



Benutzer-Leitfaden

AWS Push-Nachrichten für Endbenutzer



AWS Push-Nachrichten für Endbenutzer: Benutzer-Leitfaden

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS End User Messaging Push?	1
Sind Sie zum ersten Mal Nutzer von AWS End User Messaging Push?	1
Funktionen von AWS End User Messaging Push	1
Zugreifen auf AWS End User Messaging Push	2
Regionale Verfügbarkeit	3
Einrichtung eines AWS-Konto	4
Melde dich an für eine AWS-Konto	4
Erstellen eines Benutzers mit Administratorzugriff	5
Erste Schritte	7
Eine Anwendung erstellen und Push-Kanäle aktivieren	8
Kontextuell	8
Voraussetzungen	9
Verfahren	9
Push-Kanäle deaktivieren	11
Eine Push-Nachricht senden	12
Weitere Ressourcen	25
Empfangen von Push-Benachrichtigungen in Ihrer Anwendung	26
Einrichten von Swift-Push-Benachrichtigungen	26
Mit Tokens arbeiten APNs	26
Einrichten von Android-Push-Benachrichtigungen	27
Einrichten von Flutter-Push-Benachrichtigungen	27
Einrichten von React-Native-Push-Benachrichtigungen	27
Erstellen einer Anwendung	27
Umgang mit Push-Benachrichtigungen	28
Löschen einer Anwendung	29
Kontextuell	29
Verfahren	29
Best Practices	30
Senden einer großen Anzahl von Push-Benachrichtigungen	30
Sicherheit	31
Datenschutz	32
Datenverschlüsselung	33
Verschlüsselung während der Übertragung	33
Schlüsselverwaltung	33

Datenschutz für den Datenverkehr zwischen Netzwerken	34
Identity and Access Management	35
Zielgruppe	35
Authentifizierung mit Identitäten	35
Verwalten des Zugriffs mit Richtlinien	37
So funktioniert AWS End User Messaging Push mit IAM	39
Beispiele für identitätsbasierte Richtlinien	44
Fehlerbehebung	49
Compliance-Validierung	51
Ausfallsicherheit	51
Infrastruktursicherheit	52
Konfigurations- und Schwachstellenanalyse	52
Bewährte Methoden für die Gewährleistung der Sicherheit	52
Überwachen	54
Überwachung mit CloudWatch	55
CloudTrail protokolliert	55
AWS Nachrichtenübermittlung an Endbenutzer: Geben Sie Informationen ein CloudTrail	55
Grundlegendes zu den Einträgen in der Push-Protokolldatei von AWS End User Messaging	57
AWS PrivateLink	58
Überlegungen	58
Erstellen eines Schnittstellenendpunkts	58
Erstellen einer Endpunktrichtlinie	59
Kontingente	61
Dokumentverlauf	62
.....	lxiii

Was ist AWS End User Messaging Push?

Note

Die Push-Benachrichtigungsfunktionen von Amazon Pinpoint heißen jetzt AWS End User Messaging.

Mit AWS End User Messaging Push können Sie Nutzer Ihrer Apps ansprechen, indem Sie Push-Benachrichtigungen über einen Push-Benachrichtigungskanal senden. Wir unterstützen Apple Push Notification Service (APNs), Firebase Cloud Messaging (FCM), Amazon Device Messaging (ADM) und Baidu Push.

Themen

- [Sind Sie zum ersten Mal Nutzer von AWS End User Messaging Push?](#)
- [Funktionen von AWS End User Messaging Push](#)
- [Zugreifen auf AWS End User Messaging Push](#)
- [Regionale Verfügbarkeit](#)

Sind Sie zum ersten Mal Nutzer von AWS End User Messaging Push?

Wenn Sie zum ersten Mal AWS End User Messaging Push verwenden, empfehlen wir Ihnen, zunächst die folgenden Abschnitte zu lesen:

- [Einrichtung eines AWS-Konto](#)
- [Erste Schritte mit AWS End User Messaging Push](#)
- [Anwendung erstellen und Push-Kanäle aktivieren](#)

Funktionen von AWS End User Messaging Push

Sie können Push-Benachrichtigungen an Ihre Apps senden, indem Sie separate Kanäle für die folgenden Push-Benachrichtigungsdienste verwenden:

- Firebase Cloud Messaging (FCM)
- Apple-Push-Benachrichtigungsdienst (APNs)

Note

Sie können APNs damit Nachrichten an iOS-Geräte wie iPhones und iPads sowie an den Safari-Browser auf macOS-Geräten wie Mac-Laptops und -Desktops senden.

- Baidu Cloud Push
- Amazon Device Messaging (ADM)

Zugreifen auf AWS End User Messaging Push

Erläutern Sie kurz die verschiedenen Möglichkeiten, auf den Dienst zuzugreifen, sei es über die Konsole, CLI oder API.

Sie können AWS End User Messaging Push über die folgenden Schnittstellen verwalten:

AWS Push-Konsole für Endbenutzer-Messaging

Die Weboberfläche, auf der Sie Push-Ressourcen für AWS Endbenutzernachrichten erstellen und verwalten. Wenn Sie sich für eine angemeldet haben AWS-Konto, können Sie über die Endbenutzer-Nachrichten-Push-Konsole auf die AWS Endbenutzer-Messaging-Push-Konsole zugreifen AWS-Managementkonsole.

AWS Command Line Interface

Interagieren Sie mit AWS Diensten mithilfe von Befehlen in Ihrer Befehlszeilen-Shell. Das AWS Command Line Interface wird unter Windows, MacOS und Linux unterstützt. Weitere Informationen zu finden Sie im [AWS CLI AWS Command Line Interface Benutzerhandbuch](#). Die Push-Befehle für AWS End User Messaging finden Sie in der [AWS CLI Befehlsreferenz](#).

AWS SDKs

Wenn Sie ein Softwareentwickler sind, der es vorzieht, Anwendungen sprachspezifisch zu erstellen, APIs anstatt eine Anfrage über HTTP oder HTTPS einzureichen, AWS bietet dieser Service Bibliotheken, Beispielcode, Tutorials und andere Ressourcen. Diese Bibliotheken bieten grundlegende Funktionen zur Automatisierung von Aufgaben, wie z. B. das kryptografische Signieren Ihrer Anfragen, das Wiederholen von Anfragen und das Behandeln

von Fehlerantworten. Diese Funktionen helfen Ihnen dabei, den Einstieg effizienter zu gestalten. Weitere Informationen finden Sie unter [Tools für AWS](#).

Regionale Verfügbarkeit

AWS End User Messaging Push ist in mehreren AWS-Regionen Ländern in Nordamerika, Europa, Asien und Ozeanien verfügbar. AWS Unterhält in jeder Region mehrere Availability Zones. Diese Availability Zones sind physisch voneinander isoliert, jedoch durch private, hochredundante Netzwerkverbindungen mit geringer Latenz und hohem Durchsatz miteinander verbunden. Diese Availability Zones werden verwendet, um ein sehr hohes Maß an Verfügbarkeit und Redundanz zu gewährleisten und gleichzeitig die Latenz zu minimieren.

Weitere Informationen dazu finden Sie unter [Geben Sie an AWS-Regionen, was AWS-Regionen Ihr Konto verwenden kann](#) in der. Allgemeine Amazon Web Services-Referenz Eine Liste aller Regionen, in denen AWS End User Messaging Push derzeit verfügbar ist, sowie den Endpunkt für jede Region finden Sie unter [Endpunkte und Kontingente](#) für Amazon Pinpoint API 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](#).

Einrichtung eines AWS-Konto

Bevor Sie AWS End User Messaging Push verwenden können, um Push-Benachrichtigungen an Ihre App zu senden, müssen Sie zunächst eine AWS-Konto mit ausreichenden IAM-Berechtigungen einholen. Dies AWS-Konto kann auch für andere Dienste im AWS Ökosystem verwendet werden.

Topics

- [Melde dich an für eine AWS-Konto](#)
- [Erstellen eines Benutzers mit Administratorzugriff](#)

Melde dich an für eine AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/die-Anmeldung>.
2. Folgen Sie den Online-Anweisungen.

Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff erfordern](#).

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu <https://aws.amazon.com/> gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS-Managementkonsole](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter [Benutzerzugriff mit der Standardeinstellung konfigurieren](#).AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal](#).

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

Anweisungen hierzu finden Sie unter [Berechtigungssatz erstellen](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Eine genaue Anleitung finden Sie unter [Gruppen hinzufügen](#) im AWS IAM Identity Center Benutzerhandbuch.

Erste Schritte mit AWS End User Messaging Push

Um AWS End User Messaging Push so einzurichten, dass es Push-Benachrichtigungen an Ihre Apps senden kann, müssen Sie zunächst die Anmeldeinformationen angeben, die AWS End User Messaging Push autorisieren, Nachrichten an Ihre App zu senden. Welche Anmeldeinformationen Sie angeben, hängt davon ab, welches Push-Benachrichtigungssystem Sie verwenden:

- Informationen zu den Anmeldedaten für den Apple Push Notification Service (APN) finden Sie in der Apple Developer-Dokumentation unter [Einen Verschlüsselungsschlüssel und eine Schlüssel-ID von Apple beziehen und Ein Anbieterzertifikat von Apple beziehen](#).
- [Anmeldeinformationen für Firebase Cloud Messaging \(FCM\) können über die Firebase-Konsole abgerufen werden, siehe Firebase Cloud Messaging.](#)
- [Informationen zu Baidu finden Sie unter Baidu.](#)
- Informationen zu Amazon Device Messaging (ADM) -Anmeldeinformationen finden Sie unter [Zugangsdaten abrufen](#).

Anwendung erstellen und Push-Kanäle aktivieren

Bevor Sie AWS End User Messaging Push zum Senden von Push-Benachrichtigungen verwenden können, müssen Sie zunächst eine Anwendung erstellen und den Push-Benachrichtigungskanal aktivieren.

Kontextuell

Anwendung

Eine Anwendung ist ein Speichercontainer für all Ihre Push-Einstellungen für AWS End User Messaging. Die Anwendung speichert auch Ihre Amazon Pinpoint Pinpoint-Kanäle, Kampagnen und Journey-Einstellungen.

Key (Schlüssel)

Ein privater Signaturschlüssel, der von AWS End User Messaging Push verwendet wird, um Authentifizierungstoken kryptografisch zu signieren APNs. Sie erhalten den Signaturschlüssel aus Ihrer Apple-Entwicklerkonto.

Wenn Sie einen Signaturschlüssel angeben, verwendet AWS End User Messaging Push APNs für jede gesendete Push-Benachrichtigung ein Token zur Authentifizierung. Mit Ihrem Signaturschlüssel können Sie Push-Benachrichtigungen an APNs Produktions- und Sandbox-Umgebungen senden.

Anders als Zertifikate läuft Ihr Signaturschlüssel nicht ab. Sie stellen Ihren Schlüssel nur einmal bereit und müssen ihn später nicht verlängern. Sie können denselben Signaturschlüssel für mehrere Apps verwenden. Weitere Informationen finden Sie unter [Kommunizieren mit APNs Authentifizierungstoken](#) in der Xcode-Hilfe.

Zertifikat

Ein TLS-Zertifikat, mit dem sich AWS End User Messaging Push APNs beim Senden von Push-Benachrichtigungen authentifiziert. Ein APNs Zertifikat kann sowohl Produktions- als auch Sandbox-Umgebungen oder nur die Sandbox-Umgebung unterstützen. Sie erhalten das Zertifikat aus Ihrer Apple-Entwicklerkonto.

Ein Zertifikat läuft nach einem Jahr ab. In diesem Fall müssen Sie ein neues Zertifikat erstellen, das Sie dann AWS End User Messaging Push zur Verfügung stellen, um die Zustellung von Push-

Benachrichtigungen zu erneuern. Weitere Informationen finden Sie unter [Kommunizieren APNs mit einem TLS-Zertifikat](#) in der Xcode-Hilfe.

Voraussetzungen

Bevor Sie einen Push-Kanal verwenden können, benötigen Sie gültige Anmeldeinformationen für den Push-Dienst. Weitere Informationen zum Abrufen von Anmeldeinformationen finden Sie unter [Erste Schritte mit AWS End User Messaging Push](#).

Verfahren

Folgen Sie diesen Anweisungen, um eine Anwendung zu erstellen und einen der Push-Kanäle zu aktivieren. Um dieses Verfahren abzuschließen, müssen Sie nur einen Anwendungsnamen eingeben. Sie können jeden der Push-Kanäle zu einem späteren Zeitpunkt aktivieren oder deaktivieren.

1. Öffnen Sie die AWS End User Messaging-Push-Konsole unter <https://console.aws.amazon.com/push-notifications/>.
2. Wählen Sie Create application aus.
3. Geben Sie unter Anwendungsname den Namen für Ihre Anwendung ein.
4. (Optional) Folgen Sie diesem optionalen Schritt, um den Apple Push-Benachrichtigungsdienst zu aktivieren (APNs).
 - a. Wählen Sie für den Apple Push Notification Service (APNs) die Option Aktivieren aus.
 - b. Wählen Sie als Standard-Authentifizierungstyp entweder:
 - i. Wenn Sie Key Credentials wählen, geben Sie die folgenden Informationen aus Ihrem Apple-Entwicklerkonto ein. AWS End User Messaging Push benötigt diese Informationen, um Authentifizierungstoken zu erstellen.
 - Schlüssel-ID – Die Ihrem Signaturschlüssel zugeordnete ID.
 - Bundle-ID – Die Ihrer iOS-App zugeordnete ID.
 - Team-ID – Die Ihrem Apple-Developer-Kontoteam zugewiesene ID.
 - Authentifizierungsschlüssel – Die .p8-Datei, die Sie von Ihrem Apple-Developer-Konto herunterladen, wenn Sie einen Authentifizierungsschlüssel erstellen.
 - ii. Wenn Sie die Option Certificate credentials (Zertifikatanmeldeinformationen) auswählen, geben Sie die folgenden Informationen an:

- SSL certificate (SSL-Zertifikat) – Die .p12-Datei für Ihr TLS-Zertifikat.
 - Zertifikatpasswort – Wenn Sie Ihrem Zertifikat ein Passwort zugewiesen haben, geben Sie es hier ein.
 - Typ des Zertifikats — Wählen Sie den zu verwendenden Zertifikattyp aus.
5. (Optional) Folgen Sie diesem optionalen Schritt, um Firebase Cloud Messaging (FCM) zu aktivieren.
 - a. Wählen Sie für Firebase Cloud Messaging (FCM) die Option Aktivieren aus.
 - b. Wählen Sie für den Standardauthentifizierungstyp entweder:
 - i. Wählen Sie für Token-Anmeldeinformationen (empfohlen) die Option Dateien auswählen und dann Ihre Dienst-JSON-Datei aus.
 - ii. Geben Sie für Schlüsselanmeldedaten Ihren Schlüssel in das Feld API-Schlüssel ein.
 6. (Optional) Folgen Sie diesem optionalen Schritt, um Baidu Cloud Push zu aktivieren.
 - a. Wählen Sie für Baidu Cloud Push die Option Aktivieren aus.
 - b. Geben Sie als API-Schlüssel Ihren API-Schlüssel ein.
 - c. Geben Sie für Secret Key Ihren geheimen Schlüssel ein.
 7. (Optional) Folgen Sie diesem optionalen Schritt, um Amazon Device Messaging zu aktivieren.
 - a. Wählen Sie für Amazon Device Messaging die Option Aktivieren aus.
 - b. Geben Sie als Kunden-ID Ihre Kunden-ID ein.
 - c. Geben Sie unter Kundengeheimnis Ihr Kundengeheimnis ein.
 8. Wählen Sie Create application aus.

Push-Kanäle deaktivieren

Folgen Sie diesen Anweisungen, um einen der Push-Kanäle zu deaktivieren.

1. Öffnen Sie die AWS End User Messaging-Push-Konsole unter <https://console.aws.amazon.com/push-notifications/>.
2. Wählen Sie die Anwendung aus, die Ihre Push-Anmeldeinformationen enthält.
3. (Optional) Deaktivieren Sie für den Apple Push-Benachrichtigungsdienst (APNs) die Option Aktivieren.
4. (Optional) Deaktivieren Sie für Firebase Cloud Messaging (FCM) die Option Aktivieren.
5. (Optional) Deaktivieren Sie für Baidu Cloud Push die Option Aktivieren.
6. (Optional) Deaktivieren Sie für Amazon Device Messaging die Option Aktivieren.
7. Wählen Sie Änderungen speichern.

Senden einer Nachricht

Die AWS End User Messaging Push API kann transaktionale Push-Benachrichtigungen an bestimmte Gerätekennungen senden. Dieser Abschnitt enthält vollständige Codebeispiele, mit denen Sie mithilfe eines SDK Push-Benachrichtigungen über die AWS End User Messaging Push API senden können.

AWS

Sie können diese Beispiele verwenden, um Push-Benachrichtigungen über jeden Push-Benachrichtigungsdienst zu senden, den AWS End User Messaging Push unterstützt. Derzeit unterstützt AWS End User Messaging Push die folgenden Kanäle: Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNs), Baidu Cloud Push und Amazon Device Messaging (ADM).

[Weitere Codebeispiele zu Endpunkten, Segmenten und Kanälen finden Sie unter Codebeispiele.](#)

Note

Wenn Sie Push-Benachrichtigungen über den Firebase Cloud Messaging (FCM) -Dienst senden, verwenden Sie den Dienstenamen GCM in Ihrem Aufruf der AWS End User Messaging Push API. Der Google Cloud Messaging (GCM)-Service wurde von Google am 10. April 2018 eingestellt. Die AWS End User Messaging Push API verwendet jedoch den GCM Dienstenamen für Nachrichten, die sie über den FCM-Dienst sendet, um die Kompatibilität mit dem API-Code aufrechtzuerhalten, der vor der Einstellung des GCM-Dienstes geschrieben wurde.

GCM (AWS CLI)

Das folgende Beispiel verwendet [send-messages](#), um eine GCM-Push-Benachrichtigung mit dem zu senden. AWS CLI *token* Ersetzen Sie es durch das eindeutige Token des Geräts und *611e3e3cdd47474c9c1399a50example* durch Ihre Anwendungs-ID.

```
aws pinpoint send-messages \  
--application-id 611e3e3cdd47474c9c1399a50example \  
--message-request file://myfile.json \  
--region us-west-2
```

Contents of myfile.json:

```
{  
  "Addresses": {
```

```

    "token": {
      "ChannelType" : 'GCM'
    }
  },
  "MessageConfiguration": {
    "GCMMessage": {
      "Action": "URL",
      "Body": "This is a sample message",
      "Priority": "normal",
      "SilentPush": True,
      "Title": "My sample message",
      "TimeToLive": 30,
      "Url": "https://www.example.com"
    }
  }
}

```

Das folgende Beispiel verwendet [send-messages](#), um eine GCM-Push-Benachrichtigung unter Verwendung aller älteren Schlüssel mit dem zu senden. AWS CLI `token` Ersetzen Sie es durch das eindeutige Token des Geräts und `611e3e3cdd47474c9c1399a50example` durch Ihre Anwendungs-ID.

```

aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\"notification\": {\n \"title\": \"string\", \n \"body\":
 \"string\", \n \"android_channel_id\": \"string\", \n \"body_loc_args\": [\n \"string
 \n \n ], \n \"body_loc_key\": \"string\", \n \"click_action\": \"string\", \n \"color\":
 \"string\", \n \"icon\": \"string\", \n \"sound\": \"string\", \n \"tag\": \"string
 \", \n \"title_loc_args\": [\n \"string\" \n ], \n \"title_loc_key\": \"string\" \n },
 \"data\":{ \"message\": \"hello in data\" } }",
      "TimeToLive" : 309744
    }
  },
  "Addresses": {
    "token": {
      "ChannelType": "GCM"
    }
  }
}'

```

```
\ --region us-east-1
```

Das folgende Beispiel verwendet [send-messages](#), um eine GCM-Push-Benachrichtigung mit FCMv1 Nachrichtennutzlast mithilfe von zu senden. AWS CLI `token` Ersetzen Sie es durch das eindeutige Token des Geräts und `611e3e3cdd47474c9c1399a50example` durch Ihre Anwendungs-ID.

```
aws pinpoint send-messages \
--application-id 6a2dafd84bec449ea75fb773f4c41fa1 \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\n \"fcmV1Message\": \n {\n \"message\" :{\n \"notification
\": {\n \"title\": \"string\", \n \"body\": \"string\"\n }, \n \"android\": {\n
\"priority\": \"high\", \n \"notification\": {\n \"title\": \"string\", \n \"body
\": \"string\", \n \"icon\": \"string\", \n \"color\": \"string\", \n \"sound\":
\"string\", \n \"tag\": \"string\", \n \"click_action\": \"string\", \n \"body_loc_key
\": \"string\", \n \"body_loc_args\": [\n \"string\"\n ], \n \"title_loc_key
\": \"string\", \n \"title_loc_args\": [\n \"string\"\n ], \n \"channel_id\":
\"string\", \n \"ticker\": \"string\", \n \"sticky\": true, \n \"event_time\":
\"2024-02-06T22:11:55Z\", \n \"local_only\": true, \n \"notification_priority\":
\"PRIORITY_UNSPECIFIED\", \n \"default_sound\": false, \n \"default_vibrate_timings
\": true, \n \"default_light_settings\": false, \n \"vibrate_timings\": [\n \"22s
\"\n ], \n \"visibility\": \"VISIBILITY_UNSPECIFIED\", \n \"notification_count\": 5,
\n \"light_settings\": {\n \"color\": {\n \"red\": 1, \n \"green\": 2, \n \"blue\":
3, \n \"alpha\": 6\n }, \n \"light_on_duration\": \"112s\", \n \"light_off_duration
\": \"1123s\"\n }, \n \"image\": \"string\"\n }, \n \"data\": {\n \"dataKey1\":
\"priority message\", \n \"data_key_3\": \"priority message\", \n \"dataKey2\":
\"priority message\", \n \"data_key_5\": \"priority message\"\n }, \n \"ttl\":
\"10023.32s\"\n }, \n \"apns\": {\n \"payload\": {\n \"aps\": {\n \"alert\": {\n
\"subtitle\": \"string\", \n \"title-loc-args\": [\n \"string\"\n ], \n \"title-loc-
key\": \"string\", \n \"launch-image\": \"string\", \n \"subtitle-loc-key\": \"string
\", \n \"subtitle-loc-args\": [\n \"string\"\n ], \n \"loc-args\": [\n \"string
\"\n ], \n \"loc-key\": \"string\", \n \"title\": \"string\", \n \"body\": \"string
\"\n }, \n \"thread-id\": \"string\", \n \"category\": \"string\", \n \"content-
available\": 1, \n \"mutable-content\": 1, \n \"target-content-id\": \"string\", \n
\"interruption-level\": \"string\", \n \"relevance-score\": 25, \n \"filter-criteria
\": \"string\", \n \"stale-date\": 6483, \n \"content-state\": {}, \n \"timestamp\":
673634, \n \"dismissal-date\": 4, \n \"attributes-type\": \"string\", \n \"attributes
\": {}, \n \"sound\": \"string\", \n \"badge\": 5\n } \n } \n }, \n \"webpush\": {\n
\"notification\": {\n \"permission\": \"granted\", \n \"maxActions\": 2, \n \"actions
\": [\n \"title\"\n ], \n \"badge\": \"URL\", \n \"body\": \"Hello\", \n \"data\": {\n
```

```

\"hello\": \"hey\"\\n },\\n \\\"dir\\\": \\\"auto\\\",\\n \\\"icon\\\": \\\"icon\\\",\\n \\\"image\\\":
\\\"image\\\",\\n \\\"lang\\\": \\\"string\\\",\\n \\\"renotify\\\": false,\\n \\\"requireInteraction\\\":
true,\\n \\\"silent\\\": false,\\n \\\"tag\\\": \\\"tag\\\",\\n \\\"timestamp\\\": 1707259524964,\\n
\\\"title\\\": \\\"hello\\\",\\n \\\"vibrate\\\": [\\n 100,\\n 200,\\n 300\\n ]\\n },\\n \\\"data\\\": {\\n
\\\"data1\\\": \\\"priority message\\\",\\n \\\"data2\\\": \\\"priority message\\\",\\n \\\"data12\\\":
\\\"priority message\\\",\\n \\\"data3\\\": \\\"priority message\\\"\\n }\\n },\\n \\\"data\\\": {\\n
\\\"data7\\\": \\\"priority message\\\",\\n \\\"data5\\\": \\\"priority message\\\",\\n \\\"data8\\\":
\\\"priority message\\\",\\n \\\"data9\\\": \\\"priority message\\\"\\n }\\n }\\n \\n}\\n}\\n\",
  \"TimeToLive\" : 309744
}
},
\"Addresses\": {
  token: {
    \"ChannelType\": \"GCM\"
  }
}
}'
\\ --region us-east-1

```

Wenn Sie `ImageUrl` das Feld für GCM verwenden, sendet Pinpoint das Feld als Datenbenachrichtigung mit dem Schlüssel `pinpoint.notification.imageUrl`, wodurch verhindert werden kann, dass das Bild sofort wiedergegeben wird. Bitte verwenden `RawContent` oder fügen Sie die Handhabung der Datenschlüssel hinzu, z. B. die Integration Ihrer App mit AWS Amplify

Safari (AWS CLI)

Sie können AWS End User Messaging Push verwenden, um Nachrichten an macOS-Computer zu senden, die den Safari-Webbrowser von Apple verwenden. Um eine Nachricht an den Safari-Browser zu senden, müssen Sie den Inhalt der unformatierten Nachricht angeben und ein bestimmtes Attribut in die Nachrichtennutzlast aufnehmen. Sie können dies tun, indem Sie [eine Vorlage für Push-Benachrichtigungen mit einer Nutzlast für Rohnachrichten erstellen](#) oder den Inhalt der Rohnachricht direkt in einer [Kampagnennachricht](#) im Amazon Pinpoint Benutzerhandbuch angeben.

Note

Dieses spezielle Attribut ist für das Senden an macOS-Laptop- und -Desktop-Computer erforderlich, die den Safari-Webbrowser verwenden. Es ist nicht erforderlich für den Versand an iOS-Geräte wie iPhones und iPads.

Um eine Nachricht an Safari-Webbrowser zu senden, müssen Sie die Nutzlast für unformatierte Nachrichten angeben. Die Nutzlast der unformatierten Nachricht muss ein `url-args`-Array innerhalb des `aps`-Objekts enthalten. Das `url-args`-Array ist erforderlich, um Push-Benachrichtigungen an den Safari-Webbrowser zu senden. Das Array kann jedoch auch ein einzelnes, leeres Element enthalten.

Das folgende Beispiel verwendet [send-messages](#), um eine Benachrichtigung an den Safari-Webbrowser mit dem zu senden. AWS CLI `token` Ersetzen Sie es durch das eindeutige Token des Geräts und `611e3e3cdd47474c9c1399a50example` durch Ihre Anwendungs-ID.

```
aws pinpoint send-messages \  
--application-id 611e3e3cdd47474c9c1399a50example \  
--message-request \  
'{  
  "Addresses": {  
    "token":  
    {  
      "ChannelType": "APNS"  
    }  
  },  
  "MessageConfiguration": {  
    "APNSMessage": {  
      "RawContent":  
        "{\n\"aps\": {\n\"alert\": { \n\"title\": \n\"Title of my message\", \n\"body\":  
\"This is a push notification for the Safari web browser.\n\"},\n\"content-available\":  
1,\n\"url-args\": [\n\"\"]}}"  
    }  
  }  
}'  
\ --region us-east-1
```

Weitere Informationen zu Safari-Push-Benachrichtigungen finden Sie unter [Konfiguration von Safari-Push-Benachrichtigungen](#) auf der Apple-Developer-Website.

APNS (AWS CLI)

Das folgende Beispiel verwendet [send-messages](#), um eine APNS-Push-Benachrichtigung mit dem zu senden. AWS CLI `token` Ersetzen Sie es durch das eindeutige Token des Geräts, `611e3e3cdd47474c9c1399a50example` durch Ihre Anwendungs-ID und `GAME_INVITATION` durch eine eindeutige Kennung.

```
aws pinpoint send-messages \  

```

```
--application-id 611e3e3cdd47474c9c1399a50example \  
--message-request  
{  
  "Addresses": {  
    "token":  
    {  
      "ChannelType":"APNS"  
    }  
  },  
  "MessageConfiguration": {  
    "APNSMessage": {  
      "RawContent": "{\"aps\" : {\"alert\" : {\"title\" : \"Game Request\",  
\"subtitle\" : \"Five Card Draw\", \"body\" : \"Bob wants to play poker\"}, \"category  
\" : \"GAME_INVITATION\"}, \"gameID\" : \"12345678\"}"  
    }  
  }  
}  
\  
--region us-east-1
```

JavaScript (Node.js)

Verwenden Sie dieses Beispiel, um Push-Benachrichtigungen 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.

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 [Einrichten von Anmeldeinformationen](#) im AWS SDK für JavaScript im Node.js Developer Guide.

```
'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 API is available  
const region = 'us-east-1';  
  
// The title that appears at the top of the push notification.  
var title = 'Test message sent from End User Messaging Push.';  
  
// The content of the push notification.
```

```
var message = 'This is a sample message sent from End User Messaging Push by using
the '
    + 'AWS SDK for JavaScript in Node.js';

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

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

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

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

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

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

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

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

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

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

return messageRequest
}

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

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

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

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

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

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

SendMessage()
```

Python

Verwenden Sie dieses Beispiel, um Push-Benachrichtigungen mithilfe von AWS SDK für 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 API is available
region = "us-east-1"

# The title that appears at the top of the push notification.
title = "Test message sent from End User Messaging Push."

# The content of the push notification.
message = ("This is a sample message sent from End User Messaging Push by using the
"
           "AWS SDK für Python (Boto3).")

# The 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()
```

Weitere Ressourcen

- Weitere Informationen zu Vorlagen für Push-Kanäle finden Sie unter Vorlagen für [Push-Benachrichtigungen erstellen](#) im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

Empfangen von Push-Benachrichtigungen in Ihrer Anwendung

In den folgenden Themen wird beschrieben, wie Sie Ihre Swift-, Android-, React Native- oder Flutter-App so ändern, dass sie Push-Benachrichtigungen empfängt.

Themen

- [Einrichten von Swift-Push-Benachrichtigungen](#)
- [Einrichten von Android-Push-Benachrichtigungen](#)
- [Einrichten von Flutter-Push-Benachrichtigungen](#)
- [Einrichten von React-Native-Push-Benachrichtigungen](#)
- [Erstellen Sie eine Anwendung in AWS End User Messaging Push](#)
- [Umgang mit Push-Benachrichtigungen](#)

Einrichten von Swift-Push-Benachrichtigungen

Push-Benachrichtigungen für iOS-Apps werden über den 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 [Einrichtung von Push-Benachrichtigungsdiensten](#) in der AWS Amplify-Dokumentation.

Mit Tokens arbeiten APNs

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 jetzt ungültig ist. Wenn Sie versuchen, die Benachrichtigung zu senden, erhalten Sie eine Benachrichtigung 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 AWS Amplify-Dokumentation.

Einrichten von Flutter-Push-Benachrichtigungen

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

Erstellen Sie eine Anwendung in AWS End User Messaging Push

Um mit dem Senden von Push-Benachrichtigungen in AWS End User Messaging Push zu beginnen, müssen Sie eine Anwendung erstellen. Als Nächstes müssen Sie die Push-Benachrichtigungskanäle aktivieren, die Sie verwenden möchten, indem Sie die entsprechenden Anmeldeinformationen angeben.

Mithilfe der AWS End User Messaging Push-Konsole können Sie neue Anwendungen erstellen und Kanäle für Push-Benachrichtigungen einrichten. Weitere Informationen finden Sie unter [Anwendung erstellen und Push-Kanäle aktivieren](#).

Sie können Anwendungen auch mithilfe der [API](#), eines [AWS SDK](#) oder der [AWS Command Line Interface](#) (AWS CLI) erstellen und einrichten. Verwenden Sie die Apps Ressource, um eine Anwendung zu erstellen. 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-Dienstes.
- [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 Anmeldeinformationen erhalten haben, die zum Senden von Push-Benachrichtigungen erforderlich sind, können Sie Ihre Anwendung so aktualisieren, dass sie Push-Benachrichtigungen empfangen kann. Weitere Informationen finden Sie in der [Dokumentation unter Push-Benachrichtigungen — Erste Schritte](#). AWS Amplify

Löschen einer Anwendung

Durch dieses Verfahren wird die Anwendung aus Ihrem Konto und allen Ressourcen in der Anwendung entfernt.

Kontextuell

Anwendung

Eine Anwendung ist ein Speichercontainer für all Ihre Push-Einstellungen für AWS End User Messaging. Die Anwendung speichert auch Ihre Amazon Pinpoint Pinpoint-Kanäle, Kampagnen und Journey-Einstellungen.

Verfahren

1. Öffnen Sie die Push-Konsole für AWS Endbenutzer-Messaging unter <https://console.aws.amazon.com/push-notifications/>.
2. Wählen Sie eine Anwendung aus und klicken Sie dann auf Löschen.
3. Geben Sie im Fenster Anwendung löschen den Text ein **delete** und wählen Sie dann Löschen.

Important

Alle Amazon Pinpoint Pinpoint-Kanäle, Kampagnen, Journeys oder -Segmente werden ebenfalls gelöscht.

Best Practices

Selbst wenn Sie die Interessen Ihrer Kunden im Auge behalten, können Sie immer noch auf Situationen stoßen, die sich auf die Zustellbarkeit Ihrer Nachrichten auswirken. Die folgenden Abschnitte enthalten Empfehlungen, mit denen Sie sicherstellen können, dass Ihre E-Mail-Kommunikation Ihre Zielgruppe erreicht.

Senden einer großen Anzahl von Push-Benachrichtigungen

Bevor Sie eine große Menge an Push-Benachrichtigungen versenden, stellen Sie sicher, dass Ihr Konto so konfiguriert ist, dass es Ihre Durchsatzanforderungen erfüllt. Standardmäßig sind alle Konten so konfiguriert, dass sie 25.000 Nachrichten pro Sekunde senden. Wenn Sie mehr als 25 000 Nachrichten in einer Sekunde versenden müssen, können Sie eine Kontingenterhöhung anfordern. Weitere Informationen finden Sie unter [Kontingente für AWS Endbenutzer-Messaging-Push](#).

Stellen Sie sicher, dass Ihr Konto mit den Anmeldeinformationen für jeden Anbieter von Push-Benachrichtigungen, den Sie verwenden möchten, korrekt konfiguriert ist, z. B. FCM oder APNs.

Überlegen Sie sich abschließend, wie Sie mit Ausnahmen umgehen möchten. Jeder Push-Benachrichtigungsdienst bietet unterschiedliche Ausnahmemeldungen. Bei transaktionalen Sendungen erhalten Sie den Hauptstatuscode 200 für den API-Aufruf und den Statuscode 400 pro Endpunkt für permanente Ausfälle, falls das entsprechende Plattform-Token (z. B. FCM) oder Zertifikat (z. B. APN) beim Senden von Nachrichten als ungültig eingestuft wird.

Sicherheit bei Push-Nachrichten für AWS Endbenutzer

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. 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 AWS End User Messaging Push gelten, finden Sie unter [AWS Services im Umfang nach Compliance-Programm AWS](#) .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, 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 hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von AWS End User Messaging Push anwenden können. In den folgenden Themen erfahren Sie, wie Sie AWS End User Messaging Push konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste verwenden können, die Ihnen helfen, Ihre AWS End User Messaging Push-Ressourcen zu überwachen und zu schützen.

Topics

- [Datenschutz bei AWS End User Messaging Push](#)
- [Identitäts- und Zugriffsmanagement für AWS End User Messaging Push](#)
- [Konformitätsprüfung für AWS End User Messaging Push](#)
- [Ausfallsicherheit bei Push-Nachrichten für AWS Endbenutzer](#)
- [Infrastruktursicherheit bei AWS End User Messaging Push](#)
- [Konfigurations- und Schwachstellenanalyse](#)
- [Bewährte Methoden für die Gewährleistung der Sicherheit](#)

Datenschutz bei AWS End User Messaging Push

Das AWS [Modell](#) der gilt für den Datenschutz in AWS End User Messaging Push. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. 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).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- 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 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

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 AWS End User Messaging Push oder anderen Programmen AWS-Services über die Konsole, API oder arbeiten. AWS CLI AWS SDKs 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.

Datenverschlüsselung

AWS Push-Daten von End User Messaging werden bei der Übertragung und im Ruhezustand verschlüsselt. Wenn Sie Daten an AWS End User Messaging Push senden, werden die Daten beim Empfang und bei der Speicherung verschlüsselt. Wenn Sie Daten von AWS End User Messaging Push abrufen, werden die Daten mithilfe aktueller Sicherheitsprotokolle an Sie übertragen.

Verschlüsselung im Ruhezustand

AWS End User Messaging Push verschlüsselt alle Daten, die es für Sie speichert. Dazu gehören Konfigurationsdaten, Benutzer- und Endpunktdaten, Analysedaten und alle Daten, die Sie in AWS End User Messaging Push hinzufügen oder importieren. Um Ihre Daten zu verschlüsseln, verwendet AWS End User Messaging Push interne AWS Key Management Service (AWS KMS) Schlüssel, die der Dienst besitzt und in Ihrem Namen verwaltet. Diese Schlüssel werden regelmäßig rotiert. Informationen dazu AWS KMS finden Sie im [AWS Key Management Service Entwicklerhandbuch](#).

Verschlüsselung während der Übertragung

AWS End User Messaging Push verwendet HTTPS und Transport Layer Security (TLS) 1.2 oder höher für die Kommunikation mit Ihren Clients und Anwendungen. Für die Kommunikation mit anderen AWS Diensten verwendet AWS End User Messaging Push HTTPS und TLS 1.2. Darüber hinaus ist die gesamte Kommunikation mit HTTPS und TLS 1.2 gesichert, wenn Sie AWS End User Messaging Push-Ressourcen mithilfe der Konsole AWS Command Line Interface, eines AWS SDK oder des erstellen und verwalten.

Schlüsselverwaltung

Um Ihre AWS End User Messaging-Push-Daten zu verschlüsseln, verwendet AWS End User Messaging Push interne AWS KMS Schlüssel, die dem Dienst 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 AWS End User Messaging Push speichern.

Datenschutz für den Datenverkehr zwischen Netzwerken

Datenschutz im Netzwerkverkehr bezieht sich auf die Sicherung von Verbindungen und Datenverkehr zwischen AWS End User Messaging Push und Ihren lokalen Clients und Anwendungen sowie zwischen AWS End User Messaging Push und anderen AWS Ressourcen in derselben Region. AWS Die folgenden Funktionen und Verfahren können Ihnen dabei helfen, den Schutz des Netzwerkverkehrs für AWS End User Messaging Push zu gewährleisten.

Datenverkehr zwischen AWS End User Messaging Push und lokalen Clients und Anwendungen

Um eine private Verbindung zwischen AWS End User Messaging Push und Clients und Anwendungen in Ihrem lokalen Netzwerk herzustellen, können Sie verwenden. Direct Connect 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 Direct Connect Router verbunden. Weitere Informationen finden Sie unter [Was ist Direct Connect?](#) im Direct Connect -Benutzerhandbuch.

Um den Zugriff auf AWS End User Messaging Push über veröffentlichte Versionen zu sichern APIs, empfehlen wir, dass Sie die Push-Anforderungen für AWS Endbenutzer-Nachrichtennachrichten für API-Aufrufe einhalten. AWS Für End User Messaging Push müssen die Clients Transport Layer Security (TLS) 1.2 oder höher verwenden. 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 AWS End User Messaging Push und anderen Ressourcen AWS

Um die Kommunikation zwischen AWS End User Messaging Push und anderen AWS Ressourcen in derselben AWS Region zu sichern, verwendet AWS End User Messaging Push standardmäßig HTTPS und TLS 1.2.

Identitäts- und Zugriffsmanagement für AWS End User Messaging Push

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um AWS End User Messaging Push-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So funktioniert AWS End User Messaging Push mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push](#)
- [Fehlerbehebung bei der Push-Identität und dem Zugriff für AWS Endbenutzer-Nachrichten](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe [Fehlerbehebung bei der Push-Identität und dem Zugriff für AWS Endbenutzer-Nachrichten](#)).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe [So funktioniert AWS End User Messaging Push mit IAM](#)).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe [Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push](#)).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter [AWS Signature Version 4 for API requests](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Tasks that require root user credentials](#) im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können](#).

Eine [IAM-Gruppe](#) spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#) AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter [Methoden, um eine Rolle zu übernehmen](#) im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) 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 definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcen und unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter [Choose between managed policies and inline policies](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen 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. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#).

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im -IAM-Benutzerhandbuch.
- **Richtlinien zur Dienstkontrolle (SCPs)** — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter [Service-Kontrollrichtlinien](#) im AWS Organizations -Benutzerhandbuch.
- **Richtlinien zur Ressourcenkontrolle (RCPs)** — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter [Richtlinien zur Ressourcenkontrolle \(RCPs\)](#).
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine

Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM-Benutzerhandbuch.

So funktioniert AWS End User Messaging Push mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf AWS End User Messaging Push zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen für die Verwendung mit AWS End User Messaging Push verfügbar sind.

IAM-Funktionen, die Sie mit AWS End User Messaging Push verwenden können

IAM-Feature	AWS Push-Unterstützung für Endbenutzer-Nachrichten
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Ja
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie AWS End User Messaging Push und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für End User Messaging Push AWS

Unterstützt Richtlinien auf Identitätsbasis: Ja

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 [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

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. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push

Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push](#)

Ressourcenbasierte Richtlinien innerhalb AWS von End User Messaging Push

Unterstützt ressourcenbasierte Richtlinien: Ja

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

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Richtlinienaktionen für AWS End User Messaging Push

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der AWS Endbenutzer-Nachrichten-Push-Aktionen finden Sie unter [Durch AWS End User Messaging Push definierte Aktionen](#) in der Serviceautorisierungsreferenz.

Bei Richtlinienaktionen in AWS End User Messaging Push wird vor der Aktion das folgende Präfix verwendet:

```
mobiletargeting
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "mobiletargeting:action1",  
  "mobiletargeting:action2"  
]
```

Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push](#)

Richtlinienressourcen für AWS End User Messaging Push

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen](#)

[\(ARN\)](#) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der Ressourcentypen und der zugehörigen Typen von AWS End User Messaging Push finden Sie unter [Ressourcen ARNs, die durch AWS End User Messaging Push definiert](#) sind in der Referenz zur Serviceautorisierung. Informationen zu den Aktionen, mit denen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von AWS End User Messaging Push definierte Aktionen](#).

Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push](#)

Bedingungsschlüssel für Richtlinien für AWS End User Messaging Push

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. 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. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der Push-Bedingungsschlüssel für AWS End User Messaging finden Sie unter [Bedingungsschlüssel für AWS End User Messaging Push](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von AWS End User Messaging Push definierte Aktionen](#).

Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push](#)

ACLs unter Push für AWS Endbenutzer-Nachrichten

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit AWS Endbenutzer-Nachrichten-Push

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Definieren von Berechtigungen mit ABAC-Autorisierung](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit AWS End User Messaging Push

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie den Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipalberechtigungen für AWS End User Messaging Push

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu

stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für AWS End User Messaging Push

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, 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.

Warning

Das Ändern der Berechtigungen für eine Servicerolle kann dazu führen, dass die Push-Funktionalität von AWS End User Messaging beeinträchtigt wird. Bearbeiten Sie Servicerollen nur, wenn AWS End User Messaging Push Sie dazu anleitet.

Mit Diensten verknüpfte Rollen für AWS End User Messaging Push

Unterstützt serviceverknüpfte Rollen: Ja

Eine serviceverknüpfte 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. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS End User Messaging Push

Standardmäßig sind Benutzer und Rollen nicht berechtigt, AWS End User Messaging Push-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien \(Konsole\)](#) im IAM-Benutzerhandbuch.

Einzelheiten zu den von AWS End User Messaging Push definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für AWS End User Messaging Push](#) in der Referenz zur Serviceautorisierung.

Themen

- [Best Practices für Richtlinien](#)
- [Verwenden der AWS End User Messaging-Push-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand in Ihrem Konto AWS Endbenutzer-Nachrichten-Push-Ressourcen erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien 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 [Von AWS verwaltete Richtlinien](#) oder [Von AWS verwaltete Richtlinien für Auftragsfunktionen](#) 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. 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 [Richtlinienvvalidierung mit 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 [Sicherer API-Zugriff mit MFA](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Best Practices für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der AWS End User Messaging-Push-Konsole

Um auf die AWS End User Messaging-Push-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS End User Messaging-Push-Ressourcen in Ihrem aufzulisten und einzusehen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

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.

Um sicherzustellen, dass Benutzer und Rollen die AWS End User Messaging Push-Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die `AWSEndUserMessaging` AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSEndUserMessaging",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:CreateApp",
        "mobiletargeting:GetApp",
        "mobiletargeting:GetApps",
        "mobiletargeting>DeleteApp",
        "mobiletargeting:GetChannels",
        "mobiletargeting:GetApnsChannel",
        "mobiletargeting:GetApnsVoipChannel",
        "mobiletargeting:GetApnsVoipSandboxChannel",
        "mobiletargeting:GetApnsSandboxChannel",
        "mobiletargeting:GetAdmChannel",
        "mobiletargeting:GetBaiduChannel",
        "mobiletargeting:GetGcmChannel",
        "mobiletargeting:UpdateApnsChannel",
        "mobiletargeting:UpdateApnsVoipChannel",
        "mobiletargeting:UpdateApnsVoipSandboxChannel",
        "mobiletargeting:UpdateBaiduChannel",
        "mobiletargeting:UpdateGcmChannel",
        "mobiletargeting:UpdateAdmChannel"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

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 umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Fehlerbehebung bei der Push-Identität und dem Zugriff für AWS Endbenutzer-Nachrichten

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS End User Messaging Push und IAM auftreten können.

Themen

- [Ich bin nicht berechtigt, eine Aktion in AWS End User Messaging Push durchzuführen](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS End User Messaging Push-Ressourcen ermöglichen](#)

Ich bin nicht berechtigt, eine Aktion in AWS End User Messaging Push durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer mateojackson versucht, über die Konsole Details zu einer fiktiven *my-example-widget*-Ressource anzuzeigen, jedoch nicht über mobiletargeting:*GetWidget*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
mobiletargeting:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer mateojackson aktualisiert werden, damit er mit der mobiletargeting:*GetWidget*-Aktion auf die *my-example-widget*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Durchführung der iam:PassRole Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an AWS End User Messaging Push übergeben 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 AWS End User Messaging Push 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 von mir den Zugriff AWS-Konto auf meine AWS End User Messaging Push-Ressourcen ermöglichen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob AWS End User Messaging Push diese Funktionen unterstützt, finden Sie unter [So funktioniert AWS End User Messaging Push 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 für 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 [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Konformitätsprüfung für AWS End User Messaging Push

Informationen darüber, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der [AWS Sicherheitsdokumentation](#).

Ausfallsicherheit bei Push-Nachrichten für AWS Endbenutzer

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen 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 Availability Zones AWS-Regionen und Availability Zones finden Sie unter [AWS Globale](#) Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur bietet AWS End User Messaging Push mehrere Funktionen, die Sie bei der Erfüllung Ihrer Datenausfallsicherheit und Ihrer Backup-Anforderungen unterstützen.

Infrastruktursicherheit bei AWS End User Messaging Push

Als verwalteter Service ist AWS End User Messaging Push durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper [Amazon Web Services: Security Processes im Überblick](#) beschrieben sind.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf AWS End User Messaging Push zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Clients müssen außerdem Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen 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.

Konfigurations- und Schwachstellenanalyse

Als verwalteter Service ist AWS End User Messaging Push durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper [Amazon Web Services: Sicherheitsprozesse im Überblick](#) beschrieben sind. Das bedeutet, dass grundlegende Sicherheitsaufgaben und -verfahren AWS verwaltet und ausgeführt werden, um die zugrunde liegende Infrastruktur für Ihr Konto und Ihre Ressourcen zu sichern, zu patchen, zu aktualisieren und anderweitig zu warten. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert.

Bewährte Methoden für die Gewährleistung der Sicherheit

Verwenden Sie AWS Identity and Access Management (IAM) -Konten, um den Zugriff auf API-Operationen zu steuern, insbesondere auf Operationen, die Ressourcen erstellen, ändern oder löschen. Für die -API umfassen diese Ressourcen Projekte, Kampagnen und Reisen.

- Erstellen Sie einen individuellen Benutzer für jede Person, die -Ressourcen verwaltet, einschließlich Sie selbst. Verwenden Sie keine AWS Root-Anmeldeinformationen, um 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 zu Sicherheit finden Sie unter [Sicherheit bei Push-Nachrichten für AWS Endbenutzer](#). 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](#).

Überwachung von Push-Nachrichten für AWS Endbenutzer

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS End User Messaging Push und Ihren anderen AWS-Lösungen. AWS bietet die folgenden Überwachungstools, um den AWS Endbenutzer-Nachrichtenversand zu beobachten, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen Ihrer EC2 Amazon-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldateien von EC2 Amazon-Instances und anderen Quellen überwachen CloudTrail, speichern und darauf zugreifen. CloudWatch Logs kann Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).
- Amazon EventBridge kann verwendet werden, um Ihre AWS Services zu automatisieren und automatisch auf Systemereignisse wie Probleme mit der Anwendungsverfügbarkeit oder Ressourcenänderungen zu reagieren. Ereignisse im AWS Rahmen von Services werden nahezu EventBridge in Echtzeit zugestellt. Sie können einfache Regeln schreiben, um anzugeben, welche Ereignisse für Sie interessant sind und welche automatisierten Aktionen ausgeführt werden sollen, wenn ein Ereignis mit einer Regel übereinstimmt. Weitere Informationen finden Sie im [EventBridge Amazon-Benutzerhandbuch](#).
- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS Kontos getätigt wurden, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Aufrufe getätigt wurden und wann die Aufrufe erfolgten. Weitere Informationen finden Sie im [AWS CloudTrail -Benutzerhandbuch](#).

Überwachung von Push-Nachrichten für AWS Endbenutzer mit Amazon CloudWatch

Sie können den AWS Endbenutzer-Nachrichtendienst mithilfe von CloudWatch Push überwachen. Dabei werden Rohdaten gesammelt und zu lesbaren Metriken verarbeitet, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsdaten zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

Eine Liste der Metriken und Dimensionen finden Sie unter [Monitoring Amazon Pinpoint with CloudWatch](#) im Amazon Pinpoint Benutzerhandbuch.

Protokollieren von Push-API-Aufrufen für AWS Endbenutzer-Nachrichten mithilfe von AWS CloudTrail

AWS End User Messaging Push ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen eines Benutzers, einer Rolle oder eines AWS Dienstes in AWS End User Messaging Push bereitstellt. CloudTrail erfasst alle API-Aufrufe für AWS End User Messaging Push als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS End User Messaging Push-Konsole und Code-Aufrufe der AWS End User Messaging Push API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS End User Messaging Push. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf einsehen. Anhand der von CloudTrail gesammelten Informationen können Sie die Anfrage an AWS End User Messaging Push, 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 CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

AWS Nachrichtenübermittlung an Endbenutzer: Geben Sie Informationen ein CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto, wenn Sie das Konto erstellen. Wenn in AWS End User Messaging Push eine Aktivität auftritt, wird diese Aktivität zusammen mit anderen CloudTrail

AWS Serviceereignissen im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen AWS-Konto. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem System AWS-Konto, einschließlich der Ereignisse für AWS End User Messaging Push, 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 anlegen, gilt dieser für alle AWS-Regionen-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](#)

Alle Push-Aktionen von AWS End User Messaging werden von der [AWS End User Messaging Push API-Referenz](#) protokolliert CloudTrail und sind in dieser Dokumentation dokumentiert. Beispielsweise generieren Aufrufe von UpdateApnsChannel und GetApnsVoipChannel Aktionen Einträge in den CloudTrail Protokolldateien. GetAdmChannel

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management (IAM-) 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](#).

Grundlegendes zu den Einträgen in der Push-Protokolldatei von AWS End User Messaging

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 und enthält 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.

Greifen Sie über einen Schnittstellenendpunkt auf AWS End User Messaging Push zu (AWS PrivateLink)

Sie können AWS PrivateLink verwenden, um eine private Verbindung zwischen Ihrer VPC und AWS End User Messaging Push herzustellen. Sie können auf AWS End User Messaging Push zugreifen, als ob es in Ihrer VPC wäre, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder Direct Connect eine Verbindung zu verwenden. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um auf AWS End User Messaging Push zuzugreifen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellen-Endpunkt erstellen, der von AWS PrivateLink unterstützt wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Dabei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Einstiegspunkt für den Datenverkehr dienen, der für AWS End User Messaging Push bestimmt ist.

Weitere Informationen finden Sie AWS PrivateLink im Handbuch unter [Access AWS-Services through AWS PrivateLink](#).

Überlegungen zu AWS End User Messaging Push

Bevor Sie einen Schnittstellenendpunkt für AWS End User Messaging Push einrichten, lesen Sie die [Überlegungen](#) im AWS PrivateLink Handbuch durch.

AWS End User Messaging Push unterstützt Aufrufe aller API-Aktionen über den Schnittstellenendpunkt.

VPC-Endpunktrichtlinien werden für AWS End User Messaging Push nicht unterstützt. Standardmäßig ist der vollständige Zugriff auf AWS End User Messaging Push über den Schnittstellenendpunkt zulässig. Alternativ können Sie den Endpunkt-Netzwerkschnittstellen eine Sicherheitsgruppe zuordnen, um den Datenverkehr zu AWS End User Messaging Push über den Schnittstellenendpunkt zu steuern.

Erstellen Sie einen Schnittstellenendpunkt für AWS End User Messaging Push

Sie können einen Schnittstellenendpunkt für AWS End User Messaging Push entweder mit der Amazon VPC-Konsole oder mit AWS Command Line Interface (AWS CLI) erstellen. Weitere

Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im AWS PrivateLink - Leitfaden.

Erstellen Sie einen Schnittstellenendpunkt für AWS End User Messaging Push mit dem folgenden Servicenamen:

```
com.amazonaws.region.pinpoint
```

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen an AWS End User Messaging Push unter Verwendung des standardmäßigen regionalen DNS-Namens stellen. Beispiel, `com.amazonaws.us-east-1.pinpoint`.

Erstellen einer Endpunktrichtlinie für Ihren Schnittstellen-Endpunkt

Eine Endpunktrichtlinie ist eine IAM-Ressource, die Sie an einen Schnittstellen-Endpunkt anfügen können. Die standardmäßige Endpunktrichtlinie ermöglicht den vollen Zugriff auf AWS End User Messaging Push über den Schnittstellenendpunkt. Um den Zugriff auf AWS End User Messaging Push von Ihrer VPC aus zu kontrollieren, fügen Sie dem Schnittstellenendpunkt eine benutzerdefinierte Endpunktrichtlinie hinzu.

Eine Endpunktrichtlinie gibt die folgenden Informationen an:

- Die Prinzipale, die Aktionen ausführen können (AWS-Konten, IAM-Benutzer und IAM-Rollen).
- Aktionen, die ausgeführt werden können
- Die Ressourcen, auf denen die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Services mit Endpunktrichtlinien](#) im AWS PrivateLink -Leitfaden.

Beispiel: VPC-Endpunktrichtlinie für AWS Endbenutzer-Messaging-Push-Aktionen

Im Folgenden finden Sie ein Beispiel für eine benutzerdefinierte Endpunktrichtlinie. Wenn Sie diese Richtlinie an Ihren Schnittstellenendpunkt anhängen, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten AWS Endbenutzer-Messaging-Push-Aktionen.

```
{
  "Statement": [
    {
```

```
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
        "mobiletargeting:CreateApp",
        "mobiletargeting>DeleteApp"
    ],
    "Resource": "*"
  }
]
```

Kontingente für AWS Endbenutzer-Messaging-Push

Ihr AWS-Konto hat Standardkontingente, früher als Limits bezeichnet, für jeden AWS Dienst. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region. Sie können Erhöhungen für einige Kontingente beantragen und andere Kontingente können nicht erhöht werden.

Um die Kontingente für AWS End User Messaging Push anzuzeigen, öffnen Sie die [Konsole Service Quotas](#). Wählen Sie im Navigationsbereich AWS-Services und dann Amazon Pinpoint aus.

Ihr AWS-Konto hat die folgenden Kontingente für AWS End User Messaging Push.

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, verwenden Sie die Service Quotas Quotas-Konsole
Amazon Device Messaging (ADM)-Nachrichte – Nutzlastgröße	6 KB pro Nachricht	Nein
Größe der Nutzlast für Nachrichten des Apple-Push-Benachrichtigungsdienstes (APNs)	4 KB pro Nachricht	Nein
APNs Größe der Nutzlast von Sandbox-Nachrichten	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

Dokumentenverlauf für das AWS End User Messaging Push User Guide

In der folgenden Tabelle werden die Dokumentationsversionen für AWS End User Messaging Push beschrieben.

Änderung	Beschreibung	Datum
Erstversion	Erste Version des AWS Endbenutzerhandbuchs für Messaging-Push	24. Juli 2024

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.