kampagne bereitgestellt wird.
- *campaignId*: Die eindeutige Ressourcen-ID für die Amazon-Personalize-Kampagne, von der Empfehlungsdaten abgerufen werden sollen.

2. Wenn Sie fertig sind, speichern Sie die Datei unter `RetrieveRecommendationsPolicy.json`.
3. Navigieren Sie über die Befehlszeilenschnittstelle zu dem Verzeichnis, in dem Sie die Datei `RetrieveRecommendationsPolicy.json` gespeichert haben.
4. Geben Sie den folgenden Befehl ein, um eine Richtlinie zu erstellen und sie `RetrieveRecommendationsPolicy` zu nennen. Wenn Sie einen anderen Namen verwenden möchten, wechseln Sie *RetrieveRecommendationsPolicy*zu dem gewünschten Namen.

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document
 file://RetrieveRecommendationsPolicy.json
```

> **Note**
>
> Wenn Sie eine Nachricht erhalten, dass Ihr Konto nicht für die Operation `CreatePolicy` berechtigt ist, müssen Sie Ihrem Benutzerkonto eine Richtlinie anfügen, mit der Sie neue IAM-Richtlinien und -Rollen für Ihr Konto erstellen können. Informationen finden Sie im Abschnitt Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im -IAM-Benutzerhandbuch.

5. Kopieren Sie den Amazon-Ressourcennamen (ARN) der Richtlinie (`arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy` im vorherigen Beispiel). Im nächsten Abschnitt benötigen Sie diesen ARN, um die IAM-Rolle zu erstellen.

Schritt 2: Erstellen der IAM-Rolle

Nachdem Sie die IAM-Richtlinie erstellt haben, können Sie eine IAM-Rolle erstellen und dieser die Richtlinie anfügen.

Jede IAM-Rolle enthält eine Vertrauensrichtlinie. Hierbei handelt es sich um eine Reihe von Regeln, die angeben, welche Entitys die Rolle übernehmen dürfen. In diesem Abschnitt erstellen Sie eine

Vertrauensrichtlinie, die Amazon Pinpoint ermöglicht, die Rolle zu übernehmen. Als Nächstes erstellen Sie die Rolle selbst. Dann fügen Sie die Richtlinie der Rolle an.

So erstellen Sie die IAM-Rolle

1. Erstellen Sie in einem Texteditor eine neue Datei. Fügen Sie folgenden Code in die Datei ein:

```
{
    "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. Speichern Sie die Datei als `RecommendationsTrustPolicy.json`.
3. Navigieren Sie über die Befehlszeilenschnittstelle zu dem Verzeichnis, in dem Sie die Datei `RecommendationsTrustPolicy.json` gespeichert haben.
4. Geben Sie den folgenden Befehl ein, um eine neue Rolle zu erstellen und `PinpointRoleforPersonalize` zu nennen. Wenn Sie einen anderen Namen verwenden möchten, wechseln Sie *`PinpointRoleforPersonalize`*zu dem gewünschten Namen.

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-
document file://RecommendationsTrustPolicy.json
```

5. Geben Sie den folgenden Befehl ein, um die Richtlinie, die Sie im vorherigen Abschnitt erstellt haben, der Rolle anzufügen, die Sie soeben erstellt haben:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

Ersetzen Sie im vorherigen Befehl *arn:aws:iam: :123456789012:policy/* durch den ARN der Richtlinie, die Sie im vorherigen Abschnitt RetrieveRecommendationsPolicy erstellt haben. *PinpointRoleforPersonalize*Ersetzen Sie es auch durch den Namen der Rolle, die Sie in Schritt 4 angegeben haben, wenn Sie einen anderen Namen für die Rolle angegeben haben.

IAM-Rolle für das Streamen von Ereignissen an Kinesis

Amazon Pinpoint kann automatisch App-Nutzungsdaten oder Ereignisdaten von Ihrer App an einen Amazon Kinesis Kinesis-Datenstream oder Amazon Data Firehose-Lieferstream in Ihrem Konto senden. AWS Bevor Amazon Pinpoint mit dem Streamen der Ereignisdaten beginnen kann, müssen Sie die erforderlichen Berechtigungen auf Amazon Pinpoint übertragen.

Wenn Sie die Konsole für die Konfiguration von Ereignis-Streaming verwenden, erstellt Amazon Pinpoint automatisch eine AWS Identity and Access Management (IAM)-Rolle mit den erforderlichen Berechtigungen. Weitere Informationen finden Sie unter Streamen von Amazon Pinpoint-Ereignissen an Amazon Kinesis im Amazon-Pinpoint-Benutzerhandbuch.

Wenn Sie die Rolle manuell erstellen möchten, fügen Sie ihr die folgenden Richtlinien an:

- Eine Berechtigungsrichtlinie, mit der Amazon Pinpoint Ereignisdaten an Ihren Stream senden kann.
- Eine Vertrauensrichtlinie, die Amazon Pinpoint ermöglicht, die Rolle zu übernehmen.

Nach dem Erstellen der Rolle können Sie Amazon Pinpoint so konfigurieren, dass es automatisch Ereignisse an Ihren Stream sendet. Weitere Informationen finden Sie unter Streamen von Amazon-Pinpoint-Ereignissen zu Kinesis in diesem Handbuch.

Erstellen der IAM-Rolle (AWS CLI)

Gehen Sie wie folgt vor, um eine IAM-Rolle mit der AWS Command Line Interface (AWS CLI) manuell zu erstellen. Informationen zum Erstellen der Rolle mithilfe der Amazon-Pinpoint-Konsole finden Sie unter Streamen von Amazon-Pinpoint-Ereignissen zu Kinesis im Amazon-Pinpoint-Benutzerhandbuch.

Falls Sie das noch nicht installiert haben AWS CLI, finden Sie weitere Informationen unter Installation von AWS CLI imAWS Command Line Interface Benutzerhandbuch. Sie müssen außerdem entweder

einen Kinesis-Stream oder einen Firehose-Stream erstellt haben. Informationen zum Erstellen dieser Ressourcen finden Sie unter Creating and Managing Streams im Amazon Kinesis Data Streams Developer Guide oder Creating an Amazon Data Firehose Delivery Stream im Amazon Data Firehose Developer Guide.

Um die IAM-Rolle zu erstellen, verwenden Sie AWS CLI

1. Erstellen Sie eine neue Datei. Fügen Sie die folgende Richtlinie in das Dokument ein und nehmen Sie die folgenden Änderungen vor:

   - Ersetzen Sie *Region* durch die AWS Region, in der Sie Amazon Pinpoint verwenden.
   - Ersetzen Sie *accountId* durch die eindeutige ID für Ihr AWS Konto.
   - Ersetzen Sie *applicationId* durch die eindeutige ID des Projekts.

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

   Wenn Sie fertig sind, speichern Sie die Datei unter `PinpointEventStreamTrustPolicy.json`.

2. Verwenden Sie den Befehl `create-role`, um die Rolle zu erstellen und die Vertrauensrichtlinie anzufügen:

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-
document file://PinpointEventStreamTrustPolicy.json
```

3. Erstellen Sie eine neue Datei, die die Berechtigungsrichtlinie für Ihre Rolle enthält.

   Wenn Sie Amazon Pinpoint so konfigurieren, dass Daten an einen Kinesis-Stream gesendet werden, fügen Sie die folgende Richtlinie in die Datei ein und ersetzen Sie Folgendes:

   - Ersetzen Sie *Region* durch die AWS Region, in der Sie Amazon Pinpoint verwenden.
   - Ersetzen Sie *accountId* durch die eindeutige ID für Ihr AWS Konto.
   - Ersetzen Sie *streamName* durch den Namen Ihres Kinesis-Streams.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Action": [
            "kinesis:PutRecords",
            "kinesis:DescribeStream"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:kinesis:region:accountId:stream/streamName"
        ]
    }
}
```

   Wenn Sie Amazon Pinpoint so konfigurieren, dass Daten an einen Firehose-Stream gesendet werden, fügen Sie alternativ die folgende Richtlinie in die Datei ein und ersetzen Sie Folgendes:

   - Ersetzen Sie *Region* durch die AWS Region, in der Sie Amazon Pinpoint verwenden.
   - Ersetzen Sie *accountId* durch die eindeutige ID für Ihr AWS Konto.
   - Ersetze es *delivery-stream-name*durch den Namen deines Firehose-Streams.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": {
        "Effect": "Allow",
        "Action": [
         "firehose:PutRecordBatch",
         "firehose:DescribeDeliveryStream"
        ],
        "Resource": [
         "arn:aws:firehose:region:accountId:deliverystream/delivery-stream-name"
     ]
    }
}
```

Wenn Sie fertig sind, speichern Sie die Datei unter `PinpointEventStreamPermissionsPolicy.json`.

4. Verwenden Sie den Befehl `put-role-policy`, um die Berechtigungsrichtlinie der Rolle anzufügen:

```
aws iam put-role-policy --role-name PinpointEventStreamRole --policy-
name PinpointEventStreamPermissionsPolicy --policy-document file://
PinpointEventStreamPermissionsPolicy.json
```

IAM-Rolle für das Senden von E-Mails mit Amazon SES

Amazon Pinpoint verwendet Ihre Amazon SES SES-Ressourcen, um E-Mails für Ihre Kampagne oder Journey zu versenden. Bevor Amazon Pinpoint Ihre Amazon SES SES-Ressourcen zum Senden von E-Mails verwenden kann, müssen Sie Amazon Pinpoint die erforderlichen Berechtigungen erteilen. Ihr Konto muss über die `iam:UpdateAssumeRolePolicy` Berechtigungen `iam:PutRolePolicy` und verfügen, um IAM-Rollen zu aktualisieren oder zu erstellen.

Die Amazon Pinpoint Pinpoint-Konsole kann automatisch eine AWS Identity and Access Management (IAM-) Rolle mit den erforderlichen Berechtigungen erstellen. Weitere Informationen finden Sie unter Erstellen einer E-Mail-Orchestration-Senderolle im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

Wenn Sie die Rolle manuell erstellen möchten, fügen Sie ihr die folgenden Richtlinien an:

- Eine Berechtigungsrichtlinie, die Amazon Pinpoint Zugriff auf Ihre Amazon SES SES-Ressourcen gewährt.
- Eine Vertrauensrichtlinie, die Amazon Pinpoint ermöglicht, die Rolle zu übernehmen.

Nachdem Sie die Rolle erstellt haben, können Sie Amazon Pinpoint so konfigurieren, dass es Ihre Amazon SES SES-Ressourcen verwendet.

Sie können IAM-Richtlinien mit dem IAM-Richtliniensimulator testen. Weitere Informationen finden Sie unter Testen von IAM-Richtlinien mit dem IAM-Richtliniensimulator im IAM-Benutzerhandbuch.

Erstellen der IAM-Rolle (AWS Management Console)

Gehen Sie wie folgt vor, um manuell eine IAM-Rolle für Ihre Kampagne oder Journey zum Versenden von E-Mails zu erstellen.

1. Erstellen Sie eine neue Berechtigungsrichtlinie, indem Sie den Anweisungen unter Erstellen von Richtlinien mit dem JSON-Editor im IAM-Benutzerhandbuch folgen.

    - Verwenden Sie in Schritt 5 die folgende Berechtigungsrichtlinie für die IAM-Rolle.

        - Ersetzen Sie die *Partition* durch die Partition, in der sich die Ressource befindet. Standardmäßig ist AWS-Regionen die Partition`aws`. Wenn Sie Ressourcen in anderen Partitionen haben, lautet die Partition `aws-partitionname`. Zum Beispiel ist die Partition für Ressourcen in den AWS GovCloud (US-Westen). `aws-us-gov`
        - Ersetzen Sie *Region* durch den Namen der Region AWS-Region , die das Amazon Pinpoint Pinpoint-Projekt hostet.
        - Ersetzen Sie *accountId* durch die eindeutige ID für Ihre AWS-Konto.

```
{
    "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. Erstellen Sie eine neue Vertrauensrichtlinie, indem Sie den Anweisungen unter Erstellen einer Rolle mithilfe benutzerdefinierter Vertrauensrichtlinien im IAM-Benutzerhandbuch folgen.

   a. Verwenden Sie in Schritt 4 die folgende Vertrauensrichtlinie.

      - Ersetzen Sie *accountId* durch die eindeutige ID für Ihre AWS-Konto.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPinpoint",
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

   b. Fügen Sie in Schritt 11 die Berechtigungsrichtlinie hinzu, die Sie im vorherigen Schritt erstellt haben.

Fehlerbehebung der Identitäts- und Zugriffsverwaltung für Amazon Pinpoint

Verwenden Sie die folgenden Informationen, um häufige Probleme, die bei der Arbeit mit Amazon Pinpoint und IAM auftreten können, zu diagnostizieren und zu beheben.

Themen

- Ich bin nicht autorisiert, eine Aktion in Amazon Pinpoint auszuführen
- Ich bin nicht berechtigt, iam auszuführen: PassRole

- Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine Amazon Pinpoint Pinpoint-Ressourcen ermöglichen

Ich bin nicht autorisiert, eine Aktion in Amazon Pinpoint auszuführen

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion auszuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` -Benutzer versucht, die Konsole zum Anzeigen von Details zu einem Projekt zu verwenden, jedoch nicht über `mobiletargeting:`*`GetApp`*-Berechtigungen verfügt:

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mobiletargeting:GetApp on resource: my-example-project
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *`my-example-project`* auf die Ressource `mobiletargeting:`*`GetApp`* zugreifen zu können.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Amazon Pinpoint übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon Pinpoint auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine Amazon Pinpoint Pinpoint-Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Amazon Pinpoint diese Features unterstützt, finden Sie unter Funktionsweise von Amazon Pinpoint mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Protokollierung und Überwachung in Amazon Pinpoint

Protokollierung und Überwachung sind wichtige Teile der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer Amazon-Pinpoint-Projekte und anderer Arten von Amazon-Pinpoint-Ressourcen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer Amazon Pinpoint Pinpoint-

Projekte und -Ressourcen protokollieren und sammeln, um einen etwaigen Mehrpunktfehler leichter debuggen zu können. AWS bietet mehrere Tools, mit denen Sie diese Daten protokollieren und sammeln und auf potenzielle Vorfälle reagieren können:

AWS CloudTrail

Amazon Pinpoint ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die in Amazon Pinpoint von einem Benutzer, einer Rolle oder einem anderen AWS Service ausgeführt wurden. Dazu gehören Aktionen von der Amazon-Pinpoint-Konsole und programmgesteuerte Aufrufe von Amazon-Pinpoint-API-Operationen. Anhand der von gesammelten Informationen können Sie feststellen CloudTrail, welche Anfragen an Amazon Pinpoint gestellt wurden. Für jede Anforderung können Sie angeben, wann sie gestellt wurde, die IP-Adresse, von der sie gestellt wurde, sowie weitere Details. Weitere Informationen finden Sie unter Protokollieren von Amazon Pinpoint API-Aufrufen mit AWS CloudTrail in diesem Handbuch.

Amazon CloudWatch

Sie können Amazon verwenden, CloudWatch um mehrere wichtige Kennzahlen zu Ihrem Amazon Pinpoint Pinpoint-Konto und Ihren Projekten zu sammeln, anzuzeigen und zu analysieren. Sie können CloudWatch damit auch Alarme erstellen, die Sie benachrichtigen, wenn der Wert für eine Metrik bestimmte Bedingungen erfüllt und innerhalb oder über einem von Ihnen definierten Schwellenwert liegt. Wenn Sie einen Alarm erstellen, wird eine Benachrichtigung CloudWatch an ein von Ihnen festgelegtes Amazon Simple Notification Service (Amazon SNS) -Thema gesendet. Weitere Informationen finden Sie unter Überwachung von Amazon Pinpoint mit Amazon CloudWatch im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

AWS Health Dashboards

Mithilfe von AWS Health Dashboards können Sie den Status Ihrer Amazon Pinpoint Pinpoint-Umgebung überprüfen und überwachen. Um den Status des Amazon Pinpoint Pinpoint-Dienstes insgesamt zu überprüfen, verwenden Sie das AWS Service Health Dashboard. Verwenden Sie das AWS Personal Health Dashboard, um historische Daten zu Ereignissen oder Problemen zu überprüfen, zu überwachen und anzuzeigen, die sich speziell auf Ihre AWS Umgebung auswirken könnten. Weitere Informationen zu diesen Dashboards finden Sie im AWS Health - Benutzerhandbuch.

AWS Trusted Advisor

AWS Trusted Advisor untersucht Ihre AWS Umgebung und gibt Empfehlungen für Möglichkeiten, Sicherheitslücken zu schließen, die Systemverfügbarkeit und -leistung zu verbessern und Geld zu sparen. Alle AWS Kunden haben Zugriff auf eine Reihe zentraler Trusted Advisor Prüfungen.

Kunden mit einem Business- oder Enterprise-Supportplan haben Zugriff auf zusätzliche Trusted Advisor Schecks.

Viele dieser Prüfungen können Ihnen dabei helfen, den Sicherheitsstatus Ihrer Amazon Pinpoint Pinpoint-Ressourcen als Teil Ihres AWS Kontos insgesamt zu beurteilen. Die Trusted Advisor - Kernprüfungsgruppe beinhaltet beispielsweise Folgendes:

- Protokollierung der Konfigurationen für Ihr AWS Konto für jede unterstützte AWS Region.
- Zugriffsberechtigungen für Ihre Amazon Simple Storage Service (Amazon S3)-Buckets, die Dateien enthalten können, die Sie in Amazon Pinpoint importieren, um Segmente zu erstellen.
- Verwendung von AWS Identity and Access Management Benutzern, Gruppen und Rollen zur Steuerung des Zugriffs auf Amazon Pinpoint Pinpoint-Ressourcen.
- IAM-Konfigurationen und Richtlinieneinstellungen, die die Sicherheit Ihrer AWS Umgebung und der Amazon Pinpoint Pinpoint-Ressourcen gefährden könnten.

Weitere Informationen finden Sie unter AWS Trusted Advisor im AWS Support - Benutzerhandbuch.

Compliance-Validierung für Amazon Pinpoint

Externe Prüfer bewerten im Rahmen verschiedener AWS -Compliance-Programme die Sicherheit und Compliance von Amazon Pinpoint. Dazu gehören AWS System and Organization Controls (SOC), FedRAMP, HIPAA, ISO/IEC 27001:2013 für Sicherheitsmanagementkontrollen, ISO/IEC 27017:2015 für Cloud-spezifische Kontrollen, ISO/IEC 27018:2014 für den Schutz personenbezogener Daten, ISO/IEC 9001:2015 für Qualitätsmanagementsysteme und andere.

https://aws.amazon.com/compliance/services-in-scope/ Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern herunterladen, indem Sie AWS Artifact. Weitere Informationen finden Sie unter Herunterladen von Berichten in AWS Artifact .

Ihre Compliance-Verantwortung bei der Nutzung von Amazon Pinpoint hängt von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS bietet die folgenden Ressourcen zur Unterstützung bei der Einhaltung von Vorschriften:

- Schnellstartanleitungen zu Sicherheit und Compliance Schnellstartanleitungen zu — In diesen Bereitstellungshandbüchern werden architektonische Überlegungen erörtert und Schritte für die

Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen beschrieben. AWS

- Whitepaper „Architecting for HIPAA“ zu Sicherheit und Compliance — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen entwickeln können. AWS
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- Bewertung von Ressourcen anhand von Regeln im AWS Config Developer Guide — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus, sodass Sie überprüfen können AWS , ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

Amazon Pinpoint ist ein AWS HIPAA-fähiger Service, wenn Kunden die richtigen Kommunikationskanäle verwenden. Wenn Sie mit Amazon Pinpoint Workloads mit geschützten Gesundheitsinformationen (PHI) gemäß HIPAA und den damit verbundenen Rechtsvorschriften und Vorschriften ausführen möchten, sollten Sie den E-Mail-Kanal, den Push-Benachrichtigungskanal oder den SMS-Kanal verwenden, um Nachrichten zu senden, die PHI enthalten. Wenn Sie den SMS-Kanal zum Senden von Nachrichten verwenden, die PHI enthalten, sollten Sie diese Nachrichten über einen speziellen Kurzcode versenden, den Sie für Ihr AWS Konto angefordert haben, um Nachrichten zu senden, die PHI enthalten oder enthalten können. Der Sprachkanal ist nicht AWS HIPAA-fähig. Verwenden Sie den Sprachkanal nicht, um Nachrichten zu senden, die PHI enthalten.

Ausfallsicherheit bei Amazon Pinpoint

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Referenzarchitekturen finden Sie im Handbuch zur Amazon-Pinpoint-resistenten Architektur.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Infrastruktursicherheit in Amazon Pinpoint

Als verwalteter Service ist Amazon Pinpoint durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Pinpoint zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Obwohl Sie diese API-Aufrufe von jedem Netzwerkstandort aus tätigen können, unterstützt Amazon Pinpoint ressourcenbasierte Zugriffsrichtlinien. Diese Richtlinien können Einschränkungen basierend auf der jeweiligen Quell-IP-Adresse enthalten. Weitere Informationen zu dieser Art von Richtlinien finden Sie unter Verwalten des Zugriffs mit Richtlinien.

Darüber hinaus können Sie verschiedene AWS Sicherheitsfunktionen konfigurieren und verwenden, um den Zugriff auf Amazon Pinpoint-Ressourcen von allen mobilen oder Web-Apps aus zu kontrollieren, die Sie in Amazon Pinpoint integrieren. Dies beinhaltet Einschränkungen für API-Aufrufe für Aufgaben wie das Hinzufügen von Endpunkten, das Aktualisieren von Endpunktdaten, das Senden von Ereignisreichen und das Melden von Nutzungsdaten.

Um diese Funktionen nutzen zu können, empfehlen wir Ihnen, die AWS Mobile SDKs oder AWS Amplify JavaScript Bibliotheken zu verwenden, um mobile Apps und Web-Apps in Amazon Pinpoint zu integrieren. Für Android- oder iOS-Apps empfehlen wir, die AWS Mobile SDK for Android bzw. die

AWS Mobile SDK for iOS zu verwenden. Für JavaScript basierte Mobil- oder Web-Apps empfehlen wir, die AWS Amplify JavaScript Library for the Web oder die AWS Amplify JavaScript Library for React Native zu verwenden. Weitere Informationen zu diesen Ressourcen finden Sie unter Erste Schritte mit den AWS mobilen SDKs, Erste Schritte mit der AWS Amplify-Bibliothek für das Web und Erste Schritte mit der AWS Amplify-Bibliothek für React Native.

Konfigurations- und Schwachstellenanalyse in Amazon Pinpoint

Als verwalteter Service ist Amazon Pinpoint durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper Amazon Web Services: Sicherheitsprozesse im Überblick beschrieben sind. Das bedeutet, dass er grundlegende Sicherheitsaufgaben und -verfahren AWS verwaltet und ausführt, um die zugrunde liegende Infrastruktur für Ihr Amazon Pinpoint Pinpoint-Konto und Ihre Ressourcen zu sichern, zu patchen, zu aktualisieren und anderweitig zu warten. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert.

Weitere Informationen finden Sie in den folgenden Ressourcen:

- Compliance-Validierung für Amazon Pinpoint
- Modell der geteilten Verantwortung
- Amazon Web Services: Übersicht über Sicherheitsverfahren (Whitepaper)

Bewährte Methoden für die Sicherheit in Amazon Pinpoint

Verwenden Sie AWS Identity and Access Management (IAM) -Konten, um den Zugriff auf Amazon Pinpoint API-Operationen zu steuern, insbesondere auf Operationen, die Amazon Pinpoint Ressourcen erstellen, ändern oder löschen. Für die Amazon Pinpoint API umfassen diese Ressourcen Projekte, Kampagnen und Reisen. Für die Amazon Pinpoint -SMS- und Sprachnachricht-API umfassen diese Ressourcen Telefonnummern, Pools und Konfigurationssätze.

- Erstellen Sie einen individuellen Benutzer für jede Person, die Amazon Pinpoint Ressourcen verwaltet, auch für Sie. Verwenden Sie keine AWS Root-Anmeldeinformationen, um Amazon Pinpoint Pinpoint-Ressourcen zu verwalten.
- Gewähren Sie jedem Benutzer nur den Mindestsatz an Berechtigungen, die für die Ausführung seiner Aufgaben erforderlich sind.
- Verwenden Sie IAM-Gruppen, um Berechtigungen für mehrere Benutzer effektiv zu verwalten.
- Wechseln Sie regelmäßig die IAM-Anmeldeinformationen.

Weitere Informationen zur Amazon Pinpoint Sicherheit finden Sie unter Sicherheit in Amazon Pinpoint. Weitere Informationen zu IAM finden Sie unter AWS Identity and Access Management. Informationen zu den bewährten Methoden für IAM finden Sie unter Bewährte Methoden für IAM.

Amazon-Pinpoint-Kontingente

In den folgenden Abschnitten werden die Kontingente (früher als Limits bezeichnet), die für Amazon-Pinpoint-Ressourcen und -Vorgänge gelten, aufgeführt und beschrieben. Einige Kontingente können erhöht werden, andere dagegen nicht. Informationen dazu, ob Sie eine Erhöhung für ein Kontingent beantragen können, finden Sie in den einzelnen Abschnitten in der Spalte oder der Anweisung Eligible for Increase (Erhöhungsberechtigt).

Themen

Projekt-Kontingente

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Projekten in Amazon Pinpoint aufgeführt.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Projekte	In jedem AWS-Region können Sie bis zu 100 Projekte haben.	Nein

API-Anforderungskontingente

Amazon Pinpoint implementiert Kontingente, die die Größe und Anzahl der Anfragen einschränken, die Sie von Ihrem AWS Konto aus an die Amazon Pinpoint Pinpoint-API stellen können.

Die maximale Größe einer Aufrufnutzlast (Anforderung und Antwort) beträgt 7 MB, sofern für einen bestimmten Ressourcentyp nichts anderes angegeben ist. Wenn Sie ermitteln möchten, ob eine Ressource über ein anderes Kontingent verfügt, lesen Sie den entsprechenden Abschnitt dieses Themas für diesen Ressourcentyp.

Die maximale Anzahl von Anforderungen variiert je nach Kontingenttyp und API-Vorgang. Amazon Pinpoint implementiert zwei Arten von Kontingenten für API-Anfragen:

- Ratenkontingente: Auch als Ratenlimits bezeichnet. Diese Art von Kontingent definiert die maximale Anzahl von Anforderungen, die Sie pro Sekunde für einen bestimmten Vorgang ausführen können. Damit wird die Rate der Anforderungen gesteuert, die pro Konto gesendet oder empfangen werden.
- Burst-Kontingente: Diese Art von Kontingent wird auch als Burst-Limits oder Burst-Kapazität bezeichnet und definiert die maximale Anzahl von Anfragen, die gleichzeitig für ein Konto bearbeitet werden.

In der folgenden Tabelle sind die Raten- und Drosselungskontingente für die Amazon-Pinpoint-API aufgeführt.

Operation	Standard-Burst/Ratenkontingent (Anforderungen pro Sekunde)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10

Operation	Standard-Burst/Ratenkontingent (Anforderungen pro Sekunde)
CreateImportJob	300
CreatePushTemplate	10
CreateSegment	25
CreateSmsTemplate	10
CreateVoiceTemplate	10
DeleteCampaign	25
DeleteEndpoint	5
DeleteSegment	25
GetEndpoint	10
PhoneNumberValidate	20
PutEvents	15
SendMessages	4.000
SendUsersMessages	6 000
UpdateCampaign	25
UpdateEmailTemplate	10
UpdateEndpoint	10
UpdateEndpointsBatch	2
UpdateInAppTemplate	10
UpdatePushTemplate	10
UpdateSegment	25

Operation	Standard-Burst/Ratenkontingent (Anforderungen pro Sekunde)
UpdateSmsTemplate	10
UpdateVoiceTemplate	10
Alle anderen Vorgänge	300

In der folgenden Tabelle sind die Importkontingente der Datei für `CreateImportJob` beschrieben.

Operation	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl von Importdateien	10 000 Dateien pro Importauftrag	Nein

Wenn Sie eines dieser Kontingente überschreiten, drosselt Amazon Pinpoint die Anforderung, d. h. eine ansonsten gültige Anforderung wird abgelehnt und ein Fehler `TooManyRequests` wird zurückgegeben. Die Drosselung basiert auf der Gesamtzahl der Anforderungen, die Sie von Ihrem Konto für einen bestimmten Vorgang in einer bestimmten AWS-Region aussenden. Außerdem werden Drosselungsentscheidungen für jeden Vorgang unabhängig berechnet. Wenn Amazon Pinpoint beispielsweise eine Anforderung für den `SendMessages`-Vorgang drosselt, kann eine gleichzeitige Anforderung für den `UpdateEndpoint`-Vorgang erfolgreich abgeschlossen werden.

Kampagnenkontingente

Die folgenden Kontingente gelten für die Kampagnen-Ressource der Amazon-Pinpoint-API.

Die folgenden Kontingente gelten pro Person AWS-Region und einige können erhöht werden. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Aktive Kampagnen	200 pro Konto	Nein

Ressource	Standardkontingent	Zur Erhöhung berechtigt
	Note Eine aktive Kampagne ist eine Kampagne, die noch nicht abgeschlossen wurde oder fehlgeschlagen ist. Aktive Kampagnen haben des Status `SCHEDULED` , `EXECUTING` oder `PENDING_NEXT_RUN` .	
Maximale Segmentgröße	Für importierte Segmente: 100 000 000 pro Kampagne. Für dynamische Segmente: unbegrenzt	Nein

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Ereignisbasierte Kampagnen	Jedes Projekt kann bis zu 25 Kampagnen enthalten, die bei Auftreten von Ereignissen gesendet werden. Kampagnen, die ereignisbasierte Auslöser verwenden, müssen dynamische Segmente verwenden. Importierte Segmente können nicht verwendet werden. Wenn Sie Ihre App mithilfe eines AWS Mobile SDK in Amazon Pinpoint integrieren, werden Nachrichten aus ereignisbasierten Kampagnen nur an Kunden gesendet, deren Apps AWS Mobile SDK for Android Version 2.7.2 oder höher oder Version 2.6.30 oder AWS Mobile SDK for iOS höher ausführen. Wenn Amazon Pinpoint nicht innerhalb von fünf Minuten eine Nachricht aus einer ereignisbasierten Kampagne zustellen kann, verwirft der Service die Nachricht und versucht nicht, sie erneut zuzustellen.	Nein

E-Mail-Kontingente

Für den E-Mail-Kanal gelten die Kontingente in den folgenden Abschnitten.

E-Mail-Nachrichtenkontingente

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Nachrichtengröße, einschließlich Anhängen	10 MB pro Nachricht	Nein
Anzahl der verifizierten Identitäten	10 000 Identitäten **Note** Identitäten beziehen sich auf die E-Mail-Adressen oder Domänen oder eine beliebige Kombination davon. Jede E-Mail, die Sie mit Amazon Pinpoint senden, muss von einer verifizierten Identität gesendet werden.	Nein

Kontingente für E-Mail-Sender und -Empfänger

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Absenderadresse	Alle Absenderadressen oder -domänen müssen verifiziert werden.	Nein

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Empfängeradresse	Wenn sich Ihr Konto in der Sandbox befindet, müssen alle Empfänger-E-Mail-Adressen oder -Domänen verifiziert werden. Wenn sich Ihr Konto nicht mehr in der Sandbox befindet, können Sie Ihre Nachricht en an jede gültige Adresse senden.	Ja
Empfängeranzahl pro Nachricht	50 Empfänger pro Nachricht	Nein
Anzahl der Identitäten, die Sie verifizieren können	10.000 Identitäten pro Region AWS **Note** Identitäten beziehen sich auf die E-Mail-Ad ressen oder Domänen oder eine beliebige Kombination davon. Jede E-Mail, die Sie mit Amazon Pinpoint senden, muss von einer verifizierten Identität gesendet werden.	Nein

E-Mail-Sendekontingente

Die Sendequote, die Senderate und die Sandbox-Begrenzungen werden von den beiden Diensten in derselben Region gemeinsam genutzt. Wenn Sie Amazon SES in us-east-1 verwenden und Sie aus der Sandbox entfernt wurden und Ihre Versandquote/Versandrate erhöht wurde, gelten alle diese Änderungen für Ihr Pinpoint-Konto in us-east-1.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Anzahl der E-Mails, die innerhalb von 24 Stunden gesendet werden können (Versandquote)	Wenn sich Ihr Konto in der Sandbox befindet, 200 E-Mails pro 24-Stunden-Zeitraum. Wenn Ihr Konto nicht mehr in der Sandbox ist, variiert das Kontingent basierend auf Ihrem speziellen Anwendung sfall. **Note** Dieses Kontingent basiert auf der Anzahl der Empfänger, im Gegensatz zu der Anzahl der individue llen gesendeten Nachrichten. Jede E-Mail-Adresse auf der An-Zeile ist ein Empfänger.	Ja
Anzahl der E-Mails, die pro Sekunde gesendet werden können (Senderate)	Wenn sich Ihr Konto in der Sandbox befindet, 1 E-Mail pro Sekunde.	Ja

Ressource	Standardkontingent	Zur Erhöhung berechtigt
	Wenn Ihr Konto nicht mehr in der Sandbox ist, variiert die Rate basierend auf Ihrem speziellen Anwendungsfall. **Note** Diese Rate basiert auf der Anzahl der Empfänger, im Gegensatz zu der Anzahl der individuellen gesendeten Nachrichten. Jede E-Mail-Adresse auf der An-Zeile ist ein Empfänger.	

Endpunktkontingente

Die folgenden Kontingente gelten für die Endpunkte-Ressource der Amazon-Pinpoint-API.

Pro Endpunkt werden maximal 250 Attribute unterstützt und die maximale Endpunktgröße beträgt 15 KB. Diese Anzahl von Attributen kann jedoch durch die Gesamtgröße eines Endpunkts, der alle Attribute umfasst, begrenzt sein. Wenn Sie beim Hinzufügen von Attributen zu Ihrer Vorlage auf Fehler stoßen, sollten Sie erwägen, die Datenmenge in jedem Attribut oder die Anzahl der Attribute zu verringern.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Endpunktgröße	Maximale Größe 15 KB	Nein
Attribute werden den Parametern `Attributes` , `Metrics` und `UserAttri`	250 für alle Attributparameter pro Anwendung	Nein

Ressource	Standardkontingent	Zur Erhöhung berechtigt
`butes` gemeinsam zugewiesen		
Dem Parameter `Attributes` zugewiesene Attribute	250 für alle Attributparameter pro Anwendung	Nein
Dem Parameter `Metrics` zugewiesene Attribute	250 für alle Attributparameter pro Anwendung	Nein
Dem Parameter `UserAttributes` zugewiesene Attribute	250 für alle Attributparameter pro Anwendung	Nein
Attributnamenlänge	50 Zeichen	Nein
Attributwertlänge	100 Zeichen	Nein
`EndpointBatchItem` Objekte in einer `EndpointBatchRequest` Nutzlast	100 pro Nutzlast. Die Nutzlast darf nicht größer als 7 MB sein.	Nein
Endpunkte mit derselben Benutzer-ID	15 verschiedene Endpunkte pro Benutzer-ID	Nein
Den `Attributes` -Parameterattributen zugewiesene Werte	50 pro Attribut	Nein
Den `UserAttributes` -Parameterattributen zugewiesene Werte	50 pro Attribut	Nein

Endpunkt-Importkontingente

Die folgenden Kontingente gelten für den Import von Endpunkten in Amazon Pinpoint.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Aktive Importaufträge	10 pro Konto Importaufträge werden nur dann auf dieses Kontingent angerechnet, wenn sie ausgeführt werden. Sobald der Importauftrag abgeschlossen ist, wird er nicht mehr auf dieses Kontingent angerechnet.	Nein
Importgröße	1 GB pro Importauftrag Wenn beispielsweise jeder Endpunkt 4 KB oder kleiner ist, können Sie 250 000 Endpunkte importieren.	Nein

Ereignisaufnahmekontingente

Die folgenden Kontingente gelten für die Erfassung von Ereignissen mithilfe der AWS Mobile SDKs und der Events-Ressource der Amazon Pinpoint Pinpoint-API.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl von benutzerdefinierten Ereignistypen	1,500 pro App	Nein
Maximale Anzahl an benutzerdefinierten Attributschlüsseln	500 pro App	Nein

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl an benutzerdefinierten Attributwerten pro Attributschlüssel	100.000. Jede Zahl, die 100 000 überschreitet, ist immer noch registriert, aber nicht in der Amazon-Pinpoint-Analytics-Konsole verfügbar.	Nein
Maximale Zeichenanzahl pro Attributschlüssel	50	Nein
Maximale Zeichenanzahl pro Attributwert	200. Wenn die Anzahl der Zeichen 200 überschreitet, wird das Ereignis verworfen.	Nein
Maximale Anzahl an benutzerdefinierten Metrikschlüsseln	500 pro App	Nein
Maximale Anzahl von Ereignissen in einer Anforderung	100 pro Anforderung	Nein
Maximale Größe einer Anforderung	4 MB	Nein
Maximale Größe eines einzelnen Ereignisses	1,000 KB	Nein
Maximale Anzahl von Attributschlüsseln und metrischen Schlüsseln für jedes Ereignis	40 pro Anforderung	Nein

Journey-Kontingente

Für Journeys gelten folgende Kontingente.

Die folgenden Kontingente gelten pro AWS-Region und einige können erhöht werden. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl der aktiven Journeys	50 pro -Konto	Nein
Maximale Anzahl von aktiven EventTriggeredJourneys	20 pro Konto	Nein
Maximale Anzahl von Journey-Aktivitäten	40 pro Journey	Nein
Maximale Segmentgröße	Für importierte Segmente: 100 000 000 pro Journey. Für dynamische Segmente: unbegrenzt	Nein
Maximale Anzahl an Kontaktcenter-Aktivitäten	3 pro Journey	Nein
Maximale Anzahl von Geschlossene-Tage-Regeln	20 pro Kanal	Nein
Maximale Länge des Namens für eine Geschlossene-Tage-Regel	150 Zeichen	Nein
Maximale Anzahl von Tagen zwischen Start- und Endzeit für eine Geschlossene-Tage-Regel	7 Tage	Nein
Maximale Anzahl von Offene-Stunden-Regeln	4 pro Tag	Nein

Lambda-Kontingente

Die folgenden Kontingente gelten für Amazon-Pinpoint-Konfigurationen zum Abrufen und Verarbeiten von Daten aus Lambda

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Größe einer Aufrufnutzlast (Anforderung und Antwort) für eine Lambda-Funktion	6 MB	Nein
Maximale Wartezeit, bis eine Lambda-Funktion Daten verarbeitet	15 Sekunden	Nein
Die maximale Anzahl von Ereignisattributen pro Endpunkt	5	Nein
Maximale Anzahl von Zeichen für einen Ereignisattributna men	128 Zeichen	Nein
Maximale Anzahl von Zeichen für einen Ereignisattributwert	128 Zeichen	Nein
Maximale Anzahl von Tagen, die eine Journey laufen kann	540 Tage	Nein

Machine Learning-Kontingente

Die folgenden Kontingente gelten für Amazon-Pinpoint-Konfigurationen zum Abrufen und Verarbeiten von Daten aus ML-Modellen (Machine Learning).

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl an Modellkonfigurationen	1 pro Nachrichtenvorlage 100 pro Konto	Nein
Maximale Anzahl an Empfehlungen	5 pro Endpunkt oder Benutzer	Nein
Maximale Anzahl der empfohlenen Attribute pro Endpunkt oder Benutzer	1, wenn die Attributwerte nicht von einer AWS Lambda -F unktion verarbeitet werden 10, wenn die Attributwerte von einer AWS Lambda -Funktion verarbeitet werden	Nein
Maximale Länge eines empfohlenen Attributnamens	50 Zeichen für einen Attributn amen 25 Zeichen für einen Attributa nzeigenamen (der Name, der im Attributfinder auf der Konsole angezeigt wird)	Nein
Maximale Länge eines empfohlenen Attributwerts, der von Amazon Personalize abgerufen wird	100 Zeichen	Nein
Maximale Größe einer Aufrufnutzlast (Anforderung und Antwort) für eine Lambda-Funktion	6 MB	Nein
Maximale Wartezeit, bis eine Lambda-Funktion Daten verarbeitet	15 Sekunden	Nein

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl der Versuche, eine Lambda-Funktion aufzurufen	3 Versuche	Nein

Je nachdem, wie Sie Amazon Pinpoint für die Verwendung eines ML-Modells konfigurieren, können zusätzliche Kontingente gelten. Weitere Informationen zu Amazon Personalize finden Sie unter Kontingente im Amazon-Personalize-Entwicklerhandbuch. Weitere Informationen zu AWS Lambda - Kontingenten finden Sie unter Kontingente im AWS Lambda Entwicklerhandbuch.

Kontingente für Nachrichtenvorlagen

Die folgenden Kontingente gelten für Nachrichtenvorlagen für Ihr Amazon-Pinpoint-Konto.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl von Nachrichtenvorlagen	20 000 pro Konto	Nein
Maximale Versionsanzahl	5.000 pro Vorlage	Nein
Maximale Anzahl von Zeichen in einer E-Mail-Vorlage	600 000 Zeichen	Nein
Maximale Anzahl von Zeichen in einer In–App-Vorlage	200 000 Zeichen	Nein
Maximale Anzahl von Zeichen in den Standardvorlagenteilen einer Pushbenachrichtigungsvorlage	4 000 Zeichen	Nein
Maximale Anzahl von Zeichen in ADM-spezifischen Vorlagenteilen einer Push-Benachrichtigungsvorlage	6 000 Zeichen	Nein

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl von Zeichen in APN-spezifischen Vorlagenteilen einer Push-Benachrichtigungsvorlage	4 000 Zeichen	Nein
Maximale Anzahl von Zeichen in Baidu-spezifischen Vorlagenteilen einer Push-Benachrichtigungsvorlage	4 000 Zeichen	Nein
Maximale Anzahl von Zeichen in FCM-spezifischen Vorlagenteilen einer Push-Benachrichtigungsvorlage	4 000 Zeichen	Nein
Maximale Anzahl von Zeichen in einer SMS-Vorlage	1 600 Zeichen	Nein
Maximale Anzahl von Zeichen in einer Sprach-Vorlage	10,000 Zeichen	Nein

Push-Benachrichtigungskontingente

Die folgenden Kontingente gelten für Nachrichten, die Amazon Pinpoint über Push-Benachrichtigungskanäle sendet.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl der Push-Benachrichtigungen, die pro Sekunde in einer Kampagne gesendet werden können	25.000 Benachrichtigungen pro Sekunde	Ja

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Amazon Device Messaging (ADM)-Nachrichte – Nutzlastgröße	6 KB pro Nachricht	Nein
Nachricht von Apple Push Notification Service (APNs) – Nutzlastgröße	4 KB pro Nachricht	Nein
APNs-Sandbox-Nachricht – Nutzlastgröße	4 KB pro Nachricht	Nein
Baidu Cloud Push-Nachricht – Nutzlastgröße	4 KB pro Nachricht	Nein
Nachricht von Firebase Cloud Messaging (FCM) – Nutzlastgröße	4 KB pro Nachricht	Nein

In-App-Nachrichtenkontingente

Das folgende Kontingent gilt für In-App-Nachrichten, die Sie mit Amazon Pinpoint verwalten.

Die folgenden Kontingente gelten pro Person AWS-Region und einige können erhöht werden. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Häufigkeit, mit der Sie die `GetInAppMessages` -API pro Sekunde aufrufen können.	5 000 Anforderungen pro Sekunde	Ja
In-App-Messaging-Kampagnen	Jedes Projekt kann bis zu 25 Kampagnen umfassen, die	Ja, siehe Anfordern einer Kontingenterhöhung im

Ressource	Standardkontingent	Zur Erhöhung berechtigt
	den In-App-Nachrichten-Kanal verwenden.	Service-Quotas-Benutzerhandbuch

Segmentkontingente

Die folgenden Kontingente gelten für die Segmente-Ressource der Amazon-Pinpoint-API.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximale Anzahl der Dimensionen, die zur Erstellung eines Segments verwendet werden können	100 pro Segment	Nein
Maximale Anzahl von Segmentgruppen pro Segment	5	Nein
Maximale Anzahl von Quellsegmenten pro Segment	5	Nein
Maximale Tiefe der Quellsegmente. Wenn ein Segment beispielsweise ein Quellsegment hat, das auch ein Quellsegment hat, ist die Tiefenkette nicht länger als dieses Limit.	5	Nein

SMS-Kontingente

Die folgenden Kontingente gelten für den SMS-Kanal.

Weitere Informationen zu den SMS-Kosten finden Sie unter Grundlegendes zu SMS-Abrechnungs- und Nutzungsberichten für Amazon Pinpoint im Amazon-Pinpoint-Benutzerhandbuch.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Ausgabenschwelle	1,00 USD pro Konto	Ja, aber die Ausgabenlimits variieren je nach Region. Sie müssen die Region(en) angeben, in denen Sie eine Erhöhung benötigen.
Anzahl der SMS-Nachrichten, die pro Sekunde gesendet werden können (Senderate#)	Variiert je nach Zielland und Ursprungs-Telefonnummer. Weitere Informationen finden Sie unter Message Parts per Second (MPS)-Limits im Amazon-Pinpoint-Benutzerhandbuch.	Ja, möglicherweise benötigen Sie jedoch eine Telefonnummer, die einen höheren Durchsatz unterstützt. Wenn Sie sich nicht sicher sind, welchen Rufnummerntyp Sie verwenden sollen, wenden Sie sich an AWS Support Ihren AWS Account Manager, um weitere Informationen zu erhalten Wenn Sie zum Senden von Nachrichten eine alphanumerische Absender-ID verwenden, können Sie möglicherweise Ihre Durchsatzrate erhöhen. Um herauszufinden, ob eine Durchsatzsteigerung für Ihre Absender-ID verfügbar ist, öffnen Sie eine Absender-ID-Anfrage in der Support-Center-Konsole. Geben Sie in Ihrer Anfrage Ihre bestehende Absender-ID, das Land, in dem Sie diese Absender-ID verwenden, und

Ressource	Standardkontingent	Zur Erhöhung berechtigt
		die Durchsatzrate, die Sie anfordern möchten, an.
Anzahl der SMS-Nachrichten, die pro Sekunde an einen einzelnen Empfänger gesendet werden können	1 Nachricht pro Sekunde	Nein
Anzahl der Amazon-SNS-Themen für ein- und ausgehende SMS-Nachrichten	100,000 pro Konto	Ja
Anzahl der Schlüsselwörter für ein- und ausgehende SMS-Nachrichten	30 Schlüsselwörter pro Zahl	Ja
Anzahl der SMS- und Sprachnachrichten-Nummern	(25 pro Konto und Region)	Ja
Anzahl der dedizierten Telefonnummern	25 pro Konto	Ja
Anzahl der Opt-Out-Listen Hinweis: Die erforderliche Standard-Opt-Out-Liste wird auf dieses Kontingent angerechnet.	25 pro Konto	Ja
Anzahl von Konfigurationssätzen	25 pro Konto	Ja
Anzahl der Ereignisziele	5 pro Konfigurationssatz	Nein
Anzahl der verifizierten Zieltelefonnummern in der SMS-Sandbox	10 pro Konto	Ja

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Anzahl der Telefonnu mmernpools	25 pro Konto	Ja
Anzahl der Ursprungsidentität en, die einem Telefonnu mmernpool zugeordnet werden können	100 pro Rufnummernpool	Ja

10 DLC-Kontingente

Die folgenden Kontingente gelten für SMS-Nachrichten, die über 10DLC-Telefonnummern gesendet werden. 10DLC-Nummern können nur zum Senden von Nachrichten an Empfänger in den USA verwendet werden.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Maximal 10 DLC-Unter nehmen pro AWS-Konto	25	Ja
Max. 10DLC-Kampagnen pro 10DLC-Unternehmen	10	Ja
Max. 10DLC-Nummern pro 10DLC-Kampagne	49	Nein

Sprachnachrichtenkontingente

Die folgenden Kontingente gelten für den Sprachkanal.

> **Note**
>
> Wenn Ihr Konto aus der Sandbox entfernt wird, sind Sie automatisch für die maximalen Kontingente qualifiziert, die in der folgenden Tabelle aufgeführt sind.

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Anzahl von Sprachnachrichten, die Sie in einem Zeitraum von 24 Stunden senden können.	Wenn sich Ihr Konto in der Sandbox befindet: 20 Nachrichten	Nein
Anzahl der Sprachnachrichten, die innerhalb von 24 Stunden an einen einzelnen Empfänger gesendet werden können	5 Nachrichten	Nein
Anzahl von Sprachnachrichten, die Sie pro Minute senden können.	Wenn sich Ihr Konto in der Sandbox befindet: 5 Aufrufe pro Minute Wenn sich Ihr Konto nicht in der Sandbox befindet: 20 Aufrufe pro Minute	Nein
Anzahl der Sprachnachrichten, die pro Sekunde von einer einzigen Nummer gesendet werden können	1 Nachricht pro Sekunde	Nein
Länge der Sprachnachricht	Wenn sich Ihr Konto in der Sandbox befindet: 30 Sekunden Wenn sich Ihr Konto nicht in der Sandbox befindet: 5 Minuten	Nein
Möglichkeit zum Senden von Sprachnachrichten an internationale Telefonnummern.	Wenn sich Ihr Konto in der Sandbox befindet, können Sie Nachrichten nur in den folgenden Ländern an Empfänger senden:	Nein

Ressource	Standardkontingent	Zur Erhöhung berechtigt
	• Australien • Kanada • Deutschland • Hong Kong • Israel • Japan • Mexiko • Singapur • Schweden • Vereinigte Staaten • Großbritannien und Nordirland Wenn sich Ihr Konto außerhalb der Sandbox befindet, können Sie Nachrichten an Empfänger in einem beliebigen Land senden. **Note** Für internationale Anrufe fallen zusätzliche Gebühren an, die je nach Zielland oder Region variieren.	

Ressource	Standardkontingent	Zur Erhöhung berechtigt
Anzahl der Zeichen in einer Sprachnachricht	3 000 kostenpflichtige Zeichen in Wörtern, die gesprochen werden 6 000 Zeichen insgesamt, einschließlich kostenpflichtige Zeichen und SSML-Tags	Nein
Anzahl von Konfigurationssätzen	10.000 Sprachkonfigurationssätze pro AWS Region	Nein

Beantragen einer Kontingenterhöhung

Wenn der Wert in der Spalte Eligible for Increase (Zur Erhöhung berechtigt) in einer der vorhergehenden Tabellen Yes (Ja) ist, können Sie eine Änderung des Kontingents beantragen.

So fordern Sie eine Kontingenterhöhung an

1. Melden Sie sich AWS Management Console unter https://console.aws.amazon.com/ bei der an.
2. Erstellen Sie einen neuen AWS Support-Fall unter https://console.aws.amazon.com/support/home#/case/create.
3. Wählen Sie auf der Registerkarte Offene Support-Fälle die Option Fall erstellen aus.
4. Klicken Sie auf Sie wünschen eine Erhöhung des Servicelimits?.
5. Wählen Sie unter Erhöhung der Servicequote für Service eine der folgenden Optionen aus:

   - Um eine Kontingenterhöhung in Bezug auf den E-Mail-Kanal anzufordern, wählen Sie Pinpoint Email.
   - Um eine Erhöhung des Kontingents für SMS-Ausgabenlimits oder SMS-Senderate zu beantragen, wählen Sie Pinpoint-SMS. Für alle anderen SMS-Kontingenterhöhungen wählen Sie Pinpoint
   - Um eine Kontingenterhöhung in Bezug auf den Sprachkanal anzufordern, wählen Sie Pinpoint Voice.
   - Um eine Kontingenterhöhung in Bezug auf ein anderes Amazon-Pinpoint-Feature anzufordern, wählen Sie Pinpoint.

6. Je nachdem, für welchen Dienst Sie sich entscheiden, werden Sie möglicherweise aufgefordert, Folgendes einzugeben:

   - (Optional) Geben Sie unter Link zur Website oder App angeben, die die SMS-Nachrichten senden wird Informationen über die Website, die Anwendung oder den Service an, die bzw. der SMS-Nachrichten senden wird.
   - (Optional) Wählen Sie für Art der Nachrichten, die gesendet werden sollen die Art der Nachrichten aus, die Sie mit Ihren Langwahlnummern senden möchten:
     - One-time Password (Einmaliges Passwort) – Nachrichten, die für Ihre Kunden Passwörter zur Authentifizierung bei Ihrer Website oder Anwendung bereitstellen.
     - Promotional (Werbung) – Nicht kritische Nachrichten, die Ihr Unternehmen oder Ihren Service bewerben, wie beispielsweise Sonderangebote oder Ankündigungen.
     - Transactional (Transaktionsnachrichten) – Wichtige Informationsmeldung, die Kundentransaktionen unterstützen, wie beispielsweise Bestellbestätigungen oder Kontowarnungen. Transaktionsnachrichten dürfen keine Werbeaktionen oder Marketinginhalte enthalten.
   - (Optional) Für welche AWS Region werden Sie Nachrichten senden, wählen Sie die Region aus, aus der Sie Nachrichten senden möchten.
   - (Optional) Geben Sie für In welche Länder möchten Sie Nachrichten senden das Land oder die Region ein, in dem bzw. der Sie Kurzwahlnummern erwerben möchten.
   - (Optional) Geben Sie unter Wie entscheiden sich Ihre Kunden dafür, Nachrichten von Ihnen zu erhalten Einzelheiten zu Ihrem Anmeldeverfahren an.
   - (Optional) Geben Sie im Feld Bitte geben Sie die Nachrichtenvorlage an, die Sie verwenden möchten, um Nachrichten an Ihre Kunden zu senden die Vorlage ein, die Sie verwenden werden.

7. Gehen Sie unter Requests (Anfragen) wie folgt vor:

   - Wählen Sie für Region Ihre AWS-Region.
   - Wählen Sie für Resource Type (Ressourcentyp) die Option General Limits (Allgemeine Limits) aus. Das Feld Ressourcentyp ist nur für einige Dienste vorhanden.
   - Wählen Sie unter Kontingent das zu ändernde Kontingent aus.
   - Geben Sie unter Neuer Kontingentwert einen neuen Wert für das Kontingent ein.
   - Um eine Erhöhung desselben Kontingents in einer zusätzlichen Anfrage zu beantragen AWS-Region, wählen Sie Weitere Anfrage hinzufügen, wählen Sie dann die zusätzliche Anfrage aus AWS-Region und füllen Sie die neue Anfrage aus.

8. Wählen Sie das Kontingent aus, das Sie erhöhen möchten, und geben Sie dann den gewünschten neuen Wert ein.
9. Erläutern Sie unter Fallbeschreibung, warum Sie die Erhöhung des Kontingents beantragen.
10. Wählen Sie unter Kontaktoptionen für Bevorzugte Kontaktsprache die Sprache aus, die Sie bei der Kommunikation mit dem AWS Support-Team bevorzugen.
11. Wählen Sie unter Kontaktmethode Ihre bevorzugte Methode für die Kommunikation mit dem AWS Support-Team aus.
12. Wählen Sie Absenden aus.

Das AWS Support-Team gibt innerhalb von 24 Stunden eine erste Antwort auf Ihre Anfrage.

Da wir verhindern möchten, dass unerwünschte oder schädliche Inhalte in unseren Systemen eingehen, müssen wir jede Anfrage sorgfältig prüfen. Nach einer erfolgreichen Prüfung kommen wir Ihrer Anfrage innerhalb dieses 24-Stunden-Zeitraums nach. Für den Fall, dass wir weitere Informationen von Ihnen benötigen, kann die Bearbeitung Ihrer Anfrage länger dauern.

Wenn Ihr Anwendungsfall gegen unsere Richtlinien verstößt, können wir Ihrer Anfrage möglicherweise nicht nachkommen.

Dokumentverlauf für Amazon Pinpoint

In der folgenden Tabelle werden wichtige Änderungen an den einzelnen Versionen des Amazon-Pinpoint-Entwicklerhandbuchs nach Dezember 2018 beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

- Letzte Aktualisierung der Dokumentation: 16. November 2023

Änderung	Beschreibung	Datum
E-Mail-Header	Sie können Ihren E-Mail-Nachrichten E-Mail-Header hinzufügen. Weitere Informationen finden Sie unter Senden einer E-Mail mit Abmelde-Headern.	7. Mai 2024
E-Mail-Orchestrierung	Amazon Pinpoint hat aktualisiert, wie es Ihre Amazon SES SES-Ressourcen zum Senden von E-Mails verwendet. Weitere Informationen finden Sie unter IAM-Rolle für das Senden von E-Mails mit Amazon SES.	30. April 2024
Amazon Pinpoint hat die Dokumentation seines Benutzerhandbuchs aktualisiert	Die Themen zur Verwaltung von SMS- und Sprachressourcen werden jetzt zum Amazon Pinpoint SMS-Benutzerhandbuch weitergeleitet. Weitere Informationen finden Sie im Amazon Pinpoint SMS-Benutzerhandbuch.	8. Februar 2024
Amazon-Pinpoint-Kontingente	Es wurden Kontingente für die Regeln „Maximale	19. Dezember 2023

	Anzahl geschlossener Tage“, „Maximale Länge geschloss ener Tage“, „Maximale Anzahl von Tagen zwischen Start- und Endzeit für die geschloss ene-Tage-Regel“ und Regeln „Maximale Anzahl an geöffneten Stunden“ hinzugefü gt. Weitere Informationen finden Sie unter Amazon-Pi npoint-Kontingente.	
Amazon Pinpoint hat die Dokumentation seines Benutzerhandbuchs aktualisi ert	Aktuelle Informationen zur Erstellung, Konfiguration und Verwaltung Ihrer Amazon-Pi npoint-SMS- und -Sprachre ssourcen finden Sie im neuen Amazon-Pinpoint-SMS-Benutzerhandbuch.	16. November 2023
Amazon-Pinpoint-Kontingente	Die Kontingente für UpdateEndpointsBatch, UpdateEndpoint PutEvents DeleteEndpoint, und wurden aktualisiert GetEndpoint. Weitere Informationen finden Sie unter Amazon-Pinpoint-Ko ntingente.	22. September 2023

Amazon-Pinpoint-Kontingente	Die Kontingente für CreateEmailTemplate CreateSmsTemplate, CreatePushTemplate ,, CreateInAppTemplate, CreateVoiceTemplate, UpdateEmailTemplate, UpdateSmsTemplate UpdatePushTemplate UpdateInAppTemplate, UpdateVoiceTemplate und wurden aktualisiert CreateImportJob. Weitere Informationen finden Sie unter Amazon-Pinpoint-Kontingente.	12. September 2023
Ausführungsmetriken zu Journey und Kampagne	Neue analytische Metriken wurden für Journeys und Kampagnen hinzugefügt. Weitere Informationen finden Sie unter Ausführungsmetriken zu Journey und Kampagne.	25. April 2023
Erstellen eines Schnittstellen-VPC-Endpunkts für Amazon Pinpoint	Amazon Pinpoint unterstützt jetzt VPC-Endpunkte mit Schnittstellen. Weitere Informationen finden Sie unter Erstellen eines Schnittstellen-VPC-Endpunkts für Amazon Pinpoint	11. April 2023

Verschlüsselung während der Übertragung	Ab 22.03.2023 unterstützt Amazon Pinpoint TLS 1.0 nicht mehr, Sie können jedoch weiterhin TLS 1.2 oder höher verwenden. Weitere Informationen finden Sie unter Verschlüsselung während der Übertragung.	20. März 2023
Amazon-Pinpoint-Kontingente	Das Verfahren zur Beantragung einer Kontingenterhöhung für Kampagnen, Journeys und In-App-Nachrichten wurde aktualisiert. Weitere Informationen finden Sie unter Amazon-Pinpoint-Kontingente.	16. Dezember 2022
Regionale Verfügbarkeit	Amazon Pinpoint ist jetzt in der Region USA Ost (Ohio) verfügbar.	5. Oktober 2022
Beispielaktualisierungen für IAM-Rollen	Im gesamten Dokument wurden mehrere Beispiele für IAM-Rollen aktualisiert, um sie besser an die bewährten Sicherheitsverfahren anzupassen.	27. Mai 2022

SMS- und Sprachnachrichten-API, Version 2	Amazon Pinpoint enthält jetzt eine spezielle API zum Senden von SMS- und Sprachnachrichten. Diese API enthält neue Features wie Konfigurationssätze, Pools und Opt-Out-Listen, die für Kunden hilfreich sind, die SMS und Sprachnachrichten transaktionell versenden. Weitere Informationen finden Sie unter Verwenden der Amazon-Pinpoint-SMS- und -Sprachnachrichten-API.	1. April 2022
Erstellung und Validierung von Einmalpasswörtern	Amazon Pinpoint enthält jetzt eine Funktion, die Einmalpasswörter (OTPs) generiert und diese als SMS-Nachrichten an Ihre Benutzer sendet. Es enthält auch eine API zur Validierung der OTP-Codes, wenn Ihre Benutzer sie in Ihre Anwendung oder Website eingeben. Weitere Informationen finden Sie unter Senden und Überprüfen von Einmalpasswörtern (OTPs).	26. November 2021
In-App-Nachrichten	Es wurden Informationen zur Integration der In-App-Nahrichten-Funktion von Amazon Pinpoint in Ihre Apps hinzugefügt.	10. November 2021

Codebeispiele	Es wurde eine Bibliothek mit Codebeispielen für allgemeine Amazon Pinpoint--Operationen hinzugefügt.	3. November 2021
Projekt-Kontingente	Die maximale Anzahl von Amazon Pinpoint Pinpoint-Projekten liegt weiterhin bei 100, aber dieses Kontingen t kann jetzt erhöht werden, indem eine Anfrage zur Erhöhung des Service Limit mit AWS Support geöffnet wird.	11. Oktober 2021
Lambda-Richtlinienaktualisierungen.	Bestimmte Lambda-Ge nehmigungsrichtlinien müssen jetzt eine `AWS:SourceAccount` -Bedingung enthalten. Die Beispielr ichtlinien in den Themen Erstellen von benutzerdefinierten Kanälen in Amazon Pinpoint und Anpassung von Segmenten mit AWS Lambdawurden aktualisiert, um diese Anforderung zu erfüllen.	7. Oktober 2021
UpdateEndpoint	Die Amazon Pinpoint UpdateEndpoint API wird jetzt protokolliert CloudTrail.	16. November 2020
Custom attributes (Benutzer definierte Attribute)	Amazon Pinpoint unterstützt jetzt 250 Attribute in E-Mail-Nachrichtenvorlagen. Siehe Kontingente.	18. September 2020

Regionale Verfügbarkeit	Amazon Pinpoint ist jetzt in folgenden Regionen verfügbar : Asien-Pazifik (Tokio), Europa (London) und Kanada (Zentral) . Beachten Sie, dass die Amazon-Pinpoint-SMS- und -Sprachnachrichten-API in diesen Regionen nicht verfügbar ist.	10. September 2020
Regionale Verfügbarkeit	Amazon Pinpoint ist jetzt in der Region Asien-Pazifik (Tokio) verfügbar. Beachten Sie, dass die Amazon-Pinpoint-SMS- und -Sprachnachrichten-API Sprachnachrichten in dieser Region nicht unterstützt.	2. September 2020
Kampagnenereignisse	Informationen zu einem neuen `delivery_type` -Parameter für Kampagnenereignisse wurden zu Kampagnenereignissen hinzugefügt.	02. August 2020
Regionale Verfügbarkeit	Amazon Pinpoint ist jetzt in der Region Asien-Pazifik (Seoul) verfügbar. Beachten Sie, dass die Amazon-Pinpoint-API SMS- und -Sprachnachrichten in dieser Region nicht unterstützt.	31. Juli 2020
Regionale Verfügbarkeit	Amazon Pinpoint ist jetzt in der AWS GovCloud (US) Region verfügbar.	30. April 2020

Benutzerdefinierte Kanäle	Aktualisierte Informationen über das Erstellen von benutzerdefinierten Kanälen mithilfe von Lambda-Funktionen oder Webhooks.	23. April 2020
Machine Learning	Es wurden Informationen zum Abrufen personalisierter Empfehlungen aus Empfehlungsmodellen und zur optionalen Erweiterung dieser Empfehlungen mithilfe von Funktionen hinzugefügt. AWS Lambda	4. März 2020
Sicherheit	Es wurde ein Sicherheitskapitel, hinzugefügt, das Informationen zu verschiedenen Sicherheitskontrollen und Features von Amazon Pinpoint enthält.	4. Februar 2020
Journeys	Es wurden Informationen zur Verwendung von Amazon-Pinpoint-Journeys hinzugefügt, um automatisierte Workflows zu entwickeln, die Messaging-Aktivitäten für Projekte durchführen. Außerdem wurden Informationen zum Abfragen von Analysedaten für eine Teilmenge der Metriken hinzugefügt, die für Journeys gelten.	31. Oktober 2019

Analysen	Es wurden Verfahren hinzugefügt, die erklären, wie Analysedaten für Kampagnen und Transaktionsnachrichten abgefragt werden, und Informationen zur Verwendung von Abfrageergebnissen hinzugefügt.	17. Oktober 2019
Analysen	Es wurden Informationen zum Abfragen von Analysedaten für eine Teilmenge von Metriken hinzugefügt, die für Transaktions-E-Mail- und -SMS-Nachrichten gelten.	6. September 2019
Codebeispiele	Es wurden Codebeispiele hinzugefügt, die Sie verwenden können, um transaktionale Push-Benachrichtigungen unter Verwendung aller Services zu senden, die Amazon Pinpoint unterstützt.	30. Juli 2019
Analysen	Informationen zum Abfragen von Analysedaten für eine Teilmenge der Metriken hinzugefügt, die für Projekte (Anwendungen) und Kampagnen relevant sind.	24. Juli 2019

Segmente	Es wurde ein Tutorial hinzugefügt, in dem eine Lösung zum Importieren von Kundendaten aus externen Systemen (z. B. Salesforce oder Marketo) in Amazon Pinpoint beschrieben wird.	14. Mai 2019
Regionale Verfügbarkeit	Amazon Pinpoint ist jetzt in den Regionen AWS Asien-Pazifik (Mumbai) und Asien-Pazifik (Sydney) verfügbar.	25. April 2019
Verwenden von Postman mit Amazon Pinpoint	Es wurde ein Tutorial hinzugefügt, in dem beschrieben wird, wie Sie mithilfe von Postman mit der Amazon-Pinpoint-API interagieren.	8. April 2019
Tagging	Es wurden Informationen zum Markieren von Amazon-Pinpoint-Ressourcen hinzugefügt.	27. Februar 2019
SMS-Registrierung	Es wurden ein Tutorial-Kapitel und ein Tutorial hinzugefügt, in denen beschrieben wird, wie Sie eine Lösung für die Verarbeitung der SMS-Benutzerregistrierung erstellen.	27. Februar 2019

Codebeispiele	Es wurden Codebeispiele in mehreren Programmiersprachen hinzugefügt, um zu zeigen, wie Sie programmgesteuert E-Mail-, SMS- und Sprach-Nachrichten versenden können.	6. Februar 2019

Frühere Aktualisierungen

In der folgenden Tabelle werden wichtige Änderungen an den einzelnen Versionen des Amazon-Pinpoint-Entwicklerhandbuchs bis Dezember 2018 beschrieben.

Änderung	Beschreibung	Datum
Regionale Verfügbarkeit	Amazon Pinpoint ist jetzt in den Regionen AWS USA West (Oregon) und Europa (Frankfurt) verfügbar.	21. Dezember 2018
Sprachkanal	Mit dem neuen Amazon-Pinpoint-Sprachkanal können Sie Sprachnachrichten erstellen und über das Telefon an Ihre Kunden weiterleiten. Gegenwärtig können Sie Sprachnachrichten nur mit der Amazon-Pinpoint-SMS- und Sprachnachrichten-API senden.	15. November 2018
Europa (Irland) – Verfügbarkeit	Amazon Pinpoint ist jetzt in der Region AWS Europa (Irland) verfügbar.	25. Oktober 2018

Änderung	Beschreibung	Datum
Ereignis-API	Verwenden Sie die Amazon-Pinpoint-API, um Ereignisse aufzuzeichnen und mit Endpunkten zu verknüpfen.	7. August 2018
Codebeispiele für die Definition und die Abfrage von Endpunkten	Es wurden Codebeispiele hinzugefügt, die veranschaulichen, wie Sie Endpunkte definieren, aktualisieren, löschen und abrufen. Es werden Beispiele für die AWS CLI AWS SDK for Java, und die Amazon Pinpoint Pinpoint-API bereitgestellt. Weitere Informationen finden Sie unter Definieren Ihrer Zielgruppe für Amazon Pinpoint und Zugreifen auf Zielgruppendaten in Amazon Pinpoint.	7. August 2018
Endpunktexportberechtigungen	Konfigurieren Sie eine IAM-Richtlinie, die es Ihnen erlaubt, Amazon-Pinpoint-Endpunkte in einen Amazon-S3-Bucket zu exportieren.	1. Mai 2018
Verifizierung der Telefonnummer für SMS	Verwenden Sie die Amazon-Pinpoint-API, um eine Telefonnummer zu verifizieren und zu bestimmen, ob es sich um eine gültige Zieladresse für SMS-Nachrichten handelt.	23. April 2018

Änderung	Beschreibung	Datum
Themen für die Integration von Amazon Pinpoint aktualisiert	Integrieren Sie Amazon Pinpoint mithilfe von AWS SDKs oder Bibliotheken in Ihr Android-, iOS- oder JavaScript Anwendungsprogramm.	23. März 2018
AWS CloudTrail Protokollierung	Es wurden Informationen zur Protokollierung von Amazon Pinpoint API-Aufrufen mit CloudTrail hinzugefügt.	6. Februar 2018
Aktualisierte Service Quotas	Kontingente wurde mit zusätzlichen Informationen zu E-Mail-Kontingenten aktualisiert.	19. Januar 2018
Öffentliche Beta-Version für Amazon-Pinpoint-Erweiterungen	Verwenden Sie AWS Lambda Funktionen, um Segmente anzupassen oder benutzerdefinierte Messaging-Kanäle zu erstellen.	28. November 2017
Nutzlast-Kontingente für Push-Benachrichtigungen	Die Kontingente umfassen Nutzlastgrößen für mobile Push-Nachrichten.	25. Oktober 2017
Aktualisierte Service Quotas	Informationen über SMS- und E-Mail-Kanäle zu Kontingente hinzugefügt.	9. Oktober 2017
Mobile Push-Kanäle ADM und Baidu	Aktualisieren Sie Ihre App-Code für die Verarbeitung von Push-Benachrichtigungen aus den mobilen Push-Kanälen Baidu und ADM.	27. September 2017

Änderung	Beschreibung	Datum
Benutzer-IDs und Authentifizierungsereignisse bei Amazon-Cognito-Benutzerpools.	Wenn Sie Amazon-Cognito-Benutzerpools für die Verwaltung einer Benutzeranmeldung bei Ihren mobilen Apps verwenden, weist Amazon Cognito den Endpunkten Benutzer-IDs zu und meldet Authentifizierungsereignisse an Amazon Pinpoint.	26. September 2017
Benutzer-IDs	Weisen Sie Endpunkten Benutzer-IDs zu, um die App-Nutzung durch einzelne Benutzer zu überwachen. Es gibt Beispiele für die AWS SDKs für Mobilgeräte und SDK für Java.	31. August 2017
Authentifizierungsereignisse	Melden Sie Authentifizierungsereignisse, um zu erfahren, wie häufig sich Benutzer bei Ihrer App authentifizieren. Beispiele finden Sie unter Melden von Ereignissen in Ihrer Anwendung.	31. August 2017
Beispiel-Ereignisse aktualisiert	Die Beispiel-Ereignisse enthalten Ereignisse, die Amazon Pinpoint für E-Mail- und SMS-Aktivitäten streamt.	08. Juni 2017

Änderung	Beschreibung	Datum
Android-Sitzungsverwaltung	Verwalten von Sitzungen in Android-Apps durch Verwenden einer Klasse, die von der AWS Mobile Hub - Beispielanwendung bereitgestellt wird	20. April 2017
Aktualisierte Monetarisierungs-Ereignisbeispiele	Der Beispiel-Code wurde für die Meldung von Monetarisierungsereignissen aktualisiert.	31. März 2017
Ereignis-Streams	Amazon Pinpoint kann so konfiguriert werden, dass App- und Kampagnenereignisse an einen Kinesis-Stream gesendet werden.	24. März 2017
Berechtigungen	Funktionsweise von Amazon Pinpoint mit IAMInformationen darüber, wie Sie AWS Benutzern in Ihrem Konto und Benutzern Ihrer mobilen App Zugriff auf Amazon Pinpoint gewähren, finden Sie unter.	12. Januar 2017
Allgemeine Verfügbarkeit von Amazon Pinpoint	In dieser Version wird Amazon Pinpoint eingeführt.	1. Dezember 2016

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.