



Panduan Pengguna

AWS Push Pesan Pengguna Akhir



AWS Push Pesan Pengguna Akhir: Panduan Pengguna

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

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

Table of Contents

Apa itu AWS End User Messaging Push?	1
Apakah Anda pengguna AWS End User Messaging Push pertama kali?	1
Fitur dari AWS End User Messaging Push	1
Mengakses AWS Push Pesan Pengguna Akhir	2
Ketersediaan wilayah	3
Menyiapkan Akun AWS	4
Mendaftar untuk Akun AWS	4
Buat pengguna dengan akses administratif	4
Memulai	7
Membuat aplikasi dan mengaktifkan saluran push	8
Kontekstual	8
Prasyarat	9
Prosedur	9
Menonaktifkan saluran push	11
Mengirim pesan push	12
Sumber daya tambahan	25
Menerima pemberitahuan push di aplikasi Anda	26
Menyiapkan Pemberitahuan Push Swift	26
Bekerja dengan APNs token	26
Siapkan notifikasi push Android	27
Menyiapkan Pemberitahuan Flutter Push	27
Menyiapkan Pemberitahuan Push React Native	27
Membuat aplikasi	27
Menangani pemberitahuan push	28
Menghapus aplikasi	29
Kontekstual	29
Prosedur	29
Praktik terbaik	30
Mengirim notifikasi push volume tinggi	30
Keamanan	31
Perlindungan data	32
Enkripsi data	33
Enkripsi bergerak	33
Manajemen kunci	33

Privasi lalu lintas antar jaringan	34
Pengelolaan identitas dan akses	35
Audiens	35
Mengautentikasi dengan identitas	36
Mengelola akses menggunakan kebijakan	40
Cara Kerja AWS End User Messaging Push IAM	42
Contoh kebijakan berbasis identitas	49
Pemecahan Masalah	53
Validasi kepatuhan	55
Ketangguhan	56
Keamanan Infrastruktur	57
Konfigurasi dan analisis kerentanan	57
Praktik terbaik keamanan	58
Pemantauan	59
Pemantauan CloudWatch dengan	60
CloudTrail log	60
AWS Informasi Push Pesan Pengguna Akhir di CloudTrail	60
AWS Memahami entri file log Push Pesan Pengguna Akhir	61
AWS PrivateLink	63
Pertimbangan	63
Membuat sebuah titik akhir antarmuka	63
Membuat kebijakan titik akhir	64
Kuota	66
Riwayat dokumen	67
.....	lxviii

Apa itu AWS End User Messaging Push?

Note

Fitur pemberitahuan Push dari Amazon Pinpoint sekarang disebut AWS End User Messaging.

Dengan AWS End User Messaging Push, Anda dapat melibatkan pengguna aplikasi Anda dengan mengirimkan pemberitahuan push melalui saluran notifikasi push. Kami mendukung Layanan Pemberitahuan Push Apple (APNs), Firebase Cloud Messaging (FCM), Amazon Device Messaging (ADM), dan Baidu Push.

Topik

- [Apakah Anda pengguna AWS End User Messaging Push pertama kali?](#)
- [Fitur dari AWS End User Messaging Push](#)
- [Mengakses AWS Push Pesan Pengguna Akhir](#)
- [Ketersediaan wilayah](#)

Apakah Anda pengguna AWS End User Messaging Push pertama kali?

Jika Anda adalah pengguna pertama kali dari AWS End User Messaging Push, kami sarankan Anda mulai dengan membaca bagian berikut:

- [Menyiapkan Akun AWS](#)
- [Memulai dengan AWS End User Messaging Push](#)
- [Membuat aplikasi dan mengaktifkan saluran push](#)

Fitur dari AWS End User Messaging Push

Anda dapat mengirim pemberitahuan push ke aplikasi Anda menggunakan saluran terpisah untuk layanan pemberitahuan push berikut:

- Firebase Cloud Messaging (FCM)
- Layanan Pemberitahuan Push Apple (APNs)

Note

Anda dapat menggunakannya APNs untuk mengirim pesan ke perangkat iOS seperti iPhones dan iPads, serta ke browser Safari di perangkat macOS, seperti laptop dan desktop Mac.

- Dorongan Awan Baidu
- Pesan Perangkat Amazon (ADM)

Mengakses AWS Push Pesan Pengguna Akhir

Jelaskan secara singkat berbagai cara untuk mendapatkan akses ke layanan, baik dengan konsol CLI, atau API.

Anda dapat mengelola AWS End User Messaging Push menggunakan antarmuka berikut:

AWS Konsol Push Pesan Pengguna Akhir

Antarmuka web tempat Anda membuat dan mengelola sumber daya AWS End User Messaging Push. Jika Anda telah mendaftar Akun AWS, Anda dapat mengakses konsol Push Pesan Pengguna AWS Akhir dari AWS Management Console.

AWS Command Line Interface

Berinteraksi dengan AWS layanan menggunakan perintah di shell baris perintah Anda. AWS Command Line Interface ini didukung di Windows, macOS, dan Linux. Untuk informasi selengkapnya tentang AWS CLI, lihat [Panduan AWS Command Line Interface Pengguna](#). Anda dapat menemukan perintah AWS End User Messaging Push di [AWS CLI Command Reference](#).

AWS SDKs

Jika Anda seorang pengembang perangkat lunak yang lebih suka membangun aplikasi menggunakan bahasa khusus APIs daripada mengirimkan permintaan melalui HTTP atau HTTPS, AWS menyediakan pustaka, kode sampel, tutorial, dan sumber daya lainnya. Pustaka ini menyediakan fungsi dasar yang mengotomatiskan tugas, seperti menandatangani permintaan Anda secara kriptografis, mencoba ulang permintaan, dan menangani respons kesalahan. Fungsi-

fungsi ini membantu membuatnya lebih efisien bagi Anda untuk memulai. Untuk informasi lebih lanjut, lihat [Alat untuk Membangun di AWS](#).

Ketersediaan wilayah

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

Untuk mempelajari selengkapnya Wilayah AWS, lihat [Menentukan Wilayah AWS akun mana yang dapat digunakan](#) di Referensi Umum Amazon Web. [Untuk daftar semua Wilayah tempat Push Pesan Pengguna AWS Akhir saat ini tersedia dan titik akhir untuk setiap Wilayah, lihat Titik akhir dan kuota untuk Amazon API Pinpoint dan titik akhir layanan di AWS](#) Referensi Umum Amazon Web. Untuk mempelajari lebih lanjut tentang jumlah Availability Zone yang tersedia di setiap Wilayah, lihat [infrastruktur AWS global](#).

Menyiapkan Akun AWS

Sebelum Anda dapat menggunakan AWS End User Messaging Push untuk mengirim pemberitahuan push ke aplikasi Anda, Anda harus terlebih dahulu mendapatkan IAM izin Akun AWS yang memadai. Ini juga Akun AWS dapat digunakan untuk layanan lain di AWS ekosistem.

Topik

- [Mendaftar untuk Akun AWS](#)
- [Buat pengguna dengan akses administratif](#)

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/pendaftaran>.
2. Ikuti petunjuk online.

Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua AWS layanan dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan mengunjungi <https://aws.amazon.com/> dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Pengguna root akun AWS

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

Untuk bantuan masuk dengan menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) di AWS Sign-In Panduan Pengguna.

2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat [Mengaktifkan MFA perangkat virtual untuk pengguna Akun AWS root \(konsol\) Anda](#) di Panduan IAM Pengguna.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat IAM Identitas.

Untuk mendapatkan petunjuk, silakan lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat IAM Identitas, berikan akses administratif ke pengguna.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM](#) di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat IAM Identitas, gunakan login URL yang dikirim ke alamat email saat Anda membuat pengguna Pusat IAM Identitas.

Untuk bantuan masuk menggunakan pengguna Pusat IAM Identitas, lihat [Masuk ke portal AWS akses](#) di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat IAM Identitas, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

Untuk petunjuknya, lihat [Membuat set izin](#) di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

Untuk petunjuk, lihat [Menambahkan grup](#) di Panduan AWS IAM Identity Center Pengguna.

Memulai dengan AWS End User Messaging Push

Untuk menyiapkan AWS End User Messaging Push sehingga dapat mengirim pemberitahuan push ke aplikasi Anda, Anda harus terlebih dahulu memberikan kredensi yang mengizinkan AWS End User Messaging Push untuk mengirim pesan ke aplikasi Anda. Kredensi yang Anda berikan bergantung pada sistem notifikasi push yang Anda gunakan:

- Untuk kredensi layanan (APN) Pemberitahuan Push Apple, lihat [Mendapatkan kunci enkripsi dan ID kunci dari Apple](#) dan [Mendapatkan sertifikat penyedia dari Apple](#) dalam dokumentasi Pengembang Apple.
- [Untuk kredensial Firebase Cloud Messaging \(FCM\) yang dapat diperoleh melalui Firebase console, lihat Firebase Cloud Messaging.](#)
- [Untuk kredensi Baidu, lihat Baidu.](#)
- Untuk kredensial Amazon Device Messaging (ADM), lihat [Mendapatkan](#) Kredensial.

Membuat aplikasi dan mengaktifkan saluran push

Sebelum Anda dapat menggunakan AWS End User Messaging Push untuk mengirim pemberitahuan push, Anda harus terlebih dahulu membuat aplikasi dan mengaktifkan saluran pemberitahuan push.

Kontekstual

Aplikasi

Aplikasi adalah wadah penyimpanan untuk semua pengaturan AWS End User Messaging Push Anda. Aplikasi ini juga menyimpan saluran Amazon Pinpoint, kampanye, dan pengaturan perjalanan Anda.

Kunci

Kunci penandatanganan pribadi yang digunakan oleh AWS End User Messaging Push untuk menandatangani token APNs otentikasi secara kriptografis. Anda mendapatkan kunci penandatanganan dari akun pengembang Apple Anda.

Jika Anda memberikan kunci penandatanganan, AWS End User Messaging Push menggunakan token untuk mengautentikasi setiap pemberitahuan push yang Anda kirim. APNs Dengan kunci penandatanganan, Anda dapat mengirim pemberitahuan push ke lingkungan APNs produksi dan kotak pasir.

Tidak seperti sertifikat, kunci penandatanganan Anda tidak kedaluwarsa. Anda hanya memberikan kunci Anda sekali, dan Anda tidak perlu memperbaruinya nanti. Anda dapat menggunakan kunci penandatanganan yang sama untuk beberapa aplikasi. Untuk informasi selengkapnya, lihat [Berkomunikasi dengan APNs menggunakan token autentikasi](#) di Bantuan Xcode.

Sertifikat

TLS Sertifikat yang digunakan AWS End User Messaging Push untuk mengautentikasi APNs saat Anda mengirim pemberitahuan push. APNs Sertifikat dapat mendukung lingkungan produksi dan kotak pasir, atau hanya dapat mendukung lingkungan kotak pasir. Anda mendapatkan sertifikat dari akun pengembang Apple Anda.

Sertifikat berakhir setelah satu tahun. Ketika ini terjadi, Anda harus membuat sertifikat baru, yang kemudian Anda berikan ke AWS End User Messaging Push untuk memperbarui pengiriman

pemberitahuan push. Untuk informasi selengkapnya, lihat [Berkomunikasi dengan APNs menggunakan TLS sertifikat](#) di Bantuan Xcode.

Prasyarat

Sebelum Anda dapat menggunakan saluran push apa pun, Anda memerlukan kredensi yang valid untuk layanan push. Untuk informasi lebih lanjut tentang mendapatkan kredensial, lihat [Memulai dengan AWS End User Messaging Push](#)

Prosedur

Ikuti petunjuk ini untuk membuat aplikasi dan mengaktifkan salah satu saluran push. Untuk menyelesaikan prosedur ini, Anda hanya perlu memasukkan nama aplikasi. Anda dapat mengaktifkan atau menonaktifkan salah satu saluran push di lain waktu.

1. Buka konsol Push Pesan Pengguna AWS Akhir di <https://console.aws.amazon.com/push-notifications/>.
2. Pilih Create application (Buat aplikasi).
3. Untuk nama Aplikasi masukkan nama untuk aplikasi Anda.
4. (Opsional) Ikuti langkah opsional ini untuk mengaktifkan layanan Pemberitahuan Push Apple (APNs).
 - a. Untuk layanan Pemberitahuan Push Apple (APNs) pilih Aktifkan.
 - b. Untuk jenis otentikasi default pilih:
 - i. Jika Anda memilih Kredensi kunci, berikan informasi berikut dari akun pengembang Apple Anda. AWS End User Messaging Push memerlukan informasi ini untuk membuat token otentikasi.
 - ID Kunci — ID yang ditetapkan ke kunci penandatanganan Anda.
 - Bundle identifier — ID yang ditetapkan ke aplikasi iOS Anda.
 - Pengenal tim — ID yang ditetapkan ke tim akun pengembang Apple Anda.
 - Kunci autentikasi — File.p8 yang Anda unduh dari akun pengembang Apple saat Anda membuat kunci otentikasi.
 - ii. Jika Anda memilih kredensi Sertifikat, berikan informasi berikut:

- SSLsertifikat - Berkas.p12 untuk sertifikat AndaTLS.
 - Kata sandi sertifikat — Jika Anda menetapkan kata sandi ke sertifikat Anda, masukkan di sini.
 - Jenis sertifikat — Pilih jenis sertifikat yang akan digunakan.
5. (Opsional) Ikuti langkah opsional ini untuk mengaktifkan Firebase Cloud Messaging (FCM).
 - a. Untuk Firebase Cloud Messaging (FCM) pilih Aktifkan.
 - b. Untuk jenis otentikasi default pilih:
 - i. Untuk kredensi Token (disarankan) pilih Pilih file dan kemudian pilih file layanan JSON Anda.
 - ii. Untuk kredensi Kunci masukkan kunci Anda di API kunci.
 6. (Opsional) Ikuti langkah opsional ini untuk mengaktifkan Baidu Cloud Push.
 - a. Untuk Baidu Cloud Push pilih Aktifkan.
 - b. Untuk APIkunci masukkan API kunci Anda.
 - c. Untuk kunci Rahasia masukkan kunci rahasia Anda.
 7. (Opsional) Ikuti langkah opsional ini untuk mengaktifkan Pesan Perangkat Amazon.
 - a. Untuk Amazon Device Messaging pilih Aktifkan.
 - b. Untuk ID Klien, masukkan ID klien Anda.
 - c. Untuk rahasia Klien masukkan rahasia klien Anda.
 8. Pilih Create application (Buat aplikasi).

Menonaktifkan saluran push

Ikuti petunjuk ini untuk menonaktifkan salah satu saluran push.

1. Buka konsol Push Pesan Pengguna AWS Akhir di <https://console.aws.amazon.com/push-notifications/>.
2. Pilih aplikasi yang berisi kredensi push Anda.
3. (Opsional) Untuk layanan Pemberitahuan Push Apple (APNs) hapus Aktifkan.
4. (Opsional) Untuk Firebase Cloud Messaging (FCM) hapus Aktifkan.
5. (Opsional) Untuk Baidu Cloud Push clear Enable.
6. (Opsional) Untuk Amazon Device Messaging clear Enable.
7. Pilih Simpan perubahan.

Mengirim pesan

AWS End User Messaging Push API dapat mengirim pemberitahuan push transaksional ke pengidentifikasi perangkat tertentu. Bagian ini berisi contoh kode lengkap yang dapat Anda gunakan untuk mengirim pemberitahuan push melalui AWS End User Messaging Push API dengan menggunakan file AWS SDK.

Anda dapat menggunakan contoh ini untuk mengirim pemberitahuan push melalui layanan pemberitahuan push apa pun yang didukung oleh AWS End User Messaging Push. Saat ini, AWS End User Messaging Push mendukung saluran berikut: Firebase Cloud Messaging (FCM), Layanan Pemberitahuan Push Apple (APNs), Baidu Cloud Push, dan Amazon Device Messaging (ADM).

Untuk contoh kode lainnya pada titik akhir, segmen, dan saluran, lihat [Contoh kode](#).

Note

Saat Anda mengirim notifikasi push melalui layanan Firebase Cloud Messaging (FCM), gunakan nama layanan GCM dalam panggilan Anda ke AWS End User Messaging Push API. Layanan Google Cloud Messaging (GCM) dihentikan oleh Google pada 10 April 2018. Namun, AWS End User Messaging Push API menggunakan nama GCM layanan untuk pesan yang dikirim melalui FCM layanan untuk menjaga kompatibilitas dengan API kode yang ditulis sebelum penghentian layanan. GCM

GCM (AWS CLI)

Contoh berikut menggunakan [send-messages](#) untuk mengirim notifikasi GCM Push dengan. AWS CLI Ganti *token* dengan token unik perangkat dan *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda.

```
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"
  }
}
}
}

```

Contoh berikut menggunakan [kirim pesan](#) untuk mengirim pemberitahuan GCM Push, menggunakan semua kunci lama, dengan tombol. AWS CLI Ganti *token* dengan token unik perangkat dan *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda.

```

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 \"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 }, \n \"data\": {\"message\": \"hello in data\"} }",
      "TimeToLive" : 309744
    }
  },
  "Addresses": {
    "token": {
      "ChannelType": "GCM"
    }
  }
}'
\ --region us-east-1

```

Contoh berikut menggunakan [send-messages](#) untuk mengirim pemberitahuan GCM Push dengan payload FCMv1 pesan menggunakan AWS CLI Ganti *token* dengan token unik perangkat dan *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda.

```
aws pinpoint send-messages \
--application-id 6a2dafd84bec449ea75fb773f4c41fa1 \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\n \"fcmV1Message\": \n {\n \"message\" :{\n \"notification
\n: {\n \"title\": \"string\", \n \"body\": \"string\"\n }, \n \"android\": {\n
\n \"priority\": \"high\", \n \"notification\": {\n \"title\": \"string\", \n \"body
\n: \"string\", \n \"icon\": \"string\", \n \"color\": \"string\", \n \"sound\":
\n \"string\", \n \"tag\": \"string\", \n \"click_action\": \"string\", \n \"body_loc_key
\n: \"string\", \n \"body_loc_args\": [\n \"string\"\n ], \n \"title_loc_key
\n: \"string\", \n \"title_loc_args\": [\n \"string\"\n ], \n \"channel_id\":
\n \"string\", \n \"ticker\": \"string\", \n \"sticky\": true, \n \"event_time\":
\n \"2024-02-06T22:11:55Z\", \n \"local_only\": true, \n \"notification_priority\":
\n \"PRIORITY_UNSPECIFIED\", \n \"default_sound\": false, \n \"default_vibrate_timings
\n: true, \n \"default_light_settings\": false, \n \"vibrate_timings\": [\n \"22s
\n\n ], \n \"visibility\": \"VISIBILITY_UNSPECIFIED\", \n \"notification_count\": 5,
\n \"light_settings\": {\n \"color\": {\n \"red\": 1, \n \"green\": 2, \n \"blue\":
\n 3, \n \"alpha\": 6\n }, \n \"light_on_duration\": \"112s\", \n \"light_off_duration
\n: \"1123s\"\n }, \n \"image\": \"string\"\n }, \n \"data\": {\n \"dataKey1\":
\n \"priority message\", \n \"data_key_3\": \"priority message\", \n \"dataKey2\":
\n \"priority message\", \n \"data_key_5\": \"priority message\"\n }, \n \"ttl\":
\n \"10023.32s\"\n }, \n \"apns\": {\n \"payload\": {\n \"aps\": {\n \"alert\": {\n
\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\", \n \"subtitle-loc-args\": [\n \"string\"\n ], \n \"loc-args\": [\n \"string
\n\n ], \n \"loc-key\": \"string\", \n \"title\": \"string\", \n \"body\": \"string
\n\n }, \n \"thread-id\": \"string\", \n \"category\": \"string\", \n \"content-
available\": 1, \n \"mutable-content\": 1, \n \"target-content-id\": \"string\", \n
\n \"interruption-level\": \"string\", \n \"relevance-score\": 25, \n \"filter-criteria
\n: \"string\", \n \"stale-date\": 6483, \n \"content-state\": {}, \n \"timestamp\":
\n 673634, \n \"dismissal-date\": 4, \n \"attributes-type\": \"string\", \n \"attributes
\n: {}\", \n \"sound\": \"string\", \n \"badge\": 5\n }\n }\n }, \n \"webpush\": {\n
\n \"notification\": {\n \"permission\": \"granted\", \n \"maxActions\": 2, \n \"actions
\n: [\n \"title\"\n ], \n \"badge\": \"URL\", \n \"body\": \"Hello\", \n \"data\": {\n
\n \"hello\": \"hey\"\n }, \n \"dir\": \"auto\", \n \"icon\": \"icon\", \n \"image\":
\n \"image\", \n \"lang\": \"string\", \n \"renotify\": false, \n \"requireInteraction\":
\n true, \n \"silent\": false, \n \"tag\": \"tag\", \n \"timestamp\": 1707259524964, \n
```

```

\"title\": \"hello\",\\n \\\"vibrate\": [\\n 100,\\n 200,\\n 300\\n ]\\n },\\n \\\"data\": {\\n
\\\"data1\": \\\"priority message\\\",\\n \\\"data2\": \\\"priority message\\\",\\n \\\"data12\":
\\\"priority message\\\",\\n \\\"data3\": \\\"priority message\\\"\\n }\\n },\\n \\\"data\": {\\n
\\\"data7\": \\\"priority message\\\",\\n \\\"data5\": \\\"priority message\\\",\\n \\\"data8\":
\\\"priority message\\\",\\n \\\"data9\": \\\"priority message\\\"\\n }\\n }\\n \\n}\\n}\\n\",
  \"TimeToLive\" : 309744
}
},
\"Addresses\": {
  token: {
    \"ChannelType\": \"GCM\"
  }
}
}'
\\ --region us-east-1

```

jika menggunakan `ImageUrl` bidang untuk GCM, pinpoint mengirimkan bidang sebagai pemberitahuan data, dengan kunci `notification.imageUrl`, yang dapat mencegah gambar dirender keluar dari kotak. Harap gunakan `RawContent` atau tambahkan penanganan kunci data seperti mengintegrasikan aplikasi Anda. AWS Amplify

Safari (AWS CLI)

Anda dapat menggunakan AWS End User Messaging Push untuk mengirim pesan ke komputer macOS yang menggunakan browser web Safari Apple. Untuk mengirim pesan ke browser Safari, Anda harus menentukan konten pesan mentah, dan Anda harus menyertakan atribut tertentu dalam payload pesan. Anda dapat melakukannya dengan [membuat templat pemberitahuan push dengan payload pesan mentah](#), atau dengan menentukan konten pesan mentah secara langsung dalam pesan [kampanye](#), di Panduan Pengguna Amazon Pinpoint.

Note

Atribut khusus ini diperlukan untuk mengirim ke laptop macOS dan komputer desktop yang menggunakan browser web Safari. Tidak diperlukan untuk mengirim ke perangkat iOS seperti iPhones dan iPads.

Untuk mengirim pesan ke browser web Safari, Anda harus menentukan payload pesan mentah. Payload pesan mentah harus menyertakan `url-args` array dalam `aps` objek. `url-argsArray` diperlukan untuk mengirim pemberitahuan push ke browser web Safari. Namun, array dapat diterima untuk berisi satu elemen kosong.

Contoh berikut menggunakan [send-messages](#) untuk mengirim notifikasi ke browser web Safari dengan file. AWS CLI Ganti *token* dengan token unik perangkat dan *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda.

```
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
```

Untuk informasi selengkapnya tentang pemberitahuan push Safari, lihat [Mengonfigurasi Pemberitahuan Push Safari](#) di situs web Pengembang Apple.

APNS (AWS CLI)

Contoh berikut menggunakan [send-messages](#) untuk mengirim notifikasi APNS Push dengan. AWS CLI Ganti *token* dengan token unik perangkat, *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda, dan *GAME_INVITATION* dengan pengenal unik.

```
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)

Gunakan contoh ini untuk mengirim pemberitahuan push dengan menggunakan AWS SDK for JavaScript di Node.js. Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi SDK for JavaScript di Node.js.

Contoh ini juga mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Menyetel kredensial](#) di AWS SDK 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

Gunakan contoh ini untuk mengirim pemberitahuan push dengan menggunakan AWS SDK for Python (Boto3). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi SDK untuk Python (Boto3).

Contoh ini juga mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat [Kredensial di Referensi](#) untuk AWS SDKPython (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()
```

Sumber daya tambahan

- Untuk informasi selengkapnya tentang templat saluran Push, lihat [Membuat templat pemberitahuan push](#) di Panduan Pengguna Amazon Pinpoint.

Menerima pemberitahuan push di aplikasi Anda

Topik berikut menjelaskan cara memodifikasi aplikasi Swift, Android, React Native, atau Flutter Anda sehingga menerima pemberitahuan push.

Topik

- [Menyiapkan Pemberitahuan Push Swift](#)
- [Menyiapkan pemberitahuan push Android](#)
- [Menyiapkan Pemberitahuan Flutter Push](#)
- [Menyiapkan Pemberitahuan Push React Native](#)
- [Buat aplikasi di AWS End User Messaging Push](#)
- [Menangani pemberitahuan push](#)

Menyiapkan Pemberitahuan Push Swift

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

Bekerja dengan APNs token

Sebagai praktik terbaik, Anda harus mengembangkan aplikasi sehingga token perangkat pelanggan dibuat ulang saat aplikasi diinstal ulang.

Jika penerima memutakhirkan perangkatnya ke iOS versi utama yang baru (misalnya, dari iOS 12 ke iOS 13), dan kemudian menginstal ulang aplikasi Anda, aplikasi akan menghasilkan token baru. Jika aplikasi Anda tidak me-refresh token, token yang lebih lama akan digunakan untuk mengirim notifikasi. Akibatnya, layanan Pemberitahuan Push Apple (APNs) menolak notifikasi, karena token sekarang tidak valid. Saat Anda mencoba mengirim notifikasi, Anda menerima pemberitahuan kegagalan pesan dari APNs.

Menyiapkan pemberitahuan push Android

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

Menyiapkan Pemberitahuan Flutter Push

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

Menyiapkan Pemberitahuan Push React Native

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

Buat aplikasi di AWS End User Messaging Push

Untuk mulai mengirim pemberitahuan push di AWS End User Messaging Push, Anda harus membuat aplikasi. Selanjutnya, Anda harus mengaktifkan saluran pemberitahuan push yang ingin Anda gunakan dengan memberikan kredensi yang sesuai.

Anda dapat membuat aplikasi baru dan mengatur saluran notifikasi push dengan menggunakan konsol AWS End User Messaging Push. Untuk informasi selengkapnya, lihat [Membuat aplikasi dan mengaktifkan saluran push](#).

Anda juga dapat membuat dan mengatur aplikasi dengan menggunakan [API](#), an [AWS SDK](#), atau [AWS Command Line Interface](#) (AWS CLI). Untuk membuat aplikasi, gunakan Apps sumber daya. Untuk mengonfigurasi saluran pemberitahuan push, gunakan sumber daya berikut:

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

Menangani pemberitahuan push

Setelah Anda mendapatkan kredensi yang diperlukan untuk mengirim pemberitahuan push, Anda dapat memperbarui aplikasi Anda sehingga mereka dapat menerima pemberitahuan push. Untuk informasi selengkapnya, lihat [Pemberitahuan push—Memulai dokumentasi](#). AWS Amplify

Menghapus aplikasi

Prosedur ini menghapus aplikasi dari akun Anda dan semua sumber daya dalam aplikasi.

Kontekstual

Aplikasi

Aplikasi adalah wadah penyimpanan untuk semua pengaturan AWS End User Messaging Push Anda. Aplikasi ini juga menyimpan saluran Amazon Pinpoint, kampanye, dan pengaturan perjalanan Anda.

Prosedur

1. Buka konsol Push Pesan Pengguna AWS Akhir di <https://console.aws.amazon.com/push-notifications/>.
2. Pilih aplikasi dan kemudian pilih Hapus.
3. Di jendela Hapus aplikasi masukkan **delete** dan kemudian pilih Hapus.

Important

Setiap saluran, kampanye, perjalanan, atau segmen Amazon Pinpoint juga akan dihapus.

Praktik terbaik

Bahkan ketika Anda memiliki minat terbaik pelanggan, Anda mungkin masih menghadapi situasi yang berdampak pada kemampuan pengiriman pesan Anda. Bagian berikut berisi rekomendasi untuk membantu memastikan bahwa komunikasi push Anda menjangkau audiens yang dituju.

Mengirim notifikasi push volume tinggi

Sebelum Anda mengirim notifikasi push volume tinggi, pastikan akun Anda dikonfigurasi untuk mendukung persyaratan throughput Anda. Secara default, semua akun dikonfigurasi untuk mengirim 25.000 pesan per detik. Jika Anda harus dapat mengirim lebih dari 25.000 pesan dalam satu detik, Anda dapat meminta peningkatan kuota. Untuk informasi selengkapnya, lihat [Kuota untuk Push Pesan Pengguna AWS Akhir](#).

Pastikan akun Anda dikonfigurasi dengan benar dengan kredensi untuk masing-masing penyedia pemberitahuan push yang akan Anda gunakan, seperti FCM atau APNs

Akhirnya, rancang cara untuk menangani pengecualian. Setiap layanan pemberitahuan push menyediakan pesan pengecualian yang berbeda. Untuk pengiriman transaksional, Anda menerima kode status utama 200 untuk API panggilan, dengan kode status per titik akhir 400 kegagalan permanen jika token platform yang sesuai (misalnya, FCM) atau sertifikat (misalnya, APN) ditentukan tidak valid selama pengiriman pesan.

Keamanan di AWS End User Messaging Push

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program AWS Kepatuhan Program AWS Kepatuhan](#) . Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS End User Messaging Push, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan](#) .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS End User Messaging Push. Topik berikut menunjukkan cara mengonfigurasi AWS End User Messaging Push untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Push Pesan Pengguna AWS Akhir Anda.

Topik

- [Perlindungan data di AWS End User Messaging Push](#)
- [Manajemen identitas dan akses untuk AWS End User Messaging Push](#)
- [Validasi kepatuhan untuk AWS End User Messaging Push](#)
- [Ketahanan dalam Push AWS Pesan Pengguna Akhir](#)
- [Keamanan Infrastruktur di AWS End User Messaging Push](#)
- [Konfigurasi dan analisis kerentanan](#)
- [Praktik terbaik keamanan](#)

Perlindungan data di AWS End User Messaging Push

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS End User Messaging Push. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk AWS layanan yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat [Privasi Data FAQ](#). Untuk informasi tentang perlindungan data di Eropa, lihat [Model Tanggung Jawab AWS Bersama dan](#) posting GDPR blog di Blog AWS Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan otentikasi multi-faktor (MFA) dengan setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya AWS layanan.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan FIPS 140-2 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atau, gunakan titik akhir API FIPS Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS End User Messaging Push atau lainnya AWS layanan menggunakan konsol, API, AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Jika Anda memberikan URL ke server eksternal,

kami sangat menyarankan agar Anda tidak menyertakan informasi kredensial dalam URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi data

AWS Data End User Messaging Push dienkripsi saat transit dan saat istirahat. Ketika Anda mengirimkan data ke AWS End User Messaging Push, itu mengenkripsi data saat menerima dan menyimpannya. Saat Anda mengambil data dari AWS End User Messaging Push, data akan ditransmisikan kepada Anda dengan menggunakan protokol keamanan saat ini.

Enkripsi diam

AWS End User Messaging Push mengenkripsi semua data yang disimpan untuk Anda. Ini termasuk data konfigurasi, data pengguna dan titik akhir, data analitik, dan data apa pun yang Anda tambahkan atau impor ke AWS End User Messaging Push. Untuk mengenkripsi data Anda, AWS End User Messaging Push menggunakan kunci internal AWS Key Management Service (AWS KMS) yang dimiliki dan dikelola oleh layanan atas nama Anda. Kami memutar tombol-tombol ini secara teratur. Untuk selengkapnya AWS KMS, lihat [Panduan AWS Key Management Service Pengembang](#).

Enkripsi bergerak

AWS End User Messaging Push menggunakan HTTPS dan Transport Layer Security (TLS) 1.2 atau yang lebih baru untuk berkomunikasi dengan klien dan aplikasi Anda. Untuk berkomunikasi dengan AWS layanan lain, AWS End User Messaging Push menggunakan HTTPS dan TLS 1.2. Selain itu, saat Anda membuat dan mengelola sumber daya AWS End User Messaging Push dengan menggunakan konsol AWS SDK, atau AWS Command Line Interface, semua komunikasi diamankan menggunakan HTTPS dan TLS 1.2.

Manajemen kunci

Untuk mengenkripsi data Push Pesan Pengguna AWS Akhir Anda, AWS End User Messaging Push menggunakan AWS KMS kunci internal yang dimiliki dan dikelola oleh layanan atas nama Anda. Kami memutar tombol-tombol ini secara teratur. Anda tidak dapat menyediakan dan menggunakan kunci Anda sendiri AWS KMS atau lainnya untuk mengenkripsi data yang Anda simpan di AWS End User Messaging Push.

Privasi lalu lintas antar jaringan

Privasi lalu lintas internetwork mengacu pada pengamanan koneksi dan lalu lintas antara AWS End User Messaging Push dan klien dan aplikasi lokal Anda, dan antara AWS End User Messaging Push dan AWS sumber daya lain di Wilayah yang sama. AWS Fitur dan praktik berikut dapat membantu Anda memastikan privasi lalu lintas internetwork untuk AWS End User Messaging Push.

Lalu lintas antara AWS End User Messaging Push dan klien dan aplikasi lokal

Untuk membuat koneksi pribadi antara AWS End User Messaging Push dan klien dan aplikasi di jaringan lokal Anda, Anda dapat menggunakannya AWS Direct Connect. Ini memungkinkan Anda untuk menghubungkan jaringan Anda ke suatu AWS Direct Connect lokasi dengan menggunakan kabel Ethernet serat optik standar. Salah satu ujung kabel terhubung ke router Anda. Ujung lainnya terhubung ke AWS Direct Connect router. Untuk informasi selengkapnya, lihat [Apa itu AWS Direct Connect?](#) dalam AWS Direct Connect Panduan Pengguna.

Untuk membantu mengamankan akses ke AWS End User Messaging Push melalui publikasi APIs, kami sarankan Anda mematuhi persyaratan Push Pesan Pengguna AWS Akhir untuk API panggilan. AWS End User Messaging Push mengharuskan klien untuk menggunakan Transport Layer Security (TLS) 1.2 atau yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS), seperti Ephemeral Diffie-Hellman () atau Elliptic Curve Diffie-Hellman Ephemeral (). DHE ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan AWS Identity and Access Management (IAM) prinsipal untuk AWS akun Anda. Atau, Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Lalu lintas antara AWS End User Messaging Push dan AWS sumber daya lainnya

Untuk mengamankan komunikasi antara AWS End User Messaging Push dan AWS sumber daya lain di AWS Wilayah yang sama, AWS End User Messaging Push menggunakan HTTPS dan TLS 1.2 secara default.

Manajemen identitas dan akses untuk AWS End User Messaging Push

AWS Identity and Access Management (IAM) adalah AWS layanan yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. IAM administrator mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Push Pesan Pengguna AWS Akhir. IAM adalah AWS layanan yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Cara Kerja AWS End User Messaging Push IAM](#)
- [Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push](#)
- [Pemecahan Masalah AWS End User Messaging Push identitas dan akses](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di AWS End User Messaging Push.

Pengguna layanan — Jika Anda menggunakan layanan AWS End User Messaging Push untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur AWS End User Messaging Push untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS End User Messaging Push, lihat [Pemecahan Masalah AWS End User Messaging Push identitas dan akses](#).

Administrator layanan — Jika Anda bertanggung jawab atas sumber daya AWS End User Messaging Push di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS End User Messaging Push. Tugas Anda adalah menentukan fitur dan sumber daya AWS End User Messaging Push mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke IAM administrator Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini

untuk memahami konsep dasar IAM. Untuk mempelajari selengkapnya tentang cara perusahaan Anda dapat digunakan IAM dengan AWS End User Messaging Push, lihat [Cara Kerja AWS End User Messaging Push IAM](#).

IAM Administrator - Jika Anda seorang IAM administrator, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Push Pesan Pengguna AWS Akhir. Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir yang dapat Anda gunakan, lihat. IAM [Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push](#)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai IAM pengguna, atau dengan mengambil peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (Pusat IAM Identitas), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan IAM peran. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang menggunakan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani AWS API permintaan](#) di Panduan IAM Pengguna.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat [Autentikasi multi-faktor](#) di Panduan AWS IAM Identity Center Pengguna dan [Menggunakan otentikasi multi-faktor \(MFA\) AWS di](#) Panduan Pengguna. IAM

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua AWS layanan dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensi pengguna root](#) di IAMPanduan Pengguna.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS layanan dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses AWS layanan dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat IAM Identitas, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat IAM Identitas, lihat [Apa itu Pusat IAM Identitas?](#) dalam AWS IAM Identity Center User Guide.

Pengguna dan grup IAM

[IAMPengguna](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya mengandalkan kredensi sementara daripada membuat IAM pengguna yang memiliki kredensi jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan IAM pengguna, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya, lihat [Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensi jangka panjang](#) di IAMPanduan Pengguna.

[IAMGrup](#) adalah identitas yang menentukan kumpulan IAM pengguna. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup bernama IAMAdmins dan memberikan izin grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari lebih lanjut, lihat [Kapan membuat IAM pengguna \(bukan peran\)](#) di Panduan IAM Pengguna.

Peran IAM

[IAMPeran](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Ini mirip dengan IAM pengguna, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil IAM peran sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil AWS CLI atau AWS API operasi atau dengan menggunakan kustom URL. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat [Menggunakan IAM peran](#) di Panduan IAM Pengguna.

IAM peran dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengotentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) di Panduan IAM Pengguna. Jika Anda menggunakan Pusat IAM Identitas, Anda mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, Pusat IAM Identitas mengkorelasikan izin yang disetel ke peran. Untuk informasi tentang set izin, lihat [Set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin IAM pengguna sementara — IAM Pengguna atau peran dapat mengambil IAM peran untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Akses lintas akun — Anda dapat menggunakan IAM peran untuk memungkinkan seseorang (prinsipal tepercaya) di akun lain mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa AWS layanan, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai

- proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM](#) Panduan Pengguna. IAM
- Akses lintas layanan — Beberapa AWS layanan menggunakan fitur lain AWS layanan. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
 - Sesi akses teruskan (FAS) — Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama AWS layanan, dikombinasikan dengan permintaan AWS layanan untuk membuat permintaan ke layanan hilir. FAS permintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain AWS layanan atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat [Meneruskan sesi akses](#).
 - Peran layanan — Peran layanan adalah [IAM peran](#) yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAM Administrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke AWS layanan](#) dalam IAM Panduan Pengguna.
 - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke AWS layanan. Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Akun AWS dan dimiliki oleh layanan. IAM Administrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.
 - Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan IAM peran untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS API meminta. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan IAM peran untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon](#) di IAM Panduan Pengguna.

Untuk mempelajari apakah akan menggunakan IAM peran atau IAM pengguna, lihat [Kapan membuat IAM peran \(bukan pengguna\)](#) di Panduan IAM Pengguna.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai JSON dokumen. Untuk informasi selengkapnya tentang struktur dan isi dokumen JSON kebijakan, lihat [Ringkasan JSON kebijakan](#) di Panduan IAM Pengguna.

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

IAMkebijakan menentukan izin untuk tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasi. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan itu bisa mendapatkan informasi peran dari AWS Management Console, AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat IAM kebijakan di Panduan](#) Pengguna. IAM

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan sebaris, lihat [Memilih antara kebijakan terkelola dan kebijakan sebaris](#) di IAMPanduan Pengguna.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau AWS layanan

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACLs. Untuk mempelajari selengkapnya ACLs, lihat [Ikhtisar daftar kontrol akses \(ACL\)](#) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- **Batas izin** — Batas izin adalah fitur lanjutan tempat Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas (pengguna atau peran). IAM IAM Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batas izin, lihat [Batas izin untuk IAM entitas](#) di IAMPanduan Pengguna.
- **Kebijakan kontrol layanan (SCPs)** — SCPs adalah JSON kebijakan yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations

adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP Membatasi izin untuk entitas di akun anggota, termasuk masing-masing Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat [Cara SCPs kerja](#) di Panduan AWS Organizations Pengguna.

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan secara tegas dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) di Panduan IAM Pengguna.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan IAM Pengguna.

Cara Kerja AWS End User Messaging Push IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke AWS End User Messaging Push, pelajari IAM fitur apa saja yang tersedia untuk digunakan dengan AWS End User Messaging Push.

IAM fitur yang dapat Anda gunakan dengan AWS End User Messaging Push

IAM fitur	AWS Dukungan Push Pesan Pengguna Akhir
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya

IAM fitur	AWS Dukungan Push Pesan Pengguna Akhir
ACLs	Tidak
ABAC(tag dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin prinsipal	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja AWS End User Messaging Push dan AWS layanan lainnya dengan sebagian besar IAM fitur, lihat [AWS layanan yang berfungsi IAM](#) di Panduan IAM Pengguna.

Kebijakan berbasis identitas untuk AWS End User Messaging Push

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat IAM kebijakan di Panduan](#) Pengguna. IAM

Dengan kebijakan IAM berbasis identitas, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam JSON kebijakan, lihat [referensi elemen IAM JSON kebijakan](#) di Panduan IAM Pengguna.

Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir, lihat. [Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push](#)

Kebijakan berbasis sumber daya dalam AWS End User Messaging Push

Mendukung kebijakan berbasis sumber daya: Ya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau AWS layanan

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau IAM entitas di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, IAM administrator di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun IAM di](#) Panduan IAM Pengguna.

Tindakan kebijakan untuk Push Pesan Pengguna AWS Akhir

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

ActionElemen JSON kebijakan menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan AWS API operasi terkait. Ada beberapa pengecualian, seperti tindakan khusus izin yang tidak memiliki operasi yang cocok. API Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan Push Pesan Pengguna AWS Akhir, lihat [Tindakan yang Ditentukan oleh Push Pesan Pengguna AWS Akhir](#) di Referensi Otorisasi Layanan.

Tindakan kebijakan di AWS End User Messaging Push menggunakan awalan berikut sebelum tindakan:

```
mobiletargeting
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

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

Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir, lihat. [Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push](#)

Sumber daya kebijakan untuk Push Pesan Pengguna AWS Akhir

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen Resource JSON kebijakan menentukan objek atau objek yang tindakan tersebut berlaku. Pernyataan harus menyertakan elemen Resource atau NotResource. Sebagai praktik terbaik, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya Push Pesan Pengguna AWS Akhir dan jenisnya ARNs, lihat Sumber Daya yang [Ditentukan oleh Push Pesan Pengguna AWS Akhir](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang Ditentukan oleh Push Pesan Pengguna AWS Akhir](#).

Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir, lihat. [Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push](#)

Kunci kondisi kebijakan untuk AWS End User Messaging Push

Mendukung kunci kondisi kebijakan khusus layanan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin IAM pengguna untuk mengakses sumber daya hanya jika ditandai dengan nama IAM pengguna mereka. Untuk informasi selengkapnya, lihat [elemen IAM kebijakan: variabel dan tag](#) di Panduan IAM Pengguna.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan IAM Pengguna.

Untuk melihat daftar tombol kondisi Push Pesan Pengguna AWS Akhir, lihat [Kunci Kondisi untuk Push Pesan Pengguna AWS Akhir](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang Ditentukan oleh Push Pesan Pengguna AWS Akhir](#).

Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir, lihat. [Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push](#)

ACLs di Push Pesan Pengguna AWS Akhir

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

ABAC dengan AWS End User Messaging Push

Mendukung ABAC (tag dalam kebijakan): Sebagian

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke IAM entitas (pengguna atau peran) dan ke banyak AWS sumber daya. Menandai entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian Anda merancang ABAC kebijakan untuk mengizinkan operasi ketika tag prinsipal cocok dengan tag pada sumber daya yang mereka coba akses.

ABAC membantu dalam lingkungan yang berkembang pesat dan membantu dengan situasi di mana manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi lebih lanjut tentang ABAC, lihat [Apa itu ABAC?](#) dalam IAM User Guide. Untuk melihat tutorial dengan langkah-langkah persiapan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) di IAM Panduan Pengguna.

Menggunakan kredensial sementara dengan AWS End User Messaging Push

Mendukung kredensial sementara: Ya

Beberapa AWS layanan tidak berfungsi saat Anda masuk menggunakan kredensial sementara. Untuk informasi tambahan, termasuk yang AWS layanan bekerja dengan kredensial sementara, lihat [AWS layanan yang berfungsi IAM](#) di IAMPanduan Pengguna.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan link sign-on (SSO) tunggal perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang beralih peran, lihat [Beralih ke peran \(konsol\)](#) di Panduan IAM Pengguna.

Anda dapat secara manual membuat kredensial sementara menggunakan atau. AWS CLI AWS API Anda kemudian dapat menggunakan kredensial sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensi keamanan sementara](#) di. IAM

Izin utama lintas layanan untuk AWS End User Messaging Push

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama AWS layanan, dikombinasikan dengan permintaan AWS layanan untuk membuat permintaan ke layanan hilir. FAS permintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain AWS layanan atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat [Meneruskan sesi akses](#).

Peran layanan untuk AWS End User Messaging Push

Mendukung peran layanan: Ya

Peran layanan adalah [IAM peran](#) yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAM Administrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke AWS layanan](#) dalam IAMPanduan Pengguna.

⚠ Warning

Mengubah izin untuk peran layanan dapat merusak fungsionalitas Push Pesan Pengguna AWS Akhir. Edit peran layanan hanya jika AWS End User Messaging Push memberikan panduan untuk melakukannya.

Peran terkait layanan untuk AWS End User Messaging Push

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke AWS layanan Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. IAM Administrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan, lihat [AWS layanan yang berfungsi](#) dengannya. IAM Cari layanan dalam tabel yang memiliki Yes di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Push Pesan Pengguna AWS Akhir. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan IAM berbasis identitas menggunakan contoh dokumen kebijakan ini, lihat [Membuat JSON IAM kebijakan di Panduan](#) Pengguna. IAM

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Push Pesan Pengguna AWS Akhir, termasuk format ARNs untuk setiap jenis sumber daya, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk Push Pesan Pengguna AWS Akhir](#) di Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)

- [Menggunakan konsol Push Pesan Pengguna AWS Akhir](#)
- [Mengizinkan pengguna melihat izin mereka sendiri](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Push Pesan Pengguna AWS Akhir di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [kebijakan AWSAWS terkelola atau kebijakan terkelola untuk fungsi pekerjaan](#) di Panduan IAM Pengguna.
- Menerapkan izin hak istimewa paling sedikit — Saat Anda menetapkan izin dengan IAM kebijakan, berikan hanya izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat [Kebijakan dan izin IAM di IAM](#) Panduan Pengguna.
- Gunakan ketentuan dalam IAM kebijakan untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik AWS layanan, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat [elemen IAM JSON kebijakan: Kondisi](#) dalam Panduan IAM Pengguna.
- Gunakan IAM Access Analyzer untuk memvalidasi IAM kebijakan Anda guna memastikan izin yang aman dan fungsional — IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan mematuhi bahasa IAM kebijakan () JSON dan praktik terbaik. IAM IAMAccess Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [Validasi kebijakan IAM Access Analyzer](#) di IAMPanduan Pengguna.

- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan IAM pengguna atau pengguna root di Anda Akun AWS, aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA kapan API operasi dipanggil, tambahkan MFA kondisi ke kebijakan Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi API akses MFA yang dilindungi](#) di IAMPanduan Pengguna.

Untuk informasi selengkapnya tentang praktik terbaik diIAM, lihat [Praktik terbaik keamanan IAM di Panduan IAM Pengguna](#).

Menggunakan konsol Push Pesan Pengguna AWS Akhir

Untuk mengakses konsol AWS End User Messaging Push, Anda harus memiliki seperangkat izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Push Pesan Pengguna AWS Akhir di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang cocok dengan API operasi yang mereka coba lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol Push Pesan Pengguna AWS Akhir, lampirkan juga kebijakan AWSEndUserMessaging AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan izin ke pengguna](#) di Panduan IAM Pengguna.

```
{
  "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": [
        "*"
    ]
}
]
}
}
}

```

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan IAM pengguna melihat kebijakan sebaris dan terkelola yang dilampirkan pada identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau secara terprogram menggunakan atau. AWS CLI AWS API

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",

```

```
    "Action": [  
      "iam:GetGroupPolicy",  
      "iam:GetPolicyVersion",  
      "iam:GetPolicy",  
      "iam:ListAttachedGroupPolicies",  
      "iam:ListGroupPolicies",  
      "iam:ListPolicyVersions",  
      "iam:ListPolicies",  
      "iam:ListUsers"  
    ],  
    "Resource": "*"    
  }  
]  
}
```

Pemecahan Masalah AWS End User Messaging Push identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS End User Messaging Push dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di AWS End User Messaging Push](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Push Pesan Pengguna AWS Akhir saya](#)

Saya tidak berwenang untuk melakukan tindakan di AWS End User Messaging Push

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika `mateojackson` IAM pengguna mencoba menggunakan konsol untuk melihat detail tentang `my-example-widget` sumber daya fiksi tetapi tidak memiliki izin `mobiletargeting:GetWidget` fiksi.

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

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `mobiletargeting:GetWidget`.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan `iam:PassRole`

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke AWS End User Messaging Push.

Beberapa AWS layanan memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika IAM pengguna bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di AWS End User Messaging Push. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Push Pesan Pengguna AWS Akhir saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah AWS End User Messaging Push mendukung fitur-fitur ini, lihat [Cara Kerja AWS End User Messaging Push IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke IAM pengguna lain Akun AWS yang Anda miliki](#) di Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna yang diautentikasi secara eksternal \(federasi identitas\) di Panduan Pengguna IAM](#).
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM](#) Panduan Pengguna IAM.

Validasi kepatuhan untuk AWS End User Messaging Push

Untuk mempelajari apakah an AWS layanan berada dalam lingkup program kepatuhan tertentu, lihat [AWS layanan di Lingkup oleh Program Kepatuhan AWS layanan](#) dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#) .

Tanggung jawab kepatuhan Anda saat menggunakan AWS layanan ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk HIPAA Keamanan dan Kepatuhan di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat HIPAA aplikasi yang memenuhi syarat.

Note

Tidak semua AWS layanan HIPAA memenuhi syarat. Untuk informasi selengkapnya, lihat [Referensi Layanan yang HIPAA Memenuhi Syarat](#).

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan AWS layanan dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi ()). ISO
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini AWS layanan memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [Amazon GuardDuty](#) — Ini AWS layanan mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCIDSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- [AWS Audit Manager](#)Ini AWS layanan membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan dalam Push AWS Pesan Pengguna Akhir

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan

Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Selain infrastruktur AWS global, AWS End User Messaging Push menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda.

Keamanan Infrastruktur di AWS End User Messaging Push

Sebagai layanan terkelola, AWS End User Messaging Push dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper [Amazon Web Services: Overview of Security Processes](#).

Anda menggunakan API panggilan yang AWS dipublikasikan untuk mengakses AWS End User Messaging Push melalui jaringan. Klien harus mendukung Transport Layer Security (TLS) 1.2 atau yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS) seperti (Ephemeral Diffie-Hellman) atau DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan IAM prinsipal. Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Konfigurasi dan analisis kerentanan

Sebagai layanan terkelola, AWS End User Messaging Push dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam [whitepaper Amazon Web Services: Tinjauan proses keamanan](#). Ini berarti bahwa AWS mengelola dan melakukan tugas dan prosedur keamanan dasar untuk mengeras, menambal, memperbarui, dan memelihara infrastruktur dasar untuk akun dan sumber daya Anda. Prosedur ini telah ditinjau dan disertifikasi oleh pihak ketiga yang sesuai.

Praktik terbaik keamanan

Gunakan akun AWS Identity and Access Management (IAM) untuk mengontrol akses ke API operasi, terutama operasi yang membuat, memodifikasi, atau menghapus sumber daya. Untuk API, sumber daya tersebut termasuk proyek, kampanye, dan perjalanan.

- Buat pengguna individu untuk setiap orang yang mengelola sumber daya, termasuk Anda sendiri. Jangan gunakan kredensi AWS root untuk mengelola sumber daya.
- Beri setiap pengguna set izin minimum yang diperlukan untuk melakukan tugas-tugasnya.
- Gunakan IAM grup untuk mengelola izin secara efektif untuk beberapa pengguna.
- Putar IAM kredensial Anda secara teratur.

Untuk informasi lebih lanjut tentang keamanan, lihat [Keamanan di AWS End User Messaging Push](#). Untuk informasi selengkapnya IAM, lihat [AWS Identity and Access Management](#). Untuk informasi tentang praktik IAM terbaik, lihat [praktik IAM terbaik](#).

Memantau Push Pesan Pengguna AWS Akhir

Pemantauan adalah bagian penting untuk menjaga keandalan, ketersediaan, dan kinerja AWS End User Messaging Push dan AWS solusi Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton AWS End User Messaging Push, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak CPU penggunaan atau metrik lain dari EC2 instans Amazon Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari EC2 instans Amazon CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon CloudWatch Logs](#).
- Amazon EventBridge dapat digunakan untuk mengotomatiskan AWS layanan Anda dan merespons secara otomatis peristiwa sistem, seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Acara dari AWS layanan dikirimkan ke EventBridge dalam waktu dekat. Anda dapat menuliskan aturan sederhana untuk menunjukkan peristiwa mana yang sesuai kepentingan Anda, dan tindakan otomatis mana yang diambil ketika suatu peristiwa sesuai dengan suatu aturan. Untuk informasi selengkapnya, lihat [Panduan EventBridge Pengguna Amazon](#).
- AWS CloudTrail menangkap API panggilan dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS CloudTrail](#).

Memantau Push Pesan Pengguna AWS Akhir dengan Amazon CloudWatch

Anda dapat memantau AWS End User Messaging Push menggunakan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati real-time. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

Untuk daftar metrik dan dimensi, lihat [Memantau Amazon Pinpoint CloudWatch](#) dengan di Panduan Pengguna Amazon Pinpoint.

Logging AWS End User Messaging API Panggilan push menggunakan AWS CloudTrail

AWS End User Messaging Push terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS End User Messaging Push. CloudTrail menangkap semua API panggilan untuk AWS End User Messaging Push sebagai acara. Panggilan yang diambil termasuk panggilan dari konsol Push Pesan Pengguna AWS Akhir dan panggilan kode ke API operasi AWS End User Messaging Push. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk peristiwa untuk Push Pesan Pengguna AWS Akhir. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk AWS End User Messaging Push, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

AWS Informasi Push Pesan Pengguna Akhir di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di AWS End User Messaging Push, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh

acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat [Melihat peristiwa dengan Riwayat CloudTrail acara](#).

Untuk catatan peristiwa yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk AWS End User Messaging Push, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- [Gambaran umum untuk membuat jejak](#)
- [CloudTrail layanan dan integrasi yang didukung](#)
- [Mengkonfigurasi SNS notifikasi Amazon untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua tindakan Push Pesan Pengguna AWS Akhir dicatat oleh CloudTrail dan didokumentasikan dalam [APIReferensi Push Pesan Pengguna AWS Akhir](#). Misalnya, panggilan ke `getAdmChannel`, `updateApnsChannel` dan `getApnsVoipChannel` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan dibuat dengan root atau AWS Identity and Access Management (IAM) kredensial pengguna.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lebih lanjut, lihat [CloudTrail userIdentityelemen](#).

AWS Memahami entri file log Push Pesan Pengguna Akhir

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili

permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari API panggilan publik, sehingga tidak muncul dalam urutan tertentu.

Akses AWS End User Messaging Push menggunakan endpoint antarmuka ()AWS PrivateLink

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara Anda VPC dan AWS End User Messaging Push. Anda dapat mengakses AWS End User Messaging Push seolah-olah itu ada di AndaVPC, tanpa menggunakan gateway internet, NAT perangkat, VPN koneksi, atau AWS Direct Connect koneksi. Instans di Anda VPC tidak memerlukan alamat IP publik untuk mengakses AWS End User Messaging Push.

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, yang didukung oleh AWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditujukan untuk AWS End User Messaging Push.

Untuk informasi selengkapnya, lihat [Akses AWS layanan melalui AWS PrivateLink](#) di AWS PrivateLink Panduan.

Pertimbangan untuk Push Pesan Pengguna AWS Akhir

Sebelum Anda menyiapkan titik akhir antarmuka untuk AWS End User Messaging Push, tinjau [Pertimbangan](#) dalam Panduan.AWS PrivateLink

AWS End User Messaging Push mendukung melakukan panggilan ke semua API tindakannya melalui titik akhir antarmuka.

VPCkebijakan endpoint tidak didukung untuk AWS End User Messaging Push. Secara default, akses penuh ke AWS End User Messaging Push diizinkan melalui titik akhir antarmuka. Atau, Anda dapat mengaitkan grup keamanan dengan antarmuka jaringan titik akhir untuk mengontrol lalu lintas ke AWS End User Messaging Push melalui titik akhir antarmuka.

Buat titik akhir antarmuka untuk AWS End User Messaging Push

Anda dapat membuat titik AWS akhir antarmuka untuk End User Messaging Push menggunakan VPC konsol Amazon atau AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat [Membuat titik akhir antarmuka](#) di AWS PrivateLink Panduan.

Buat endpoint antarmuka untuk AWS End User Messaging Push menggunakan nama layanan berikut:

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

Jika Anda mengaktifkan private DNS untuk titik akhir antarmuka, Anda dapat membuat API permintaan ke AWS End User Messaging Push menggunakan DNS nama Regional default. Misalnya, `com.amazonaws.us-east-1.pinpoint`.

Buat kebijakan titik akhir untuk titik akhir antarmuka Anda

Kebijakan endpoint adalah IAM sumber daya yang dapat Anda lampirkan ke titik akhir antarmuka. Kebijakan endpoint default memungkinkan akses penuh ke AWS End User Messaging Push melalui titik akhir antarmuka. Untuk mengontrol akses yang diizinkan ke AWS End User Messaging Push dari AndaVPC, lampirkan kebijakan endpoint khusus ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (Akun AWS, IAM pengguna, dan IAM peran).
- Tindakan yang dapat dilakukan.
- Sumber daya untuk melakukan tindakan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke layanan menggunakan kebijakan titik akhir](#) di Panduan AWS PrivateLink .

Contoh: kebijakan VPC endpoint untuk tindakan AWS End User Messaging Push

Berikut ini adalah contoh kebijakan endpoint kustom. Saat Anda melampirkan kebijakan ini ke titik akhir antarmuka Anda, kebijakan ini akan memberikan akses ke tindakan Push Pesan Pengguna AWS Akhir yang terdaftar untuk semua prinsip di semua sumber daya.

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

```
]
}
```

Kuota untuk Push Pesan Pengguna AWS Akhir

Anda Akun AWS memiliki kuota default, sebelumnya disebut sebagai batas, untuk setiap layanan. AWS Kecuali dinyatakan lain, setiap kuota bersifat khusus per Wilayah. Anda dapat meminta peningkatan untuk beberapa kuota dan kuota lainnya tidak dapat ditingkatkan.

Untuk melihat kuota untuk AWS End User Messaging Push, buka konsol [Service Quotas](#). Di panel navigasi, pilih AWSlayanan dan pilih Amazon Pinpoint.

AWSAkun Anda memiliki kuota berikut yang terkait dengan AWS End User Messaging Push.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum notifikasi push yang dapat dikirim per detik dalam kampanye	25.000 notifikasi per detik	Ya, gunakan konsol Service Quotas
Amazon Device Messaging (ADM) ukuran payload pesan	6 KB per pesan	Tidak
Layanan Pemberitahuan Push Apple (APNs) ukuran payload pesan	4 KB per pesan	Tidak
APNsukuran payload pesan kotak pasir	4 KB per pesan	Tidak
Ukuran payload pesan Baidu Cloud Push	4 KB per pesan	Tidak
Ukuran payload pesan Firebase Cloud Messaging (FCM)	4 KB per pesan	Tidak

Riwayat dokumen untuk Panduan Pengguna Push Pesan Pengguna AWS Akhir

Tabel berikut menjelaskan rilis dokumentasi untuk AWS End User Messaging Push.

Perubahan	Deskripsi	Tanggal
Rilis awal	Rilis awal Panduan Pengguna Push Pesan Pengguna AWS Akhir	Juli 24, 2024

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.