



Benutzerhandbuch

AWS Messaging-Push für Endbenutzer



AWS Messaging-Push für Endbenutzer: Benutzerhandbuch

Copyright © 2024 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
Zugriff auf Push für AWS Endbenutzer	2
Regionale Verfügbarkeit	3
Einrichtung eines AWS-Konto	4
Melde dich an für ein 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
Bewährte Methoden	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	36
Verwalten des Zugriffs mit Richtlinien	40
Wie AWS End User Messaging Push funktioniert mit IAM	42
Beispiele für identitätsbasierte Richtlinien	50
Fehlerbehebung	54
Compliance-Validierung	56
Ausfallsicherheit	57
Sicherheit der Infrastruktur	58
Konfigurations- und Schwachstellenanalyse	58
Bewährte Methoden für die Gewährleistung der Sicherheit	59
Überwachen	60
Überwachung mit CloudWatch	61
CloudTrail protokolliert	61
AWS Nachrichtenübermittlung an Endbenutzer: Geben Sie Informationen ein CloudTrail	61
Grundlegendes zu den Einträgen in der Push-Protokolldatei von AWS End User Messaging	63
AWS PrivateLink	64
Überlegungen	64
Erstellen eines Schnittstellenendpunkts	65
Erstellen einer Endpunktrichtlinie	65
Kontingente	67
Dokumentverlauf	68
.....	Ixix

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](#)
- [Zugriff auf Push für AWS Endbenutzer](#)
- [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](#)
- [Eine 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-Gerätenachrichten (ADM)

Zugriff auf Push für AWS Endbenutzer

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 Management Console.

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 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 zu stellen HTTPS, AWS bietet er 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 der Endpunkte 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 du benutzen kannst AWS End User Messaging Push Um Push-Benachrichtigungen an Ihre App zu senden, benötigen Sie zunächst eine AWS-Konto mit ausreichenden IAM Berechtigungen. Dieser AWS-Konto kann auch für andere Dienste in der verwendet werden AWS Ökosystem.

Themen

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

Melde dich an für ein AWS-Konto

Wenn Sie kein haben AWS-Konto, führen Sie die folgenden Schritte aus, um einen zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie [https://portal.aws.amazon.com/billing/die Anmeldung](https://portal.aws.amazon.com/billing/die-Anmeldung).
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, ein Root-Benutzer des AWS-Kontos wird erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen im Konto. 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 sich Ihre Root-Benutzer des AWS-Kontos, aktivieren AWS IAM Identity Center, und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melde dich an bei [AWS Management Console](#) als Kontoinhaber wählen Sie Root-Benutzer und geben Sie Ihren AWS-Konto E-Mail-Adresse. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Als Root-Benutzer anmelden im AWS-Anmeldung Benutzerleitfaden](#).

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

Anweisungen finden Sie unter [Aktivieren Sie ein virtuelles MFA Gerät für AWS-Konto Root-Benutzer \(Konsole\)](#) im IAMBenutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) in der AWS IAM Identity Center Benutzerleitfaden.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Für ein Tutorial zur Verwendung des IAM-Identity-Center-Verzeichnis als Identitätsquelle finden [Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) in der AWS IAM Identity Center Benutzerleitfaden.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM Identity Center-Benutzer anzumelden, verwenden Sie die Anmeldung, 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 unter [Anmelden bei AWS Zugriffsportal](#) im AWS-Anmeldung Benutzerleitfaden.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie unter [Einen Berechtigungssatz erstellen in](#) der AWS IAM Identity Center Benutzerleitfaden.

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

Anweisungen finden Sie unter [Gruppen hinzufügen](#) in der AWS IAM Identity Center Benutzerleitfaden.

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 unter [Einen Verschlüsselungsschlüssel und eine Schlüssel-ID von Apple beziehen und Ein Anbieterzertifikat von Apple beziehen](#) in der Apple-Dokumentation für Entwickler.
- 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.](#)

Eine 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 von Ihnen 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 authentifiziert, APNs wenn Sie Push-Benachrichtigungen senden. 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 für den Standard-Authentifizierungstyp eine der folgenden Optionen:
 - 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:

- SSLcertificate — 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 () zu aktivieren.
FCM
 - a. Wählen Sie für Firebase Cloud Messaging (FCM) die Option Aktivieren aus.
 - b. Wählen Sie für den Standardauthentifizierungstyp eine der folgenden Optionen:
 - i. Wählen Sie für Token-Anmeldeinformationen (empfohlen) die Option Dateien auswählen und dann Ihre JSON Servicedatei aus.
 - ii. Geben Sie für Schlüsselanmeldedaten Ihren Schlüssel in das APISchlüsselfeld 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 APISchlüssel Ihren API Schlüssel ein.
 - c. Geben Sie unter 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 Save Changes.

Senden einer Nachricht

Der AWS End User Messaging Push API kann transaktionale Push-Benachrichtigungen an bestimmte Gerätekennungen senden. Dieser Abschnitt enthält vollständige Codebeispiele, die Sie verwenden können, um Push-Benachrichtigungen über den AWS End User Messaging Push mithilfe der API zu senden. AWS SDK

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 für Endpunkte, Segmente und Kanäle finden Sie unter Codebeispiele.](#)

Note

Wenn Sie Push-Benachrichtigungen über den Firebase Cloud Messaging (FCM) -Dienst senden, verwenden Sie den Dienstnamen GCM in Ihrem Aufruf an den AWS End User Messaging Push API. Der Dienst Google Cloud Messaging (GCM) wurde am 10. April 2018 von Google eingestellt. Der AWS End User Messaging Push API verwendet jedoch den GCM Dienstnamen für Nachrichten, die er über den FCM Dienst sendet, um die Kompatibilität mit API Code zu gewährleisten, 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 Ersetzen *token* mit dem eindeutigen Token des Geräts und *611e3e3cdd47474c9c1399a50example* mit Ihrer 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 Ersetzen *token* mit dem eindeutigen Token des Geräts und *611e3e3cdd47474c9c1399a50example* mit Ihrer 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 unter Verwendung von zu senden. AWS CLI Ersetzen *token* mit dem eindeutigen Token des Geräts und *611e3e3cdd47474c9c1399a50example* mit Ihrer 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\"},
  \"TimeToLive\" : 309744
}
},
\"Addresses\": {
  \"token\": {
    \"ChannelType\": \"GCM\"
  }
}
}'
\\ --region us-east-1

```

Wenn Sie `ImageUrl` field for verwenden GCM, sendet Pinpoint das Feld als Datenbenachrichtigung, mit dem Schlüssel `pinpoint.notification.imageUrl`, was verhindern 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, um an iOS-Geräte wie iPhones und zu senden 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 Ersetzen *token* mit dem eindeutigen Token des Geräts und *611e3e3cdd47474c9c1399a50example* mit Ihrer Anwendungs-ID.

```
aws pinpoint send-messages \  
--application-id 611e3e3cdd47474c9c1399a50example \  
--message-request \  
'{  
  "Addresses": {  
    "token":  
    {  
      "ChannelType": "APNS"  
    }  
  },  
  "MessageConfiguration": {  
    "APNSMessage": {  
      "RawContent":  
        "{\\"aps\\": {\\"alert\\": { \\"title\\": \\"Title of my message\\", \\"body\\":  
        \\"This is a push notification for the Safari web browser.\\"}, \\"content-available\\":  
        1, \\"url-args\\": [\\"\\"]}}"  
    }  
  }  
}'  
\  
--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 Ersetzen *token* mit dem eindeutigen Token des Geräts, *611e3e3cdd47474c9c1399a50example* mit Ihrer Anwendungs-ID und *GAME_INVITATION* mit einer eindeutigen 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 for JavaScript in Node.js zu senden. In diesem Beispiel wird vorausgesetzt, dass Sie das SDK for bereits JavaScript in Node.js 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 [Setting credentials](#) in the AWS SDK for JavaScript in 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 for Python (Boto3) zu senden. In diesem Beispiel wird vorausgesetzt, dass Sie den 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 [Anmeldeinformationen](#) in der Referenz AWS SDK für Python (Boto3) API.

```
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 for 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 mithilfe von Firebase Cloud Messaging (FCM) gesendet, das Google Cloud Messaging (GCM) ersetzt. Bevor Sie Push-Benachrichtigungen an Android-Geräte senden können, müssen Sie FCM Anmeldeinformationen anfordern. 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 für iOS gesendet. Weitere Informationen zum Ausführen dieser Schritte finden Sie im Abschnitt Push-Benachrichtigungen in der Dokumentation zu [AWS Amplify Flutter](#).

Einrichten von React-Native-Push-Benachrichtigungen

Push-Benachrichtigungen für React Native-Apps werden mit Firebase Cloud Messaging (FCM) für Android und APNs für iOS gesendet. Weitere Informationen zum Ausführen dieser Schritte finden Sie im Abschnitt Push-Benachrichtigungen 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 [Eine Anwendung erstellen und Push-Kanäle aktivieren](#).

Sie können Anwendungen auch mit dem [APIAWS SDK](#), einem oder dem [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:

- [APNsKanal](#) zum Senden von Nachrichten an Benutzer von iOS-Geräten mithilfe des Apple Push Notification-Dienstes.
- [ADMKanal](#) zum Senden von Nachrichten an Benutzer von Amazon Kindle Fire-Geräten.
- [Baidu-Kanal](#) zum Senden von Nachrichten an Baidu-Benutzer.
- [GCMKanal](#) 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.

Bewährte Methoden

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, der permanent ausfällt, falls das entsprechende Plattform-Token (z. B. FCM) oder Zertifikat (z. B. APN) beim Senden von Nachrichten als ungültig eingestuft wird.

Sicherheit bei AWS Messaging-Push für Endbenutzer

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

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

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die läuft AWS Dienstleistungen in der AWS Cloud. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheit im Rahmen der [AWS Compliance-Programme](#) . Um mehr über die Compliance-Programme zu erfahren, die gelten für AWS Push-Nachrichten für Endbenutzer finden Sie unter [AWS Im Leistungsumfang aufgeschlüsselte Dienstleistungen nach Compliance-Programmen](#) .
- Sicherheit in der Cloud — Ihre Verantwortung wird bestimmt von 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 anwenden können, wenn Sie AWS Push-Nachrichten für Endbenutzer. In den folgenden Themen wird die Konfiguration beschrieben AWS Push-Benachrichtigungen für Endbenutzer, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere verwenden AWS Dienste, die Ihnen helfen, Ihre AWS Push-Ressourcen für Endbenutzer-Nachrichten.

Themen

- [Datenschutz in AWS Messaging-Push für Endbenutzer](#)
- [Identitäts- und Zugriffsmanagement für AWS Messaging-Push für Endbenutzer](#)
- [Überprüfung der Einhaltung von Vorschriften für AWS Messaging-Push für Endbenutzer](#)
- [Resilienz in AWS Messaging-Push für Endbenutzer](#)
- [Sicherheit der Infrastruktur in AWS Messaging-Push für Endbenutzer](#)
- [Konfigurations- und Schwachstellenanalyse](#)
- [Bewährte Methoden für die Gewährleistung der Sicherheit](#)

Datenschutz in AWS Messaging-Push für Endbenutzer

Das Tool AWS [Das Modell](#) AWS Push-Nachrichten für Endbenutzer. Wie in diesem Modell beschrieben AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre Inhalte zu behalten, die auf dieser Infrastruktur gehostet werden. Sie sind auch verantwortlich für die Sicherheitskonfiguration und die Verwaltungsaufgaben für AWS-Services die Sie verwenden. Weitere Informationen zum Datenschutz finden Sie in der [Datenschutzerklärung FAQ](#). Informationen zum Datenschutz in Europa finden Sie auf der [AWS Modell der geteilten Verantwortung und GDPR](#) Blogbeitrag auf der AWS Blog zum Thema Sicherheit.

Aus Datenschutzgründen empfehlen wir Ihnen, AWS-Konto Anmeldeinformationen und richten Sie einzelne Benutzer ein mit AWS IAM Identity Center or AWS Identity and Access Management (IAM). So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit zu kommunizieren AWS Ressourcen schützen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Einrichtung API und Protokollierung von Benutzeraktivitäten mit AWS CloudTrail. Für Informationen zur Verwendung von CloudTrail Pfaden zum Erfassen AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerleitfaden.
- 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 FIPS 140-3 validierte kryptografische Module benötigen AWS über eine Befehlszeilenschnittstelle oder einen API, verwenden Sie einen Endpunkt. FIPS Weitere Informationen zu den verfügbaren FIPS Endpunkten 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. Dazu gehört auch, wenn Sie mit arbeiten AWS Endbenutzer-Nachrichtendienst, Push oder ein anderes AWS-Services mit der Konsole API, AWS CLI, oder 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 einem externen Server eine URL zur Verfügung stellen, empfehlen wir dringend, dass Sie keine Anmeldeinformationen angeben, URL um Ihre Anfrage an diesen Server zu überprüfen.

Datenverschlüsselung

AWS Push-Daten für Endbenutzer-Nachrichten werden bei der Übertragung und im Speicher verschlüsselt. Wenn Sie Daten senden an AWS End User Messaging Push verschlüsselt die Daten, sobald sie empfangen und gespeichert werden. Wenn Sie Daten abrufen von AWS End User Messaging Push überträgt die Daten mithilfe aktueller Sicherheitsprotokolle an Sie.

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 hinzufügen oder in die Sie importieren AWS Push-Nachrichten für Endbenutzer. Um Ihre Daten zu verschlüsseln, AWS End User Messaging Push verwendet interne AWS Key Management Service (AWS KMS) Schlüssel, die der Dienst besitzt und in Ihrem Namen verwaltet. Diese Schlüssel werden regelmäßig rotiert. Für Informationen über AWS KMS, siehe [AWS Key Management Service Leitfaden für Entwickler](#).

Verschlüsselung während der Übertragung

AWS End User Messaging Push verwendet HTTPS Transport Layer Security (TLS) 1.2 oder höher für die Kommunikation mit Ihren Clients und Anwendungen. Um mit anderen zu kommunizieren AWS Dienstleistungen, AWS End User Messaging verwendet Push HTTPS und TLS 1.2. Darüber hinaus bei der Erstellung und Verwaltung AWS Endbenutzer-Nachrichten Pushen Sie Ressourcen mithilfe der Konsole AWS SDK, oder das AWS Command Line Interface, die gesamte Kommunikation ist mit HTTPS und TLS 1.2 gesichert.

Schlüsselverwaltung

Um deine zu verschlüsseln AWS Push-Daten für Endbenutzer-Nachrichten, AWS Endbenutzer-Nachrichten-Push verwendet interne AWS KMS Schlüssel, die der Dienst besitzt und in Ihrem Namen verwaltet. Diese Schlüssel werden regelmäßig rotiert. Sie können keine eigenen bereitstellen und verwenden AWS KMS oder andere Schlüssel zum Verschlüsseln von Daten, die Sie speichern AWS Messaging-Push für Endbenutzer.

Datenschutz für den Datenverkehr zwischen Netzwerken

Der Datenschutz im Netzwerkverkehr bezieht sich auf die Sicherung von Verbindungen und des Datenverkehrs zwischen AWS End User Messaging Push und Ihre lokalen Clients und Anwendungen sowie dazwischen AWS Endbenutzer-Nachrichten-Push und andere AWS Ressourcen im selben AWS Region. Die folgenden Funktionen und Verfahren können Ihnen dabei helfen, den Datenschutz für den Netzwerkverkehr zu gewährleisten AWS Messaging-Push für Endbenutzer.

Verkehr zwischen AWS End User Messaging Push und lokale Clients und Anwendungen

Um eine private Verbindung herzustellen zwischen AWS End User Messaging Push und Clients und Anwendungen in Ihrem lokalen Netzwerk können Sie verwenden AWS Direct Connect. Auf diese Weise können Sie Ihr Netzwerk mit einem verbinden AWS Direct Connect Ortung mithilfe eines Standard-Glasfaser-Ethernet-Kabels. Ein Ende des Kabels ist mit Ihrem Router verbunden. Das andere Ende ist mit einem verbunden AWS Direct Connect Router. Weitere Informationen finden Sie unter [Was ist AWS Direct Connect?](#) in der AWS Direct Connect Benutzerleitfaden.

Um den Zugriff auf zu sichern AWS Endbenutzer-Nachrichtenübermittlung veröffentlicht. Wir empfehlen Ihnen APIs, sich an folgende Regeln zu halten AWS Push-Anforderungen für Endbenutzer-Nachrichten für API Anrufe. AWS Für End User Messaging Push müssen die Clients Transport Layer Security (TLS) 1.2 oder höher verwenden. Clients müssen außerdem Cipher Suites mit Perfect Forward Secrecy (PFS) unterstützen, wie Ephemeral Diffie-Hellman () oder Elliptic Curve Diffie-Hellman Ephemeral (DHE). ECDHE 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 einer AWS Identity and Access Management (IAM) Schulleiter für Ihren AWS Konto. Alternativ können Sie das verwenden [AWS Security Token Service](#) (AWS STS), um temporäre Sicherheitsanmeldedaten zum Signieren von Anfragen zu generieren.

Verkehr zwischen AWS Endbenutzer-Nachrichtenübermittlung (Push) und andere AWS Ressourcen

Zur Sicherung der Kommunikation zwischen AWS Endbenutzer-Nachrichtenübermittlung, Push und andere AWS Ressourcen im selben AWS Region, AWS End User Messaging Push verwendet HTTPS standardmäßig TLS 1.2.

Identitäts- und Zugriffsmanagement für AWS Messaging-Push für Endbenutzer

AWS Identity and Access Management (IAM) ist ein AWS-Service das hilft einem Administrator, den Zugriff auf sicher zu kontrollieren AWS Ressourcen schätzen. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und zur Nutzung autorisiert werden kann (über Berechtigungen verfügt) AWS Push-Ressourcen für Endbenutzer-Nachrichten. IAM ist ein AWS-Service das Sie ohne zusätzliche Kosten nutzen können.

Themen

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

Zielgruppe

Wie benutzt du AWS Identity and Access Management (IAM) unterscheidet sich je nach der Arbeit, die Sie in AWS Push-Nachrichten für Endbenutzer.

Servicebenutzer — Wenn Sie den verwenden AWS End User Messaging Push Service für Ihre Arbeit. Anschließend stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Je mehr Sie verwenden AWS Push-Funktionen für Endbenutzernachrichten Um Ihre Arbeit erledigen zu können, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie in nicht auf eine Funktion zugreifen können AWS Push-Nachrichten für Endbenutzer, siehe [Fehlerbehebung AWS Identität und Zugriff per Push-Push für Endbenutzer](#).

Service-Administrator — Wenn Sie verantwortlich sind für AWS Push-Ressourcen für Endbenutzernachrichten in Ihrem Unternehmen, auf die Sie wahrscheinlich vollen Zugriff haben AWS Push-Nachrichten für Endbenutzer. Es ist Ihre Aufgabe, herauszufinden, welche AWS Push-Funktionen und Ressourcen für Endbenutzernachrichten, auf die Ihre Servicebenutzer zugreifen sollten. Anschließend müssen Sie Anfragen an Ihren IAM Administrator senden, um die

Berechtigungen Ihrer Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die grundlegenden Konzepte von zu verstehen IAM. Um mehr darüber zu erfahren, wie Ihr Unternehmen Folgendes nutzen IAM kann AWS Push-Nachrichten für Endbenutzer finden Sie unter [Wie AWS End User Messaging Push funktioniert mit IAM](#).

IAM Administrator — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf AWS Messaging-Push für Endbenutzer. Um ein Beispiel anzusehen AWS Identitätsbasierte Push-Richtlinien für Endbenutzernachrichten, die Sie in verwenden können IAM, finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für AWS Messaging-Push für Endbenutzer](#)

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich anmelden AWS mit Ihren Identitätsdaten. Sie müssen authentifiziert (angemeldet) sein AWS) als Root-Benutzer des AWS-Kontos, als IAM Benutzer oder indem Sie eine IAM Rolle übernehmen.

Sie können sich anmelden bei AWS als föderierte Identität mithilfe von Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) - Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als föderierte Identität anmelden, hat Ihr Administrator zuvor einen Identitätsverbund mithilfe von Rollen eingerichtet. IAM Wenn Sie darauf zugreifen AWS Wenn Sie den Verbund verwenden, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der anmelden AWS Management Console oder das AWS Zugangportal. Weitere Informationen zur Anmeldung bei AWS, siehe [So melden Sie sich bei Ihrem an AWS-Konto](#) in der AWS-Anmeldung Benutzerleitfaden.

Wenn Sie darauf zugreifen AWS programmatisch AWS stellt ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie es nicht verwenden AWS Tools, Sie müssen Anfragen selbst unterschreiben. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie unter [Signieren AWS APIAnfragen](#) im IAM Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. Zum Beispiel AWS empfiehlt die Verwendung der Multi-Faktor-Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center

Benutzerhandbuch und [Verwendung der Multi-Faktor-Authentifizierung \(\) MFA in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Der Zugriff erfolgt, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter [Aufgaben, für die Root-Benutzeranmeldedaten erforderlich](#) sind. IAM

Verbundidentität

Es hat sich bewährt, menschlichen Benutzern, einschließlich Benutzern, die Administratorzugriff benötigen, vorzuschreiben, für den Zugriff den Verbund mit einem Identitätsanbieter zu verwenden. AWS-Services mithilfe temporärer Anmeldeinformationen.

Eine föderierte Identität ist ein Benutzer aus dem Benutzerverzeichnis Ihres Unternehmens, ein Web-Identitätsanbieter, AWS Directory Service, das Identity Center-Verzeichnis oder ein beliebiger Benutzer, der zugreift AWS-Services mithilfe von Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für eine zentralisierte Zugriffsverwaltung empfehlen wir die Verwendung AWS IAM Identity Center. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten und Anwendungen. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) in der AWS IAM Identity Center Benutzerleitfaden.

IAM-Benutzer und -Gruppen

Ein [IAMBenutzer](#) ist eine Identität innerhalb Ihres AWS-Konto das über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern

erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich](#) sind.

Eine [IAMGruppe](#) ist eine Identität, die eine Sammlung von IAM Benutzern spezifiziert. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Wann sollte ein IAM Benutzer \(statt einer Rolle\) erstellt werden?](#) im IAMBenutzerhandbuch.

IAMRollen

Eine [IAMRolle](#) ist eine Identität in deinem AWS-Konto das hat spezifische Berechtigungen. Es ähnelt einem IAM Benutzer, ist jedoch keiner bestimmten Person zugeordnet. Sie können vorübergehend eine IAM Rolle in der übernehmen AWS Management Console indem Sie die [Rollen wechseln](#). Sie können eine Rolle übernehmen, indem Sie einen anrufen AWS CLI or AWS APIOperation oder mithilfe eines benutzerdefiniertenURL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie [unter Verwenden von IAM Rollen](#) im IAMBenutzerhandbuch.

IAMRollen mit temporären Anmeldeinformationen sind in den folgenden Situationen nützlich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle für einen externen Identitätsanbieter](#). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center Benutzerleitfaden.
- **Temporäre IAM Benutzerberechtigungen** — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.

- **Kontoübergreifender Zugriff** — Sie können eine IAM Rolle verwenden, um jemandem (einem vertrauenswürdigen Principal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Allerdings mit einigen AWS-Services, Sie können eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie [IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM
- **Serviceübergreifender Zugriff** — Einige AWS-Services Funktionen in anderen verwenden AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Zugriffssitzungen weiterleiten (FAS)** — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen in AWS, Sie gelten als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Rechte des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anfrage AWS-Service um Anfragen an nachgelagerte Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, die Interaktionen mit anderen erfordert AWS-Services oder zu vervollständigende Ressourcen. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle** — Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an ein AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstbezogene Rolle** — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen durchzuführen. Mit Diensten verknüpfte Rollen erscheinen in Ihrem AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen, die auf Amazon laufen EC2** — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS CLI or AWS APIAnfragen. Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instanz vorzuziehen. Um eine zuzuweisen AWS Sie erstellen ein EC2 Instanzprofil, das an die Instanz angehängt ist. Sie müssen einer Instanz eine Rolle

zuweisen und sie allen ihren Anwendungen zur Verfügung stellen. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden](#).

Informationen darüber, ob Sie IAM Rollen oder IAM Benutzer verwenden sollten, finden [Sie im Benutzerhandbuch unter Wann sollte eine IAM Rolle \(anstelle eines IAM Benutzers\) erstellt werden](#).

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff in AWS indem Sie Richtlinien erstellen und diese anhängen AWS Identitäten oder Ressourcen. Eine Richtlinie ist ein Objekt in AWS das, wenn es mit einer Identität oder Ressource verknüpft ist, ihre Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Principal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien sind gespeichert in AWS als JSON Dokumente. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter [Überblick über JSON Richtlinien](#).

Administratoren können verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAMRichtlinien definieren Berechtigungen für eine Aktion, unabhängig von der Methode, mit der Sie den Vorgang ausführen. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der abrufen AWS Management Console, der AWS CLI, oder der AWS API.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und

unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAM Richtlinien erstellen im Benutzerhandbuch](#). IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie wählen können, finden Sie im IAM Benutzerhandbuch unter [Auswahl zwischen verwalteten Richtlinien und Inline-Richtlinien](#).

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 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](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services.

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

Zugriffskontrolllisten (ACLs)

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.

Amazon S3, AWS WAF, und Amazon VPC sind Beispiele für Dienste, die unterstützen ACLs. Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAMBenutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAMBenutzerhandbuch unter [Berechtigungsgrenzen für IAM Entitäten](#).
- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer AWS-Konten den Ihr Unternehmen besitzt. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und finden Sie SCPs unter [Richtlinien zur Servicesteuerung](#) in der AWS Organizations Benutzerleitfaden.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Sitzungsrichtlinien](#).

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Um zu erfahren, wie AWS bestimmt, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, siehe [Bewertungslogik für Richtlinien](#) im IAMBenutzerhandbuch.

Wie AWS End User Messaging Push funktioniert mit IAM

Vor der Verwendung IAM zur Verwaltung des Zugriffs auf AWS erfahren Sie, welche IAM Funktionen für End User Messaging Push verfügbar sind AWS Messaging-Push für Endbenutzer.

IAMFunktionen, die Sie mit verwenden können AWS Push-Nachrichten für Endbenutzer

IAMFunktion	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
Hauptberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Um einen allgemeinen Überblick darüber zu erhalten, wie AWS Endbenutzer-Nachrichten, Push und andere AWS Dienste funktionieren mit den meisten IAM Funktionen, siehe [AWS Dienste, mit denen IAM](#) im IAMBenutzerhandbuch gearbeitet werden kann.

Identitätsbasierte Richtlinien für AWS Messaging-Push für Endbenutzer

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAMRichtlinien erstellen im Benutzerhandbuch](#). IAM

Mit IAM identitätsbasierten Richtlinien können Sie zulässige oder verweigernde Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zulässig oder verweigert werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden können, finden Sie im IAMBenutzerhandbuch unter [Referenz zu IAM JSON Richtlinienelementen](#).

Beispiele für identitätsbasierte Richtlinien für AWS Messaging-Push für Endbenutzer

Um Beispiele zu sehen AWS Identitätsbasierte Push-Richtlinien für Endbenutzer-Nachrichten finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS Messaging-Push für Endbenutzer](#)

Ressourcenbasierte Richtlinien innerhalb AWS Messaging-Push für Endbenutzer

Unterstützt ressourcenbasierte Richtlinien: Ja

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 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](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services.

Um den kontoübergreifenden Zugriff zu ermöglichen, können Sie in einer ressourcenbasierten Richtlinie ein ganzes Konto oder IAM Entitäten in einem anderen Konto als Prinzipal angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM Administrator des vertrauenswürdigen Kontos auch der Prinzipalentscheidung (Benutzer oder Rolle) die Erlaubnis erteilen, auf die Ressource zuzugreifen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie [IAMim IAMBenutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#).

Politische Maßnahmen für AWS Messaging-Push für Endbenutzer

Unterstützt Richtlinienaktionen: Ja

Administratoren können Folgendes verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörigen AWS API-Betrieb. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur Berechtigungen erforderlich sind und für die es keine entsprechende Operation gibt. API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Um eine Liste von zu sehen AWS Push-Aktionen für Endbenutzer-Nachrichten finden Sie unter [Aktionen, die definiert sind von AWS Push-Nachrichten für Endbenutzer](#) in der Referenz zur Serviceautorisierung.

Politische Aktionen in AWS End User Messaging Push verwenden vor der Aktion das folgende Präfix:

```
mobiletargeting
```

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

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

Um Beispiele zu sehen AWS Identitätsbasierte Push-Richtlinien für Endbenutzer-Nachrichten finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für AWS Messaging-Push für Endbenutzer](#)

Richtlinienressourcen für AWS Push mit Nachrichtenübermittlung für Endbenutzer

Unterstützt Richtlinienressourcen: Ja

Administratoren können Folgendes verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem [Amazon-Ressourcennamen \(ARN\)](#) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Um eine Liste von zu sehen AWS Push-Ressourcentypen für Endbenutzer-Nachrichten und ihre ARNs, siehe [Ressourcen, definiert von AWS Push für Endbenutzer-Nachrichten](#) in der Referenz zur Serviceautorisierung. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter Definierte [Aktionen von AWS Push mit Nachrichtenübermittlung an Endbenutzer](#).

Um Beispiele zu sehen AWS Identitätsbasierte Push-Richtlinien für Endbenutzer-Nachrichten finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS Messaging-Push für Endbenutzer](#)

Schlüssel für die Bedingungen der Richtlinien für AWS Push-Nachrichten für Endbenutzer

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können Folgendes verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition` Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition` Element angeben, AWS wertet sie mithilfe einer logischen AND Operation

aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Sie können einem IAM Benutzer beispielsweise nur dann Zugriff auf eine Ressource gewähren, wenn sie mit seinem IAM Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinienelemente: Variablen und Tags](#).

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Um alle zu sehen AWS globale Bedingungsschlüssel finden Sie unter [AWS Kontexttasten für globale Bedingungen](#) im IAMBenutzerhandbuch.

Um eine Liste von zu sehen AWS Endbenutzer-Nachrichten Drücken Sie die Bedingungstasten, siehe [Bedingungstasten für AWS Push für Endbenutzer-Nachrichten](#) in der Referenz zur Serviceautorisierung. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Aktionen definiert durch AWS Push-Nachrichten für Endbenutzer](#).

Um Beispiele zu sehen AWS Identitätsbasierte Push-Richtlinien für Endbenutzer-Nachrichten finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS Messaging-Push für Endbenutzer](#)

ACLsin AWS Messaging-Push für Endbenutzer

UnterstütztACLs: Nein

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

ABACmit AWS Messaging-Push für Endbenutzer

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen auf der Grundlage von Attributen definiert werden. In AWS, diese Attribute werden Tags genannt. Sie können Tags an IAM Entitäten (Benutzer oder Rollen) und an viele andere anhängen AWS Ressourcen schätzen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt vonABAC. Anschließend entwerfen Sie ABAC Richtlinien, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt, auf die er zugreifen möchte.

ABAC ist hilfreich in Umgebungen, die schnell wachsen, und hilft in Situationen, in denen die Richtlinienverwaltung umständlich wird.

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 finden Sie ABAC unter [Was ist? ABAC](#) im IAMBenutzerhandbuch. Ein Tutorial mit Schritten zur Einrichtung finden Sie im ABAC Benutzerhandbuch unter [Verwenden der attributbasierten Zugriffskontrolle \(ABAC\)](#). IAM

Verwenden temporärer Anmeldeinformationen mit AWS Messaging-Push für Endbenutzer

Unterstützt temporäre Anmeldeinformationen: Ja

Etwas AWS-Services funktioniert nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Für zusätzliche Informationen, einschließlich AWS-Services mit temporären Anmeldeinformationen arbeiten, finden Sie unter [AWS-Services mit denen IAM](#) im IAMBenutzerhandbuch gearbeitet werden kann.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich bei der AWS Management Console mit einer beliebigen Methode außer einem Benutzernamen und einem Passwort. Zum Beispiel, wenn Sie darauf zugreifen AWS Wenn Sie den Single Sign-On-Link (SSO) Ihres Unternehmens verwenden, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Rollenwechsel finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAMBenutzerhandbuch.

Sie können temporäre Anmeldeinformationen manuell erstellen, indem Sie den AWS CLI or AWS API. Sie können dann diese temporären Anmeldeinformationen für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen unter IAM](#).

Serviceübergreifende Prinzipalberechtigungen für AWS Push mit Nachrichtenübermittlung für Endbenutzer

Unterstützt Forward-Access-Sitzungen (FAS): Ja

Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen in AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Rechte des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anfrage AWS-Service um Anfragen an nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, die Interaktionen mit anderen erfordert AWS-Services oder zu vervollständigende Ressourcen. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für AWS Push mit Nachrichtenübermittlung für Endbenutzer

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschen IAM. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an ein AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Das Ändern der Berechtigungen für eine Servicerolle kann fehlerhaft sein AWS Push-Funktionalität für Endbenutzer-Nachrichten Bearbeiten Sie Servicerollen nur, wenn AWS End User Messaging Push bietet dazu Anleitungen.

Mit Diensten verknüpfte Rollen für AWS Push mit Nachrichtenübermittlung für Endbenutzer

Unterstützt serviceverknüpfte Rollen: Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen durchzuführen. Mit Diensten verknüpfte Rollen erscheinen in Ihrem AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen Rollen finden Sie unter [AWS Dienste, 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 Messaging-Push für Endbenutzer

Standardmäßig sind Benutzer und Rollen nicht berechtigt, etwas zu erstellen oder zu ändern AWS Push-Ressourcen für Endbenutzer-Nachrichten. Sie können auch keine Aufgaben mithilfe der ausführen AWS Management Console, AWS Command Line Interface (AWS CLI), oder AWS API. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

Informationen zum Erstellen einer IAM identitätsbasierten Richtlinie anhand dieser JSON Beispieldokumente finden Sie unter [IAMRichtlinien erstellen](#) im IAMBenutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, definiert von AWS Push-Nachrichten für Endbenutzer, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für AWS Push für Endbenutzer-Nachrichten](#) in der Referenz zur Serviceautorisierung.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwendung der AWS Push-Konsole für Endbenutzer-Messaging](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand etwas erstellen, darauf zugreifen oder löschen kann AWS Push-Ressourcen für Endbenutzer-Nachrichten in Ihrem Konto. Diese Aktionen können Kosten für Sie verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Fangen Sie an mit AWS verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Um zu beginnen, Ihren Benutzern und Workloads Berechtigungen

zu gewähren, verwenden Sie die AWS verwaltete Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie Folgendes definieren AWS vom Kunden verwaltete Richtlinien, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS verwaltete Richtlinien](#) oder [AWS verwaltete Richtlinien für Jobfunktionen](#) im IAMBenutzerhandbuch.

- Berechtigungen mit den geringsten Rechten anwenden — Wenn Sie Berechtigungen mit IAM Richtlinien festlegen, gewähren Sie nur die Berechtigungen, die für die Ausführung 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 IAM zum Anwenden von Berechtigungen finden Sie [IAMim Benutzerhandbuch unter Richtlinien und Berechtigungen](#). IAM
- Verwenden Sie Bedingungen in IAM Richtlinien, um den Zugriff weiter einzuschränken — Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen einzuschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um anzugeben, dass alle Anfragen mit gesendet werden müssenSSL. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese über eine bestimmte AWS-Service, wie beispielsweise AWS CloudFormation. Weitere Informationen finden Sie unter [IAMJSONRichtlinienelemente: Zustand](#) im IAMBenutzerhandbuch.
- Verwenden Sie IAM Access Analyzer, um Ihre IAM Richtlinien zu validieren, um sichere und funktionale Berechtigungen zu gewährleisten. IAM Access Analyzer validiert neue und bestehende Richtlinien, sodass die Richtlinien der IAM Richtliniensprache (JSON) und den IAM bewährten Methoden entsprechen. IAMAccess Analyzer bietet mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen, um Sie bei der Erstellung sicherer und funktionaler Richtlinien zu unterstützen. Weitere Informationen finden Sie unter [IAMAccess Analyzer-Richtlinienvvalidierung](#) im IAMBenutzerhandbuch.
- Multi-Faktor-Authentifizierung erforderlich (MFA) — Wenn Sie ein Szenario haben, in dem IAM Benutzer oder ein Root-Benutzer in Ihrem AWS-Konto, MFA für zusätzliche Sicherheit einschalten. Wenn Sie festlegen möchten, MFA wann API Operationen aufgerufen werden, fügen Sie MFA Bedingungen zu Ihren Richtlinien hinzu. Weitere Informationen finden Sie unter [Konfiguration des MFA -geschützten API Zugriffs](#) im IAMBenutzerhandbuch.

Weitere Informationen zu bewährten Methoden finden Sie unter [Bewährte Sicherheitsmethoden IAM im IAM](#) Benutzerhandbuch. IAM

Verwendung der AWS Push-Konsole für Endbenutzer-Messaging

Um auf die zuzugreifen AWS Für die Endbenutzer-Nachrichten-Push-Konsole benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu folgenden Elementen aufzulisten und einzusehen AWS Push-Ressourcen für Endbenutzernachrichten in Ihrem AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die restriktiver ist als die erforderlichen Mindestberechtigungen, funktioniert die Konsole für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie nicht wie vorgesehen.

Sie müssen Benutzern, die nur Anrufe tätigen, keine Mindestberechtigungen für die Konsole gewähren AWS CLI oder das AWS API. Erlauben Sie stattdessen nur den Zugriff auf die Aktionen, die dem API Vorgang entsprechen, den sie ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen das weiterhin verwenden können AWS Hängen Sie auch die Push-Konsole für Endbenutzer-Messaging an `AWSEndUserMessaging` AWS verwaltete Richtlinie für die Entitäten. Weitere Informationen finden Sie im [Benutzerhandbuch unter Hinzufügen von Berechtigungen für einen IAM Benutzer](#).

```
{
  "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

Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen könnten, die es IAM Benutzern ermöglicht, die Inline-Richtlinien und verwalteten Richtlinien einzusehen, die mit ihrer Benutzeridentität verknüpft sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe von AWS CLI or AWS API.

```

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

```

```
    ],
    "Resource": "*"
  }
]
```

Fehlerbehebung AWS Identität und Zugriff per Push-Push für Endbenutzer

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS Endbenutzer-Nachrichten-Push und IAM.

Themen

- [Ich bin nicht berechtigt, eine Aktion auszuführen in AWS Messaging-Push für Endbenutzer](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Leute außerhalb meines AWS-Konto um auf meine zuzugreifen AWS Push-Ressourcen für Endbenutzer-Nachrichten](#)

Ich bin nicht berechtigt, eine Aktion auszuführen in AWS Messaging-Push für Endbenutzer

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 `mateojackson` IAM Benutzer versucht, die Konsole zu verwenden, um Details zu einer fiktiven `my-example-widget` Ressource anzuzeigen, aber nicht über die fiktiven `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 AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht autorisiert sind, die `iam:PassRole` Aktion auszuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an übergeben können AWS Push-Nachrichten für Endbenutzer.

Etwas AWS-Services ermöglicht 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 auszuführen in AWS Push für Endbenutzer-Messaging. 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 Leute außerhalb meines AWS-Konto um auf meine zuzugreifen AWS Push-Ressourcen für Endbenutzer-Nachrichten

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. 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:

- Um zu erfahren, ob AWS End User Messaging Push unterstützt diese Funktionen, siehe [Wie AWS End User Messaging Push funktioniert mit IAM](#).

- Um zu erfahren, wie Sie Zugriff auf Ihre Ressourcen gewähren können AWS-Konten die Ihnen gehören, finden Sie unter [Gewähren des Zugriffs für einen IAM Benutzer in einem anderen AWS-Konto die Sie besitzen, finden Sie](#) im IAMBenutzerhandbuch.
- Um zu erfahren, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren AWS-Konten, siehe Zugriff [gewähren auf AWS-Konten Eigentum Dritter](#) im IAMBenutzerhandbuch.
- Informationen zur [Bereitstellung des Zugriffs über einen Identitätsverbund finden Sie im Benutzerhandbuch unter Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#). IAM
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie [IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM

Überprüfung der Einhaltung von Vorschriften für AWS Messaging-Push für Endbenutzer

Um zu erfahren, ob ein AWS-Service fällt in den Geltungsbereich bestimmter Compliance-Programme, siehe [AWS-Services im Geltungsbereich nach Compliance-Programm](#) Compliance-Programmen das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme](#) .

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

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Verwendung von AWS-Services hängt von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS bietet die folgenden Ressourcen zur Unterstützung bei der Einhaltung von Vorschriften:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen auf AWS die sich auf Sicherheit und Compliance konzentrieren.
- [Architektur für HIPAA Sicherheit und Compliance auf Amazon Web Services](#) — Dieses Whitepaper beschreibt, wie Unternehmen Folgendes nutzen können AWS um geeignete Anwendungen zu erstellenHIPAA.

Note

Nicht alle AWS-Services sind HIPAA berechtigt. Weitere Informationen finden Sie in der [Referenz für HIPAA qualifizierte Dienste](#).

- [AWS Ressourcen zur Einhaltung von Vorschriften](#) — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. Die Leitfäden fassen die bewährten Methoden zur Sicherung zusammen AWS-Services und geben einen Überblick über die Leitlinien für Sicherheitskontrollen in verschiedenen Regelwerken (darunter National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI) und International Organization for Standardization (ISO)).
- [Bewertung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Das AWS Config Der Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Das AWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS Ressourcen und um Ihre Einhaltung der Sicherheitsstandards und Best Practices der Sicherheitsbranche zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [Amazon GuardDuty](#) — Das AWS-Service erkennt potenzielle Bedrohungen für Ihr AWS-Konten, Workloads, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen zu erfüllen PCIDSS, z. B. durch die Erfüllung der Anforderungen zur Erkennung von Eindringlingen, die in bestimmten Compliance-Frameworks vorgeschrieben sind.
- [AWS Audit Manager](#)— Das AWS-Service hilft Ihnen dabei, Ihre kontinuierlich zu überprüfen AWS Nutzung, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz in AWS Messaging-Push für Endbenutzer

Das Tool AWS Die globale Infrastruktur basiert auf AWS-Regionen und Availability Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke

mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zur AWS-Regionen und Availability Zones, siehe [AWS Globale Infrastruktur](#).

Zusätzlich zu AWS globale Infrastruktur, AWS End User Messaging Push bietet verschiedene Funktionen zur Unterstützung Ihrer Datenausfallsicherheit und Backup-Anforderungen.

Sicherheit der Infrastruktur in AWS Messaging-Push für Endbenutzer

Als verwalteter Dienst AWS End User Messaging Push ist geschützt durch AWS Verfahren zur globalen Netzwerksicherheit, die im Whitepaper [Amazon Web Services: Sicherheitsprozesse im Überblick](#) beschrieben werden.

Sie verwenden AWS veröffentlichte API Aufrufe zum Zugriff AWS Endbenutzer-Messaging Push über das Netzwerk. Die Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Die Clients müssen außerdem Cipher Suites mit Perfect Forward Secrecy (PFS) wie (Ephemeral Diffie-Hellman) oder DHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. ECDHE Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Darüber hinaus müssen Anfragen mithilfe einer Zugriffsschlüssel-ID und eines geheimen Zugriffsschlüssels, der einem Prinzipal zugeordnet ist, signiert werden. IAM Oder Sie können das verwenden [AWS Security Token Service](#) (AWS STS), um temporäre Sicherheitsanmeldedaten zum Signieren von Anfragen zu generieren.

Konfigurations- und Schwachstellenanalyse

Als verwalteter Service AWS End User Messaging Push ist geschützt durch AWS Verfahren zur globalen Netzwerksicherheit, die im Whitepaper [Amazon Web Services: Sicherheitsprozesse im Überblick](#) beschrieben werden. Das bedeutet, dass AWS verwaltet und führt grundlegende Sicherheitsaufgaben und -verfahren durch, 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 Zugriffsmanagementkonten (IAM) zur Steuerung des Zugriffs auf API Operationen, insbesondere Operationen, die Ressourcen erstellen, ändern oder löschen. Für sie umfassen diese Ressourcen Projekte, Kampagnen und Reisen. API

- Erstellen Sie einen individuellen Benutzer für jede Person, die -Ressourcen verwaltet, einschließlich Sie selbst. Benutze nicht AWS Root-Anmeldeinformationen zur Verwaltung von Ressourcen.
- 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 AWS Messaging-Push für Endbenutzer](#). Weitere Informationen zu finden IAM Sie unter [AWS Identity and Access Management](#). Informationen zu IAM bewährten Methoden finden Sie unter [IAMBewährte Methoden](#).

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 AWS End User Messaging Push 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 Alarmer festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU Nutzung 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 aus AWS 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 API Push-Anrufen für AWS Endbenutzer-Nachrichten mit AWS CloudTrail

AWS End User Messaging Push ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in AWS End User Messaging Push ausgeführt wurden. CloudTrail erfasst alle API Aufrufe von AWS End User Messaging Push als Ereignisse. Zu den erfassten Anrufen gehören Anrufe von der AWS End User Messaging-Push-Konsole und Code-Aufrufe an die AWS API Endbenutzer-Nachrichten-Push-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 gesammelten Informationen können Sie die Anfrage CloudTrail, die an AWS End User Messaging Push gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen 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 AWS

Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen AWS-Konto. Weitere Informationen finden Sie unter [Ereignisse mit dem 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 SNS Amazon-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 Reference](#) protokolliert CloudTrail und sind in dieser 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 im [CloudTrail userIdentityElement](#).

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 Ihrem VPC und AWS End User Messaging Push herzustellen. Sie können auf AWS End User Messaging Push zugreifen, als ob es in Ihrem System wäre VPC, ohne ein Internet-Gateway, ein NAT-Gerät, eine Verbindung oder AWS Direct Connect eine VPN-Verbindung verwenden zu müssen. Instanzen in Ihrem VPC-System 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 das Ausführen von Aufrufen aller seiner API-Aktionen über den Endpunkt der Benutzeroberfläche.

VPC-Endpunkt-Richtlinien 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 VPC Amazon-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 Private DNS für den Schnittstellenendpunkt aktivieren, können Sie API Anfragen an AWS End User Messaging Push unter Verwendung DNS des regionalen Standardnamens stellen. z. B. `com.amazonaws.us-east-1.pinpoint`.

Erstellen einer Endpunktrichtlinie für Ihren Schnittstellen-Endpunkt

Eine Endpunktrichtlinie ist eine IAM Ressource, die Sie an einen Schnittstellenendpunkt anhängen 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 Ihrem 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 AWSServices 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
Größe der Payload für Amazon Device Messaging (ADM) -Nachrichten	6 KB pro Nachricht	Nein
Größe der Nachrichtennutzlast des Apple Push-Benachrichtigungsdienstes (APNs)	4 KB pro Nachricht	Nein
APNs Sandbox-Nachricht – Nutzlastgröße	4 KB pro Nachricht	Nein
Baidu Cloud Push-Nachricht – Nutzlastgröße	4 KB pro Nachricht	Nein
Größe der Nachrichtennutzlast von Firebase Cloud Messaging (FCM)	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 Endbenutzerhandbuches 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.