



Panduan Pengguna

Izin Terverifikasi Amazon



Izin Terverifikasi Amazon: Panduan Pengguna

Copyright © 2026 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 merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu Izin Terverifikasi Amazon?	1
Otorisasi dalam Izin Terverifikasi	1
Bahasa kebijakan Cedar	2
Manfaat Izin Terverifikasi	2
Mempercepat pengembangan aplikasi	2
Aplikasi yang lebih aman	2
Fitur pengguna akhir	3
Layanan terkait	3
Mengakses Izin Terverifikasi	3
Harga untuk Izin Terverifikasi	5
Memulai dengan toko kebijakan	6
Prasyarat	7
Langkah 1: Buat toko PhotoFlash kebijakan	9
Langkah 2: Buat kebijakan	9
Langkah 3: Menguji toko kebijakan	10
Langkah 4: Bersihkan Sumber Daya	12
Merancang model otorisasi	13
Tidak ada model tunggal yang benar	14
Mengembalikan kesalahan	15
Fokus pada sumber daya	15
Pertimbangkan multi-tenancy	17
Membandingkan toko kebijakan bersama dan toko kebijakan per penyewa	18
Bagaimana memilih	19
Toko kebijakan	21
Membuat toko kebijakan	22
Membuat toko kebijakan menggunakan Rust	30
Toko kebijakan terkait API	35
Cara kerjanya	37
Pertimbangan-pertimbangan	39
Menambahkan ABAC	40
Pindah ke produksi	41
Pemecahan masalah	44
Menghapus toko kebijakan	47
Alias toko kebijakan	49

Properti alias toko kebijakan	49
Membuat alias toko kebijakan	52
Mengambil alias toko kebijakan	53
Mendapatkan alias toko kebijakan	54
Daftar alias toko kebijakan	54
Menghapus alias toko kebijakan	56
Menggunakan alias toko kebijakan	57
Menggunakan alias toko Kebijakan dalam Operasi	57
Menggunakan alias toko Kebijakan Di Seluruh Wilayah AWS	58
Mengendalikan akses	59
izin terverifikasi: CreatePolicyStoreAlias	59
izin terverifikasi: GetPolicyStoreAlias	60
izin terverifikasi: ListPolicyStoreAliases	60
izin terverifikasi: DeletePolicyStoreAlias	61
Membatasi Izin alias penyimpanan Kebijakan	61
Skema toko kebijakan	64
Skema pengeditan	66
Mode validasi kebijakan	69
Kebijakan	71
Membuat kebijakan statis	72
Mengedit kebijakan statis	75
.....	78
Evaluasi konteks contoh	80
Menguji kebijakan	85
Contoh kebijakan	88
Menggunakan notasi braknet untuk referensi atribut token	89
Menggunakan notasi titik untuk referensi atribut	89
Mencerminkan atribut token Amazon Cognito ID	90
Mencerminkan atribut token ID OIDC	90
Mencerminkan atribut token Amazon Cognito akses	91
Mencerminkan atribut token akses OIDC	91
Templat kebijakan dan kebijakan terkait templat	92
Membuat template kebijakan	93
Membuat kebijakan yang ditautkan templat	95
Mengedit templat kebijakan	97
Contoh kebijakan yang ditautkan templat	99

PhotoFlash contoh	99
DigitalPetStore contoh	101
TinyToDo contoh	101
Sumber identitas	102
Memilih penyedia identitas yang tepat	103
Bekerja dengan sumber Amazon Cognito identitas	103
Membuat sumber identitas	106
Mengedit sumber identitas	109
Memetakan token ke skema	112
Validasi klien dan audiens	122
Bekerja dengan sumber identitas OIDC	125
Membuat sumber identitas	127
Mengedit sumber identitas	130
Memetakan token ke skema	132
Validasi klien dan audiens	139
Integrasi	143
Menggunakan Express	143
Prasyarat	144
Menyiapkan integrasi	144
Mengkonfigurasi otorisasi	145
Menerapkan middleware otorisasi	148
Menguji integrasi	148
Pemecahan masalah	149
Langkah selanjutnya	149
Otorisasi permintaan	150
Operasi API	151
Model uji	152
Integrasi dengan aplikasi	154
Keamanan	157
Perlindungan data	157
Enkripsi data	159
Kunci yang dikelola pelanggan	159
Manajemen identitas dan akses	179
Audiens	180
Mengautentikasi dengan identitas	180
Mengelola akses menggunakan kebijakan	182

Cara kerja Izin Terverifikasi Amazon IAM	184
IAM kebijakan untuk Izin Terverifikasi	190
Contoh kebijakan berbasis identitas	193
AWS kebijakan terkelola	196
Pemecahan masalah	200
Validasi kepatuhan	202
Ketahanan	202
Memantau	203
CloudTrail log	203
Informasi Izin Terverifikasi di CloudTrail	203
Memahami entri file log Izin Terverifikasi	205
Bekerja dengan AWS CloudFormation	223
Izin dan CloudFormation templat terverifikasi	223
AWS Konstruksi CDK	224
Pelajari lebih lanjut tentang CloudFormation	224
Menggunakan AWS PrivateLink	225
Pertimbangan-pertimbangan	225
Membuat sebuah titik akhir antarmuka	225
Membuat kebijakan titik akhir	226
Kuota	228
Kuota untuk sumber daya	228
Contoh ukuran kebijakan terkait templat	230
Kuota untuk hierarki	231
Kuota untuk operasi per detik	233
Syarat & konsep	238
Model otorisasi	239
Permintaan otorisasi	239
Respon otorisasi	239
Kebijakan yang dipertimbangkan	239
Data konteks	240
Menentukan kebijakan	240
Data entitas	240
Izin, otorisasi, dan prinsip	240
Penegakan kebijakan	240
Toko kebijakan	241
Alias toko kebijakan	241

Nama kebijakan	241
Nama templat kebijakan	242
Kebijakan yang memuaskan	242
Perbedaan dengan Cedar	242
Definisi namespace	242
Dukungan template kebijakan	242
Dukungan skema	243
Definisi kelompok aksi	243
Pemformatan entitas	243
Batas panjang dan ukuran	248
Cedar v4 FAQ	250
Mengapa beberapa kebijakan, templat kebijakan, dan skema tidak kompatibel dengan Cedar 4?	250
Bagaimana cara mengetahui apakah toko kebijakan saya menggunakan Cedar 2 atau Cedar 4?	251
Bagaimana cara meningkatkan ke Cedar 4?	252
Dapatkah saya menurunkan versi toko polis saya dari Cedar 4 ke Cedar 2?	253
Mengapa saya menerima pesan kesalahan yang mengatakan toko kebijakan saya dikonfigurasi untuk Cedar 2?	253
Bagaimana cara membuat skema saya kompatibel dengan Cedar 4?	253
Bagaimana cara membuat kebijakan dan template saya kompatibel dengan Cedar 4?	255
Riwayat dokumen	256
.....	cclviii

Apa itu Izin Terverifikasi Amazon?

Izin Terverifikasi Amazon adalah layanan manajemen dan otorisasi izin yang dapat diskalakan dan berbutir halus untuk aplikasi khusus yang dibuat oleh Anda. Izin Terverifikasi memungkinkan pengembang Anda untuk membangun aplikasi aman lebih cepat dengan mengeksternalisasi otorisasi dan memusatkan manajemen dan administrasi kebijakan. Izin Terverifikasi menggunakan bahasa kebijakan Cedar untuk menentukan izin berbutir halus guna melindungi sumber daya aplikasi Anda.

Untuk panduan dan contoh menyiapkan titik keputusan kebijakan (PDP) menggunakan Izin Terverifikasi, lihat [Menerapkan PDP menggunakan Izin Terverifikasi Amazon dalam Panduan Preskriptif.AWS](#)

Topik

- [Otorisasi dalam Izin Terverifikasi](#)
- [Bahasa kebijakan Cedar](#)
- [Manfaat Izin Terverifikasi](#)
- [Layanan terkait](#)
- [Mengakses Izin Terverifikasi](#)
- [Harga untuk Izin Terverifikasi](#)

Otorisasi dalam Izin Terverifikasi

Izin Terverifikasi memberikan otorisasi dengan memverifikasi apakah prinsipal diizinkan untuk melakukan tindakan pada sumber daya dalam konteks tertentu dalam aplikasi Anda. Izin Terverifikasi menganggap bahwa prinsipal sebelumnya telah diidentifikasi dan diautentikasi melalui cara lain, seperti dengan menggunakan protokol seperti OpenID Connect, penyedia yang dihosting seperti, atau solusi otentikasi lainnya. Amazon Cognito Izin Terverifikasi bersifat agnostik di mana prinsipal dikelola dan bagaimana mereka diautentikasi.

Izin Terverifikasi adalah layanan yang memungkinkan pelanggan membuat, memelihara, dan menguji kebijakan di Konsol Manajemen AWS, secara terprogram menggunakan Izin Terverifikasi APIs, atau melalui infrastruktur sebagai solusi kode seperti. CloudFormation Izin dinyatakan menggunakan bahasa kebijakan Cedar. Aplikasi klien memanggil otorisasi APIs untuk mengevaluasi kebijakan Cedar yang disimpan dengan layanan dan memberikan keputusan akses apakah suatu tindakan diizinkan.

Bahasa kebijakan Cedar

Kebijakan otorisasi dalam Izin Terverifikasi ditulis dengan menggunakan bahasa kebijakan Cedar. Cedar adalah bahasa open source untuk menulis kebijakan otorisasi dan membuat keputusan otorisasi berdasarkan kebijakan tersebut. Saat Anda membuat aplikasi, Anda perlu memastikan bahwa hanya prinsipal resmi, pengguna manusia atau mesin, yang dapat mengakses aplikasi, dan hanya dapat melakukan apa yang diizinkan untuk mereka lakukan. Menggunakan Cedar, Anda dapat memisahkan logika bisnis Anda dari logika otorisasi. Dalam kode aplikasi Anda, Anda mengawali permintaan yang dibuat untuk operasi Anda dengan panggilan ke mesin otorisasi Cedar, menanyakan “Apakah permintaan ini diotorisasi?”. Kemudian, aplikasi dapat melakukan operasi yang diminta jika keputusannya “izinkan”, atau mengembalikan pesan kesalahan jika keputusannya “tolak”.

Izin Terverifikasi saat ini menggunakan Cedar versi 4.7.

Untuk informasi lebih lanjut tentang Cedar, lihat berikut ini:

- [Panduan Referensi bahasa kebijakan cedar](#)
- [Repositori cedar GitHub](#)

Manfaat Izin Terverifikasi

Mempercepat pengembangan aplikasi

Mempercepat pengembangan aplikasi dengan memisahkan otorisasi dari logika bisnis.

Izin Terverifikasi menyediakan integrasi dengan kerangka kerja pengembangan populer, membuatnya lebih mudah untuk menerapkan otorisasi dalam aplikasi Anda dengan sedikit perubahan kode. Integrasi ini memungkinkan Anda untuk fokus pada logika bisnis inti Anda sementara Izin Terverifikasi menangani keputusan otorisasi.

- Express.js — Integrasi berbasis middleware yang memungkinkan Anda melindungi titik akhir API di aplikasi Express Anda tanpa memodifikasi pengendali rute yang ada. Untuk informasi selengkapnya, lihat [the section called “Menggunakan Express”](#).

Aplikasi yang lebih aman

Izin Terverifikasi memungkinkan pengembang untuk membangun aplikasi yang lebih aman.

Fitur pengguna akhir

Izin Terverifikasi memungkinkan Anda menghadirkan fitur pengguna akhir yang lebih kaya untuk pengelolaan izin.

Layanan terkait

- Amazon Cognito — Amazon Cognito adalah platform identitas untuk aplikasi web dan seluler. Ini adalah direktori pengguna, server otentikasi, dan layanan otorisasi untuk token akses OAuth 2.0 dan AWS kredensial. Saat membuat toko kebijakan, Anda memiliki opsi untuk membangun prinsipal dan grup dari kumpulan pengguna. Amazon Cognito Untuk informasi selengkapnya, lihat [Panduan Developer Amazon Cognito](#) .
- Amazon API Gateway — Amazon API Gateway adalah AWS layanan untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket APIs pada skala apa pun. Saat membuat penyimpanan kebijakan, Anda memiliki opsi untuk membuat tindakan dan sumber daya dari API API Gateway. Untuk informasi selengkapnya API Gateway, lihat [Panduan API Gateway Pengembang](#).
- AWS IAM Identity Center— Dengan IAM Identity Center, Anda dapat mengelola keamanan masuk untuk identitas tenaga kerja Anda, juga dikenal sebagai pengguna tenaga kerja. IAM Identity Center menyediakan satu tempat di mana Anda dapat membuat atau menghubungkan pengguna tenaga kerja dan mengelola akses mereka secara terpusat di semua aplikasi dan aplikasi mereka Akun AWS . Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS IAM Identity Center](#).

Mengakses Izin Terverifikasi

Anda dapat bekerja dengan Izin Terverifikasi Amazon dengan salah satu cara berikut.

Konsol Manajemen AWS

Konsol adalah antarmuka berbasis browser untuk mengelola Izin dan sumber daya Terverifikasi. AWS Untuk informasi selengkapnya tentang mengakses Izin Terverifikasi melalui konsol, lihat [Cara masuk di AWS Sign-In](#) Panduan Pengguna. AWS

- [Konsol Izin Terverifikasi Amazon](#)

AWS Alat Baris Perintah

Anda dapat menggunakan alat baris AWS perintah untuk mengeluarkan perintah di baris perintah sistem Anda untuk melakukan Izin dan AWS tugas Terverifikasi. Menggunakan baris perintah dapat lebih cepat dan lebih nyaman dibandingkan konsol. Alat baris perintah juga berguna jika Anda ingin membangun skrip yang melakukan tugas AWS .

AWS menyediakan dua set alat baris perintah: [AWS Command Line Interface](#)(AWS CLI) dan [AWS Tools for Windows PowerShell](#). Untuk informasi tentang menginstal dan menggunakan AWS CLI, lihat [Panduan AWS Command Line Interface Pengguna](#). Untuk informasi tentang menginstal dan menggunakan Alat untuk Windows PowerShell, lihat [Panduan AWS Tools for PowerShell Pengguna](#).

- [izin terverifikasi di Referensi Perintah](#) AWS CLI
- [Izin Terverifikasi Amazon](#) di AWS Tools for Windows PowerShell

AWS SDKs

AWS menyediakan SDKs (kit pengembangan perangkat lunak) yang terdiri dari pustaka dan kode sampel untuk berbagai bahasa dan platform pemrograman (Java, Python, Ruby, .NET, iOS, Android, dll.). SDKs Menyediakan cara mudah untuk membuat akses terprogram ke Izin Terverifikasi dan. AWS Misalnya, SDKs mengurus tugas-tugas seperti menandatangani permintaan secara kriptografis, mengelola kesalahan, dan mencoba ulang permintaan secara otomatis.

Untuk mempelajari lebih lanjut dan mengunduh AWS SDKs, lihat [Alat untuk Amazon Web Services](#).

Berikut ini adalah tautan ke dokumentasi untuk sumber daya Izin Terverifikasi di berbagai AWS SDKs sumber.

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto\)](#)
- [AWS SDK for Ruby](#)

- [AWS SDK for Rust](#)

AWS Konstruksi CDK

AWS Cloud Development Kit (AWS CDK) Ini adalah kerangka pengembangan perangkat lunak open-source untuk mendefinisikan infrastruktur cloud dalam kode dan menyediakannya. CloudFormation Konstruksi, atau komponen cloud yang dapat digunakan kembali, dapat digunakan untuk membuat CloudFormation templat. Templat ini kemudian dapat digunakan untuk menyebarkan infrastruktur cloud Anda.

Untuk mempelajari lebih lanjut dan mengunduh AWS CDK, lihat [AWS Cloud Development Kit](#).

Berikut ini adalah tautan ke dokumentasi untuk AWS CDK sumber daya Izin Terverifikasi, seperti konstruksi.

- [Izin Terverifikasi Amazon Konstruksi L2 CDK](#)

API Izin Terverifikasi

Anda dapat mengakses Izin Terverifikasi dan AWS secara terprogram menggunakan API Izin Terverifikasi, yang memungkinkan Anda mengeluarkan permintaan HTTPS langsung ke layanan. Saat Anda menggunakan API, Anda harus menyertakan kode untuk menandatangani permintaan secara digital menggunakan kredensial Anda.

- [Panduan Referensi API Izin Terverifikasi Amazon](#)

Harga untuk Izin Terverifikasi

Izin Terverifikasi memberikan harga berjenjang berdasarkan jumlah permintaan otorisasi per bulan yang dibuat oleh aplikasi Anda untuk Izin Terverifikasi. Ada juga harga untuk tindakan manajemen kebijakan berdasarkan jumlah permintaan API kebijakan cURL (URL klien) per bulan yang dibuat oleh aplikasi Anda ke Izin Terverifikasi.

Untuk daftar lengkap biaya dan harga untuk Izin Terverifikasi, lihat harga Izin [Terverifikasi Amazon](#).

Untuk melihat tagihan Anda, buka Dasbor Manajemen Penagihan dan Biaya di [konsol AWS Manajemen Penagihan dan Biaya](#). Tagihan Anda berisi tautan ke laporan penggunaan yang memberikan detail tentang tagihan Anda. Untuk mempelajari selengkapnya tentang Akun AWS penagihan, lihat [Panduan AWS Billing Pengguna](#).

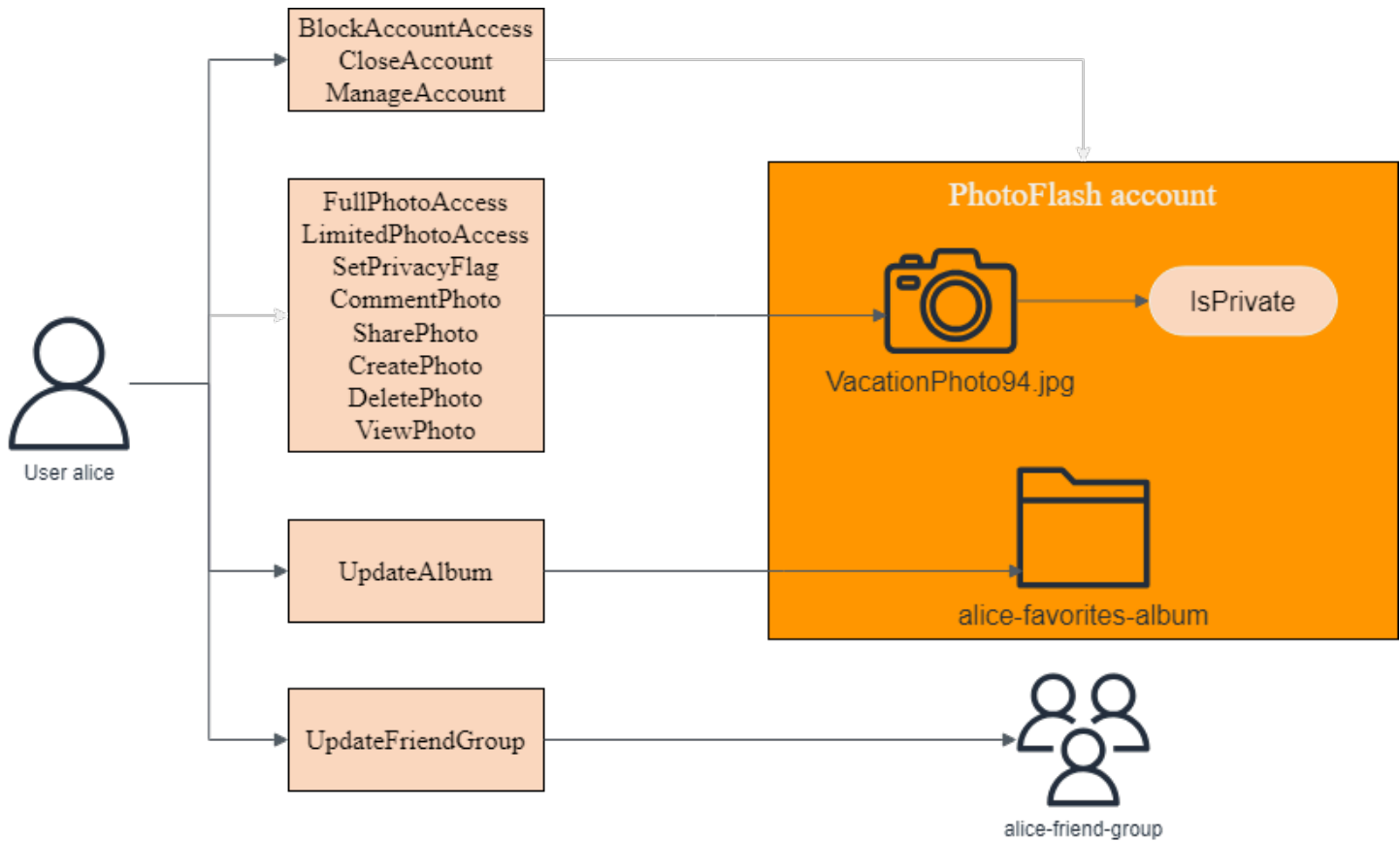
Jika Anda memiliki pertanyaan tentang AWS penagihan, akun, dan acara, [hubungi Dukungan](#).

Buat toko kebijakan Izin Terverifikasi Amazon pertama Anda

Untuk tutorial ini, mari kita asumsikan Anda adalah pengembang aplikasi berbagi foto dan Anda mencari cara untuk mengontrol tindakan apa yang dapat dilakukan pengguna aplikasi. Anda ingin mengontrol siapa yang dapat menambahkan, menghapus, atau melihat foto dan album foto. Anda juga ingin mengontrol tindakan apa yang dapat dilakukan pengguna di akun mereka. Bisakah mereka mengelola akun mereka, bagaimana dengan akun teman? Untuk mengontrol tindakan ini, Anda akan membuat kebijakan yang mengizinkan atau melarang tindakan ini berdasarkan identitas pengguna. Izin Terverifikasi menawarkan [toko kebijakan](#), atau wadah, untuk menampung kebijakan ini.

Dalam tutorial ini kita akan berjalan melalui pembuatan toko kebijakan sampel menggunakan konsol Izin Terverifikasi Amazon. Konsol menawarkan beberapa contoh opsi penyimpanan kebijakan dan kami akan membuat toko PhotoFlashkebijakan. Toko kebijakan ini memungkinkan prinsipal, seperti pengguna, untuk melakukan tindakan, seperti berbagi, pada sumber daya, seperti foto atau album.

Diagram berikut menggambarkan hubungan antara kepala sekolah, `User::alice`, dan tindakan yang dapat dia ambil pada berbagai sumber daya, yaitu PhotoFlash akunya, `VactionPhoto94.jpg` file, album foto `alice-favorites-album`, dan grup `alice-friend-group` pengguna.



Sekarang setelah Anda memahami toko PhotoFlash kebijakan, mari buat toko kebijakan dan jelajahi.

Prasyarat

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 menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS 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 mengirim Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk <https://aws.amazon.com/ke/> 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 Anda Pengguna root akun AWS

1. Masuk ke [Konsol Manajemen AWS](#) 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. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

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

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

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

2. Di Pusat Identitas IAM, 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 Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

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

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, 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.

Langkah 1: Buat toko PhotoFlash kebijakan

Dalam prosedur berikut, Anda akan membuat toko PhotoFlashkebijakan menggunakan AWS konsol.

Untuk membuat toko PhotoFlash kebijakan

1. Di [konsol Izin Terverifikasi](#), pilih Buat toko kebijakan baru.
2. Untuk opsi Memulai, pilih Mulai dari toko kebijakan sampel.
3. Untuk proyek Sampel, pilih PhotoFlash.
4. Pilih Buat toko kebijakan.

Setelah Anda melihat pesan “Toko kebijakan yang dibuat dan dikonfigurasi,” pilih Buka ikhtisar untuk menjelajahi toko kebijakan Anda.

Langkah 2: Buat kebijakan

Saat Anda membuat penyimpanan kebijakan, kebijakan default dibuat yang memungkinkan pengguna memiliki kontrol penuh atas akun mereka sendiri. Ini adalah kebijakan yang berguna, tetapi untuk tujuan kita, mari buat kebijakan yang lebih ketat untuk mengeksplorasi nuansa Izin

Terverifikasi. Jika Anda ingat diagram yang kita lihat sebelumnya dalam tutorial, kami memiliki kepala sekolah, `User::alice`, yang dapat melakukan tindakan, `UpdateAlbum`, pada sumber daya, `alice-favorites-album`. Mari tambahkan kebijakan yang akan memungkinkan Alice, dan hanya Alice, untuk mengelola album ini.

Untuk membuat kebijakan

1. Di [konsol Izin Terverifikasi](#), pilih penyimpanan kebijakan yang Anda buat di langkah 1.
2. Di navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan dan kemudian pilih Buat kebijakan statis.
4. Untuk efek Kebijakan, pilih Izin.
5. Untuk cakupan Prinsipal, pilih Prinsip khusus, lalu untuk Tentukan jenis entitas, pilih PhotoFlash: :Pengguna, dan untuk Tentukan pengenal entitas, masukkan. **alice**
6. Untuk cakupan Sumber daya, pilih Sumber daya spesifik, lalu untuk Tentukan jenis entitas, pilih PhotoFlash: :Album, dan untuk Tentukan pengenal entitas, masukkan. **alice-favorites-album**
7. Untuk cakupan Tindakan, pilih Kumpulan tindakan tertentu, lalu untuk Tindakan kebijakan ini akan berlaku, pilih UpdateAlbum.
8. Pilih Berikutnya.
9. Di bawah Detail, untuk Deskripsi kebijakan - masukkan opsional **Policy allowing alice to update alice-favorites-album..**
10. Pilih Buat kebijakan.

Setelah membuat kebijakan, Anda dapat mengujinya di konsol Izin Terverifikasi.

Langkah 3: Menguji toko kebijakan

Setelah membuat penyimpanan kebijakan dan kebijakan, Anda dapat mengujinya dengan menjalankan [permintaan otorisasi](#) simulasi menggunakan bangku uji Izin Terverifikasi.

Untuk menguji kebijakan toko kebijakan

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Test bench.
3. Pilih mode Visual.

4. Untuk Principal, lakukan hal berikut:
 - a. Untuk Principal mengambil tindakan pilih PhotoFlash: :User dan untuk Specify entity identifier, enter. **alice**
 - b. Di bawah Atribut, untuk Account: Entity, pastikan entitas PhotoFlash: :Account dipilih, dan untuk Specify entity identifier, masukkan. **alice-account**
5. Di bawah Sumber Daya, untuk Sumber Daya yang digunakan oleh prinsipal, pilih jenis sumber daya PhotoFlash: :Album dan untuk Tentukan pengenal entitas, masukkan. **alice-favorites-album**
6. Untuk Tindakan, pilih PhotoFlash: :Action:." UpdateAlbum "dari daftar tindakan yang valid.
7. Di bagian atas halaman, pilih Jalankan permintaan otorisasi untuk mensimulasikan permintaan otorisasi untuk kebijakan Cedar di toko kebijakan sampel. Bangku tes harus menampilkan Keputusan: Izinkan menunjukkan kebijakan kami berfungsi seperti yang diharapkan.

Tabel berikut memberikan nilai tambahan untuk prinsipal, sumber daya, dan tindakan yang dapat Anda uji dengan bangku tes Izin Terverifikasi. Tabel ini mencakup keputusan permintaan otorisasi berdasarkan kebijakan statis yang disertakan dengan penyimpanan kebijakan PhotoFlash sampel dan kebijakan yang Anda buat di langkah 2.

Nilai pokok	Akun Utama: Nilai entitas	Nilai sumber daya	Nilai induk sumber daya	Tindakan	Keputusan otorisasi
PhotoFlas h: :Pengguna bob	PhotoFlas h: :Akun alice-akun	PhotoFlas h: :Album alice-fav orites-album	N/A	PhotoFlas h: :Tindakan ::"" UpdateAlb um	Menyangkal
PhotoFlas h: :Pengguna alice	PhotoFlas h: :Akun alice-akun	PhotoFlas h: :Foto photo.jpeg	PhotoFlas h: :Akun bob-akun	PhotoFlas h: :Tindakan ::"" ViewPhoto	Menyangkal
PhotoFlas h: :Pengguna alice	PhotoFlas h: :Akun alice-akun	PhotoFlas h: :Foto photo.jpeg	PhotoFlas h: :Akun alice-akun	PhotoFlas h: :Tindakan ::"" ViewPhoto	Izinkan

Nilai pokok	Akun Utama: Nilai entitas	Nilai sumber daya	Nilai induk sumber daya	Tindakan	Keputusan otorisasi
PhotoFlas h: :Pengguna alice	PhotoFlas h: :Akun alice-akun	PhotoFlas h: :Foto bob- photo.jpeg	PhotoFlas h: :Album Bob-Vacation-Album	PhotoFlas h: :Tindakan :.” DeletePhoto	Menyangkal

Langkah 4: Bersihkan Sumber Daya

Setelah Anda selesai menjelajahi toko kebijakan Anda, hapus.

Untuk menghapus toko kebijakan

1. Di [konsol Izin Terverifikasi](#), pilih penyimpanan kebijakan yang Anda buat di langkah 1.
2. Di navigasi, pilih Pengaturan.
3. Di bawah Hapus toko kebijakan, pilih Hapus toko kebijakan ini.
4. Di toko Hapus kebijakan ini? kotak dialog, masukkan hapus, lalu pilih Hapus.

Praktik terbaik untuk merancang model otorisasi

Saat Anda bersiap untuk menggunakan layanan Izin Terverifikasi Amazon dalam aplikasi perangkat lunak, mungkin sulit untuk langsung menulis pernyataan kebijakan sebagai langkah pertama. Ini akan mirip dengan memulai pengembangan bagian lain dari aplikasi dengan menulis pernyataan SQL atau spesifikasi API sebelum sepenuhnya memutuskan apa yang harus dilakukan aplikasi. Sebagai gantinya, Anda harus mulai dengan pengalaman pengguna. Kemudian, bekerja mundur dari pengalaman itu untuk sampai pada pendekatan implementasi.

Saat Anda melakukan pekerjaan ini, Anda akan menemukan diri Anda mengajukan pertanyaan seperti:

- Apa sumber daya saya? Bagaimana mereka terorganisir? Misalnya, apakah file berada di dalam folder?
- Apakah organisasi sumber daya berperan dalam model izin?
- Tindakan apa yang dapat dilakukan kepala sekolah pada setiap sumber daya?
- Bagaimana cara kepala sekolah memperoleh izin tersebut?
- Apakah Anda ingin pengguna akhir Anda memilih dari izin yang telah ditentukan sebelumnya seperti “Admin”, “Operator”, atau “ReadOnly”, atau haruskah mereka membuat pernyataan kebijakan ad-hoc? Atau keduanya?
- Apakah peran global atau tercakup? Misalnya, apakah “operator” terbatas dalam satu penyewa, atau apakah “operator” berarti operator di seluruh aplikasi?
- Jenis kueri apa yang diperlukan untuk membuat pengalaman pengguna? Misalnya, apakah Anda perlu mencantumkan semua sumber daya yang dapat diakses oleh prinsipal untuk membuat halaman beranda pengguna tersebut?
- Bisakah pengguna secara tidak sengaja mengunci diri dari sumber daya mereka sendiri? Apakah itu perlu dihindari?

Hasil akhir dari latihan ini disebut sebagai model otorisasi; itu mendefinisikan prinsip, sumber daya, tindakan, dan bagaimana mereka saling berhubungan satu sama lain. Memproduksi model ini tidak memerlukan pengetahuan unik tentang Cedar atau layanan Izin Terverifikasi. Sebaliknya, ini adalah latihan desain pengalaman pengguna pertama dan terutama, seperti yang lainnya, dan dapat bermanifestasi dalam artefak seperti maket antarmuka, diagram logis, dan deskripsi keseluruhan tentang bagaimana izin memengaruhi apa yang dapat dilakukan pengguna dalam produk. Cedar dirancang agar cukup fleksibel untuk bertemu pelanggan pada model, daripada

memaksa model untuk membungkuk secara tidak wajar untuk mematuhi implementasi Cedar. Akibatnya, mendapatkan pemahaman yang tajam tentang pengalaman pengguna yang diinginkan adalah cara terbaik untuk sampai pada model yang optimal.

Untuk membantu menjawab pertanyaan dan sampai pada model yang optimal, lakukan hal berikut:

- Tinjau [pola desain Cedar](#) dalam bahasa kebijakan Cedar Panduan Referensi.
- Pertimbangkan [praktik terbaik](#) dalam Panduan Referensi bahasa kebijakan Cedar.
- Pertimbangkan praktik terbaik yang disertakan di halaman ini.

Praktik terbaik

- [Tidak ada model “benar” kanonik](#)
- [Kembalikan 403 kesalahan terlarang daripada 404 kesalahan tidak ditemukan](#)
- [Fokus pada sumber daya Anda di luar operasi API](#)
- [Pertimbangan multi-tenancy](#)

Tidak ada model “benar” kanonik

Saat Anda merancang model otorisasi, tidak ada jawaban tunggal yang benar dan unik. Aplikasi yang berbeda dapat secara efektif menggunakan model otorisasi yang berbeda untuk konsep serupa, dan ini tidak masalah. Misalnya, perhatikan representasi sistem file komputer. Saat Anda membuat file dalam sistem operasi mirip Unix, file tersebut tidak secara otomatis mewarisi izin dari folder induk. Sebaliknya, di banyak sistem operasi lain dan sebagian besar layanan berbagi file online, file mewarisi izin dari folder induknya. Kedua pilihan tersebut valid tergantung pada keadaan yang dioptimalkan aplikasi.

Kebenaran solusi otorisasi tidak mutlak, tetapi harus dilihat dalam hal bagaimana memberikan pengalaman yang diinginkan pelanggan Anda, dan apakah itu melindungi sumber daya mereka dengan cara yang mereka harapkan. Jika model otorisasi Anda memberikan ini, maka itu berhasil.

Inilah sebabnya mengapa memulai desain Anda dengan pengalaman pengguna yang diinginkan adalah prasyarat paling membantu untuk pembuatan model otorisasi yang efektif.

Kembalikan 403 kesalahan terlarang daripada 404 kesalahan tidak ditemukan

Yang terbaik adalah mengembalikan kesalahan 403 Forbidden ke permintaan yang menyertakan entitas, terutama sumber daya, yang tidak sesuai dengan kebijakan apa pun daripada kesalahan 404 Tidak ditemukan. Ini memberikan tingkat keamanan tertinggi karena Anda tidak mengekspos apakah suatu entitas ada atau tidak, hanya saja permintaan tersebut tidak memenuhi ketentuan kebijakan dalam kebijakan apa pun di toko kebijakan.

Fokus pada sumber daya Anda di luar operasi API

Di sebagian besar aplikasi, izin dimodelkan di sekitar sumber daya yang didukung. Misalnya, aplikasi berbagi file mungkin mewakili izin sebagai tindakan yang dapat dilakukan pada file atau folder. Ini adalah model yang bagus dan sederhana yang mengabstraksi implementasi yang mendasarinya dan operasi API backend.

Sebaliknya, jenis aplikasi lain, terutama layanan web, sering merancang izin di sekitar operasi API itu sendiri. Misalnya, jika layanan web menyediakan API bernama `createThing()`, model otorisasi mungkin menentukan izin yang sesuai, atau `action` dalam nama Cedar. `createThing` Ini bekerja dalam banyak situasi dan membuatnya mudah untuk memahami izin. Untuk menjalankan `createThing` operasi, Anda memerlukan izin `createThing` tindakan. Sepertinya sederhana, kan?

Anda akan menemukan bahwa proses [memulai](#) di konsol Izin Terverifikasi menyertakan opsi untuk membangun sumber daya dan tindakan Anda langsung dari API. Ini adalah garis dasar yang berguna: pemetaan langsung antara toko kebijakan Anda dan API yang diotorisasi.

Namun, saat Anda mengembangkan model Anda lebih lanjut, pendekatan yang berfokus pada API ini mungkin tidak cocok untuk aplikasi dengan model otorisasi yang sangat terperinci karena APIs hanyalah proxy untuk apa yang benar-benar ingin dilindungi oleh pelanggan Anda: data dan sumber daya yang mendasarinya. Jika beberapa APIs kontrol akses ke sumber daya yang sama, mungkin sulit bagi administrator untuk bernalar tentang jalur ke sumber daya tersebut dan mengelola akses yang sesuai.

Misalnya, pertimbangkan direktori pengguna yang berisi anggota organisasi. Pengguna dapat diatur ke dalam kelompok, dan salah satu tujuan keamanan adalah untuk melarang penemuan keanggotaan kelompok oleh pihak yang tidak berwenang. Layanan yang mengelola direktori pengguna ini menyediakan dua operasi API:

- `listMembersOfGroup`
- `listGroupMembershipsForUser`

Pelanggan dapat menggunakan salah satu dari operasi ini untuk menemukan keanggotaan grup. Oleh karena itu, administrator izin harus ingat untuk mengoordinasikan akses ke kedua operasi. Ini lebih rumit jika nanti Anda memilih untuk menambahkan operasi API baru untuk mengatasi kasus penggunaan tambahan, seperti berikut ini.

- `isUserInGroups` (API baru untuk menguji dengan cepat apakah pengguna termasuk dalam satu atau beberapa grup)

Dari perspektif keamanan, API ini membuka jalur ketiga untuk menemukan keanggotaan grup, mengganggu izin administrator yang dibuat dengan cermat.

Kami menyarankan Anda untuk fokus pada data dan sumber daya yang mendasarinya serta operasi asosiasi mereka. Menerapkan pendekatan ini ke contoh keanggotaan grup akan menghasilkan izin abstrak, seperti `viewGroupMembership`, yang masing-masing dari tiga operasi API harus berkonsultasi.

Nama API	Izin
<code>listMembersOfGroup</code>	memerlukan <code>viewGroupMembership</code> izin pada grup
<code>listGroupMembershipsForUser</code>	memerlukan <code>viewGroupMembership</code> izin pada pengguna
<code>isUserInGroups</code>	memerlukan <code>viewGroupMembership</code> izin pada pengguna

Dengan mendefinisikan izin yang satu ini, administrator berhasil mengontrol akses untuk menemukan keanggotaan grup, sekarang dan selamanya. Sebagai tradeoff, setiap operasi API sekarang harus mendokumentasikan kemungkinan beberapa izin yang diperlukan, dan administrator harus berkonsultasi dengan dokumentasi ini saat membuat izin. Ini bisa menjadi tradeoff yang valid bila diperlukan untuk memenuhi persyaratan keamanan Anda.

Pertimbangan multi-tenancy

Anda mungkin ingin mengembangkan aplikasi untuk digunakan oleh banyak pelanggan - bisnis yang menggunakan aplikasi Anda, atau penyewa - dan mengintegrasikannya dengan Izin Terverifikasi Amazon. Sebelum Anda mengembangkan model otorisasi Anda, kembangkan strategi multi-penyewa. Anda dapat mengelola kebijakan pelanggan Anda di satu toko kebijakan bersama, atau menetapkan masing-masing toko kebijakan per penyewa. Untuk informasi selengkapnya, lihat [Pertimbangan desain multi-penyewa Izin Terverifikasi Amazon](#) di Panduan Preskriptif.AWS

1. Satu toko kebijakan bersama

Semua penyewa berbagi satu toko kebijakan. Aplikasi mengirimkan semua permintaan otorisasi ke toko kebijakan bersama.

2. Toko kebijakan per penyewa

Setiap penyewa memiliki toko kebijakan khusus. Aplikasi akan menanyakan toko kebijakan yang berbeda untuk keputusan otorisasi, tergantung pada penyewa yang membuat permintaan.

Tidak ada strategi yang akan berdampak besar pada AWS tagihan Anda. Jadi bagaimana, kemudian, Anda harus merancang pendekatan Anda? Berikut ini adalah kondisi umum yang mungkin berkontribusi pada strategi otorisasi multi-tenancy Izin Terverifikasi Anda.

Isolasi kebijakan penyewa

Isolasi kebijakan masing-masing penyewa dari yang lain penting untuk melindungi data penyewa. Ketika setiap penyewa memiliki toko kebijakan mereka sendiri, mereka masing-masing memiliki serangkaian kebijakan mereka sendiri yang terisolasi.

Aliran otorisasi

Anda dapat mengidentifikasi penyewa yang membuat permintaan otorisasi dengan ID toko kebijakan dalam permintaan, dengan toko kebijakan per penyewa. Dengan penyimpanan kebijakan bersama, semua permintaan menggunakan ID penyimpanan kebijakan yang sama.

Template dan manajemen skema

Jika aplikasi Anda memiliki beberapa toko kebijakan, [templat kebijakan](#) dan [skema toko kebijakan](#) menambahkan tingkat overhead desain dan pemeliharaan di setiap toko kebijakan.

Manajemen kebijakan global

Anda mungkin ingin menerapkan beberapa kebijakan global untuk setiap penyewa. Tingkat overhead untuk pengelolaan kebijakan global bervariasi antara model toko kebijakan bersama dan per-penyewa.

Penyewa off-boarding

Beberapa penyewa akan menyumbangkan elemen ke skema dan kebijakan Anda yang spesifik untuk kasus mereka. Ketika penyewa tidak lagi aktif dengan organisasi Anda dan Anda ingin menghapus data mereka, tingkat upaya bervariasi dengan tingkat isolasi mereka dari penyewa lain.

Kuota sumber daya layanan

Izin Terverifikasi memiliki kuota sumber daya dan tingkat permintaan yang dapat memengaruhi keputusan multi-tenancy Anda. Untuk informasi lebih lanjut tentang kuota, lihat [Kuota untuk sumber daya](#).

Membandingkan toko kebijakan bersama dan toko kebijakan per penyewa

Setiap pertimbangan membutuhkan tingkat waktu dan komitmen sumber dayanya sendiri dalam model toko kebijakan bersama dan per-penyewa.

Pertimbangan	Tingkat upaya di toko kebijakan bersama	Tingkat upaya di toko kebijakan per penyewa
Isolasi kebijakan penyewa	Sedang. Harus menyertakan pengenalan penyewa dalam kebijakan dan permintaan otorisasi.	Rendah. Isolasi adalah perilaku default. Kebijakan khusus penyewa tidak dapat diakses oleh penyewa lain.
Aliran otorisasi	Rendah. Semua kueri menargetkan satu toko kebijakan.	Sedang. Harus memelihara pemetaan antara setiap penyewa dan ID toko polis mereka.
Template dan manajemen skema	Rendah. Harus membuat satu skema bekerja untuk semua penyewa.	Tinggi. Skema dan templat mungkin kurang kompleks secara individual, tetapi

perubahan membutuhkan lebih banyak koordinasi dan kompleksitas.

Manajemen kebijakan global	Rendah. Semua kebijakan bersifat global dan dapat diperbarui secara terpusat.	Tinggi. Anda harus menambahkan kebijakan global ke setiap toko kebijakan dalam orientasi. Replikasi pembaruan kebijakan global di antara banyak toko kebijakan.
Penyewa off-boarding	Tinggi. Harus mengidentifikasi dan menghapus hanya kebijakan khusus penyewa.	Rendah. Hapus toko kebijakan .
Kuota sumber daya layanan	Tinggi. Penyewa berbagi kuota sumber daya yang memengaruhi penyimpanan kebijakan seperti ukuran skema, ukuran kebijakan per sumber daya, dan sumber identitas per toko kebijakan.	Rendah. Setiap penyewa memiliki kuota sumber daya khusus.

Bagaimana memilih

Setiap aplikasi multi-tenant berbeda. Hati-hati membandingkan dua pendekatan dan pertimbangan mereka sebelum membuat keputusan arsitektur.

Jika aplikasi Anda tidak memerlukan kebijakan khusus penyewa dan menggunakan satu [sumber identitas](#), satu penyimpanan kebijakan bersama untuk semua penyewa kemungkinan akan menjadi solusi yang paling efektif. Ini menghasilkan aliran otorisasi yang lebih sederhana dan manajemen kebijakan global. Off-boarding penyewa menggunakan satu toko kebijakan bersama membutuhkan lebih sedikit usaha karena aplikasi tidak perlu menghapus kebijakan khusus penyewa.

Tetapi jika aplikasi Anda memerlukan banyak kebijakan khusus penyewa, atau menggunakan beberapa [sumber identitas](#), toko kebijakan per penyewa kemungkinan akan paling efektif. Anda dapat mengontrol akses ke kebijakan penyewa dengan IAM kebijakan yang memberikan izin per

penyewa ke setiap penyimpanan kebijakan. Off-boarding penyewa melibatkan penghapusan toko kebijakan mereka; di shared-policy-store lingkungan, Anda harus menemukan dan menghapus kebijakan khusus penyewa.

Toko kebijakan Izin Terverifikasi Amazon

Toko kebijakan adalah wadah untuk kebijakan dan templat kebijakan. Di setiap toko kebijakan, Anda dapat membuat skema yang digunakan untuk memvalidasi kebijakan yang ditambahkan ke penyimpanan kebijakan. Selain itu, Anda dapat mengaktifkan validasi kebijakan. Jika Anda menambahkan kebijakan ke penyimpanan kebijakan dengan validasi kebijakan diaktifkan, jenis entitas, tipe umum, dan tindakan yang ditentukan dalam kebijakan akan divalidasi terhadap skema dan kebijakan yang tidak valid akan ditolak.

Perlindungan penghapusan mencegah penghapusan toko kebijakan yang tidak disengaja. Perlindungan penghapusan diaktifkan pada semua toko kebijakan baru yang dibuat melalui Konsol Manajemen AWS. Sebaliknya, ini dinonaktifkan untuk semua penyimpanan kebijakan yang dibuat melalui panggilan API atau SDK.

Sebaiknya buat satu toko kebijakan per aplikasi, atau satu toko kebijakan per penyewa untuk aplikasi multi-penyewa. Anda harus menentukan toko kebijakan saat membuat [permintaan otorisasi](#). Anda juga dapat membuat alias toko kebijakan untuk merujuk ke toko kebijakan Anda dengan nama yang ramah. Untuk informasi selengkapnya, lihat [Alias penyimpanan kebijakan Izin Terverifikasi Amazon](#).

Sebaiknya gunakan ruang nama ke entitas Cedar di toko kebijakan Anda untuk mencegah ambiguitas. Namespace adalah awalan string untuk tipe, dipisahkan oleh sepasang titik dua (: :) sebagai pembatas. Sebagai contoh, `MyApplicationNamespace::exampleType`. Izin Terverifikasi mendukung satu namespace per toko kebijakan. Ruang nama ini membantu menjaga semuanya tetap lurus saat Anda bekerja dengan beberapa aplikasi serupa. Misalnya, dalam aplikasi multi-tenant, menggunakan namespace untuk menambahkan nama penyewa ke tipe yang ditentukan dalam skema akan membuat mereka berbeda dari rekan serupa yang digunakan oleh penyewa lainnya. Saat melihat log untuk permintaan otorisasi, Anda akan dapat dengan mudah mengidentifikasi penyewa yang memproses permintaan otorisasi. Untuk informasi selengkapnya, lihat [Ruang nama dalam Panduan Referensi bahasa kebijakan Cedar](#).

Topik

- [Membuat toko kebijakan Izin Terverifikasi](#)
- [Toko kebijakan terkait API](#)
- [Menghapus toko kebijakan](#)

Membuat toko kebijakan Izin Terverifikasi

Anda dapat membuat toko kebijakan menggunakan metode berikut:

- Ikuti pengaturan terpandu — Anda akan menentukan jenis sumber daya dengan tindakan yang valid dan tipe utama sebelum membuat kebijakan pertama Anda.
- Siapkan dengan API Gateway dan sumber identitas — Tentukan entitas utama Anda dengan pengguna yang masuk dengan penyedia identitas (iDP), serta entitas tindakan dan sumber daya Anda dari API Amazon API Gateway. Kami merekomendasikan opsi ini jika Anda ingin aplikasi Anda mengotorisasi permintaan API dengan keanggotaan grup pengguna atau atribut lainnya.
- Mulai dari toko kebijakan sampel — Pilih toko kebijakan proyek sampel yang telah ditentukan sebelumnya. Kami merekomendasikan opsi ini jika Anda mempelajari tentang Izin Terverifikasi dan ingin melihat dan menguji kebijakan contoh.
- Buat toko kebijakan kosong — Anda akan menentukan skema dan semua kebijakan akses sendiri. Kami merekomendasikan opsi ini jika Anda sudah terbiasa dengan mengonfigurasi toko kebijakan.

Guided setup


Untuk membuat penyimpanan kebijakan menggunakan metode konfigurasi penyiapan Terpandu

Panduan penyiapan terpandu mengarahkan Anda melalui proses pembuatan iterasi pertama penyimpanan kebijakan Anda. Anda akan membuat skema untuk jenis sumber daya pertama Anda, menjelaskan tindakan yang berlaku untuk jenis sumber daya tersebut, dan jenis utama yang Anda berikan izin. Anda kemudian akan membuat kebijakan pertama Anda. Setelah menyelesaikan wizard ini, Anda akan dapat menambahkan ke toko kebijakan Anda, memperluas skema untuk menjelaskan sumber daya dan jenis utama lainnya, dan membuat kebijakan dan templat tambahan.

1. Di [konsol Izin Terverifikasi](#), pilih Buat toko kebijakan baru.
2. Di bagian Opsi awal, pilih Pengaturan terpandu.
3. Masukkan deskripsi toko Kebijakan. Teks ini dapat berupa apa pun yang sesuai dengan organisasi Anda sebagai referensi ramah untuk fungsi toko kebijakan saat ini, misalnya aplikasi web pembaruan cuaca.
4. Di bagian Detail, ketik Namespace untuk skema Anda. Untuk informasi selengkapnya tentang ruang nama, lihat. [Definisi namespace](#)
5. Pilih Berikutnya.

6. Pada jendela Jenis sumber daya, ketikkan nama untuk jenis sumber daya Anda. Misalnya, `currentTemperature` bisa menjadi sumber daya untuk aplikasi web pembaruan Cuaca.
7. (Opsional) Pilih Tambahkan atribut untuk menambahkan atribut sumber daya. Ketik nama Atribut dan pilih tipe Atribut untuk setiap atribut sumber daya. Pilih apakah setiap atribut Diperlukan. Misalnya, `temperatureFormat` bisa menjadi atribut untuk `currentTemperature` sumber daya dan menjadi Fahrenheit atau Celcius. Untuk menghapus atribut yang telah ditambahkan untuk jenis sumber daya, pilih Hapus di sebelah atribut.
8. Di bidang Tindakan, ketik tindakan yang akan diotorisasi untuk jenis sumber daya yang ditentukan. Untuk menambahkan tindakan tambahan untuk jenis sumber daya, pilih Tambahkan tindakan. Misalnya, `viewTemperature` bisa menjadi tindakan dalam aplikasi web pembaruan Cuaca. Untuk menghapus tindakan yang telah ditambahkan untuk jenis sumber daya, pilih Hapus di samping tindakan.
9. Di bidang Nama tipe utama, ketikkan nama untuk jenis prinsipal yang akan menggunakan tindakan yang ditentukan untuk jenis sumber daya Anda. Secara default, Pengguna ditambahkan ke bidang ini tetapi dapat diganti.
10. Pilih Berikutnya.
11. Pada jendela Principal type, pilih sumber identitas untuk tipe utama Anda.
 - Pilih Kustom jika ID dan atribut kepala sekolah akan diberikan langsung oleh aplikasi Izin Terverifikasi Anda. Pilih Tambahkan atribut untuk menambahkan atribut utama. Izin Terverifikasi menggunakan nilai atribut yang ditentukan saat memverifikasi kebijakan terhadap skema. Untuk menghapus atribut yang telah ditambahkan untuk tipe utama, pilih Hapus di sebelah atribut.
 - Pilih Kumpulan Pengguna Cognito jika ID dan atribut prinsipal akan diberikan dari ID atau token akses yang dihasilkan oleh Amazon Cognito Pilih Connect user pool. Pilih AWS Region dan ketik ID kumpulan pengguna dari kumpulan Amazon Cognito pengguna untuk terhubung. Pilih Hubungkan. Untuk informasi selengkapnya, lihat [Otorisasi dengan Izin Terverifikasi Amazon di Panduan](#) Pengembang Amazon Cognito.
 - Pilih penyedia OIDC Eksternal jika ID dan atribut prinsipal akan diekstraksi dari token and/or Akses ID, yang dihasilkan oleh penyedia OIDC eksternal dan tambahkan detail penyedia dan token.
12. Pilih Berikutnya.
13. Di bagian Detail kebijakan, ketikkan deskripsi Kebijakan opsional untuk kebijakan Cedar pertama Anda.

14. Di bidang cakupan Prinsipal, pilih prinsipal yang akan diberikan izin dari kebijakan.
 - Pilih Kepala Sekolah Khusus untuk menerapkan kebijakan ke kepala sekolah tertentu. Pilih prinsipal di Principal yang akan diizinkan untuk mengambil tindakan dan ketik pengenal entitas untuk prinsipal. Misalnya, `user-id` bisa menjadi pengenal entitas dalam aplikasi web pembaruan Cuaca.

 Note

Jika Anda menggunakan Amazon Cognito, pengenal entitas harus diformat sebagai. `<userpool-id>|<sub>`

- Pilih Semua kepala sekolah untuk menerapkan kebijakan ini ke semua kepala sekolah di toko polis Anda.
15. Di bidang lingkup Sumber Daya, pilih sumber daya mana yang akan diberi wewenang untuk ditindaklanjuti oleh prinsipal tertentu.
 - Pilih Sumber daya khusus untuk menerapkan kebijakan ke sumber daya tertentu. Pilih sumber daya di Sumber daya yang harus diterapkan kebijakan ini ke bidang dan ketik pengenal entitas untuk sumber daya. Misalnya, `temperature-id` bisa menjadi pengenal entitas dalam aplikasi web pembaruan Cuaca.
 - Pilih Semua sumber daya untuk menerapkan kebijakan ke semua sumber daya di toko kebijakan Anda.
 16. Di bidang lingkup Tindakan, pilih tindakan mana yang akan diotorisasi oleh prinsipal tertentu untuk dilakukan.
 - Pilih Kumpulan tindakan khusus untuk menerapkan kebijakan pada tindakan tertentu. Pilih kotak centang di samping tindakan dalam bidang Tindakan yang harus diterapkan kebijakan ini.
 - Pilih Semua tindakan untuk menerapkan kebijakan ke semua tindakan di toko kebijakan Anda.
 17. Tinjau kebijakan di bagian Pratinjau kebijakan. Pilih Buat toko kebijakan.

Set up with API Gateway and an identity source

Untuk membuat penyimpanan kebijakan menggunakan metode Mengatur dengan API Gateway dan konfigurasi sumber identitas

API Gateway Opsi ini mengamankan APIs dengan kebijakan Izin Terverifikasi yang dirancang untuk membuat keputusan otorisasi dari grup, atau peran pengguna. Opsi ini membangun penyimpanan kebijakan untuk menguji otorisasi dengan grup sumber identitas dan API dengan otorisasi Lambda.

Pengguna dan grup mereka dalam IDP menjadi prinsipal Anda (token ID) atau konteks Anda (token akses). Metode dan jalur dalam API Gateway API menjadi tindakan yang diotorisasi oleh kebijakan Anda. Aplikasi Anda menjadi sumber daya. Sebagai hasil dari alur kerja ini, Izin Terverifikasi membuat penyimpanan kebijakan, fungsi Lambda, dan otorisasi API Lambda. Anda harus menetapkan otorisasi [Lambda](#) ke API Anda setelah menyelesaikan alur kerja ini.

1. Di [konsol Izin Terverifikasi](#), pilih Buat toko kebijakan baru.
2. Di bagian Opsi awal, pilih Siapkan dengan API Gateway dan sumber identitas dan pilih Berikutnya.
3. Pada langkah Impor sumber daya dan tindakan, di bawah API, pilih API yang akan berfungsi sebagai model untuk sumber daya dan tindakan penyimpanan kebijakan Anda.
 - a. Pilih tahap Deployment dari tahapan yang dikonfigurasi di API Anda dan pilih Impor API. Untuk informasi selengkapnya tentang tahapan API, lihat [Menyiapkan tahapan untuk REST API di Panduan Pengembang Amazon API Gateway](#).
 - b. Pratinjau Peta sumber daya dan tindakan yang diimpor.
 - c. Untuk memperbarui sumber daya atau tindakan, ubah jalur atau metode API Anda di API Gateway konsol dan pilih Impor API untuk melihat pembaruan.
 - d. Ketika Anda puas dengan pilihan Anda, pilih Berikutnya.
4. Di Sumber identitas, pilih jenis penyedia Identitas. Anda dapat memilih kumpulan Amazon Cognito pengguna atau tipe IdP OpenID Connect (OIDC).
5. Jika Anda memilih Amazon Cognito:
 - a. Pilih kumpulan pengguna yang sama AWS Region dan Akun AWS sebagai toko kebijakan Anda.
 - b. Pilih jenis Token untuk diteruskan ke API yang ingin Anda kirimkan untuk otorisasi. Jenis token mana pun berisi grup pengguna, dasar dari model otorisasi terkait API ini.

- c. Di bawah Validasi klien App, Anda dapat membatasi cakupan penyimpanan kebijakan ke subset klien Amazon Cognito aplikasi dalam kumpulan pengguna multi-penyewa. Untuk mengharuskan pengguna melakukan autentikasi dengan satu atau beberapa klien aplikasi tertentu di kumpulan pengguna Anda, pilih Hanya terima token dengan klien IDs aplikasi yang diharapkan. Untuk menerima pengguna yang melakukan autentikasi dengan kumpulan pengguna, pilih Jangan memvalidasi klien aplikasi. IDs
 - d. Pilih Berikutnya.
6. Jika Anda memilih penyedia OIDC Eksternal:
- a. Di URL Penerbit, masukkan URL penerbit OIDC Anda. Ini adalah titik akhir layanan yang menyediakan server otorisasi, kunci penandatanganan, dan informasi lain tentang penyedia Anda, misalnya. `https://auth.example.com` URL penerbit Anda harus meng-host dokumen penemuan OIDC di `/.well-known/openid-configuration`
 - b. Pada tipe Token, pilih jenis OIDC JWT yang Anda ingin aplikasi Anda kirimkan untuk otorisasi. Untuk informasi selengkapnya, lihat [Memetakan Amazon Cognito token ke skema](#) dan [Memetakan token OIDC ke skema](#).
 - c. (opsional) Dalam klaim Token - opsional, pilih Tambahkan klaim token, masukkan nama untuk token, dan pilih jenis nilai.
 - d. Dalam klaim token Pengguna dan grup, lakukan hal berikut:
 - i. Masukkan nama klaim Pengguna dalam token untuk sumber identitas. Ini adalah klaim, biasanya `sub`, dari ID atau token akses Anda yang memegang pengenalan unik untuk entitas yang akan dievaluasi. Identitas dari IDP OIDC yang terhubung akan dipetakan ke jenis pengguna di toko kebijakan Anda.
 - ii. Masukkan nama klaim Grup dalam token untuk sumber identitas. Ini adalah klaim, biasanya `groups`, dari ID atau token akses Anda yang berisi daftar grup pengguna. Toko kebijakan Anda akan mengotorisasi permintaan berdasarkan keanggotaan grup.
 - e. Dalam validasi Audiens, pilih `Add value` dan tambahkan nilai yang ingin diterima oleh toko kebijakan dalam permintaan otorisasi.
 - f. Pilih Berikutnya.
7. Jika Anda memilih Amazon Cognito, Izin Terverifikasi menanyakan kumpulan pengguna Anda untuk grup. Untuk penyedia OIDC, masukkan nama grup secara manual. Langkah Tetapkan tindakan ke grup membuat kebijakan untuk penyimpanan kebijakan Anda yang mengizinkan anggota grup melakukan tindakan.

- a. Pilih atau tambahkan grup yang ingin Anda sertakan dalam kebijakan Anda.
 - b. Tetapkan tindakan ke setiap grup yang Anda pilih.
 - c. Pilih Berikutnya.
8. Di Deploy integrasi aplikasi, pilih apakah Anda ingin melampirkan otorisasi Lambda secara manual nanti secara manual atau jika Anda ingin Izin Terverifikasi melakukannya untuk Anda sekarang dan tinjau langkah-langkah yang akan diambil Izin Terverifikasi untuk membuat toko kebijakan dan otorisasi Lambda.
 9. Saat Anda siap membuat sumber daya baru, pilih Buat toko kebijakan.
 10. Biarkan langkah status penyimpanan Kebijakan tetap terbuka di browser Anda untuk memantau kemajuan pembuatan sumber daya berdasarkan Izin Terverifikasi.
 11. Setelah beberapa waktu, biasanya sekitar satu jam, atau ketika langkah otorisasi Lambda Deploy menunjukkan Sukses, jika Anda memilih untuk melampirkan otorisasi secara manual, konfigurasi otorisasi Anda.

Izin Terverifikasi akan membuat fungsi Lambda dan otorisasi Lambda di API Anda. Pilih Open API untuk menavigasi ke API Anda.

Untuk mempelajari cara menetapkan otorisasi Lambda, lihat [Menggunakan otorisasi API Gateway Lambda](#) di Panduan Pengembang Amazon API Gateway.

- a. Arahkan ke Authorizers untuk API Anda dan catat nama otorisasi yang dibuat oleh Izin Terverifikasi.
 - b. Arahkan ke Sumber Daya dan pilih metode tingkat atas di API Anda.
 - c. Pilih Edit di bawah Pengaturan permintaan metode.
 - d. Atur Authorizer menjadi nama otorisasi yang Anda catat sebelumnya.
 - e. Perluas header permintaan HTTP, masukkan Nama atau AUTHORIZATION, dan pilih Diperlukan.
 - f. Menerapkan tahap API.
 - g. Simpan perubahan Anda.
12. Uji otorisasi Anda dengan token kumpulan pengguna dari jenis Token yang Anda pilih di langkah Pilih sumber identitas. Untuk informasi selengkapnya tentang login dan mengambil token kumpulan pengguna, lihat [Alur autentikasi kumpulan pengguna](#) di Panduan Pengembang Amazon Cognito.

13. Uji otentikasi lagi dengan token kumpulan pengguna di AUTHORIZATION header permintaan ke API Anda.
14. Periksa toko kebijakan baru Anda. Tambahkan dan perbaiki kebijakan.

Sample policy store

Untuk membuat penyimpanan kebijakan menggunakan metode konfigurasi penyimpanan kebijakan Sample

1. Di bagian Opsi awal, pilih Contoh penyimpanan kebijakan.
2. Di bagian proyek Contoh, pilih jenis contoh aplikasi Izin Terverifikasi yang akan digunakan.
 - PhotoFlash adalah contoh aplikasi web yang menghadap pelanggan yang memungkinkan pengguna untuk berbagi foto dan album individual dengan teman-teman. Pengguna dapat mengatur izin berbutir halus tentang siapa yang diizinkan untuk melihat, mengomentari, dan membagikan kembali foto mereka. Pemilik akun juga dapat membuat grup teman dan mengatur foto ke dalam album.
 - DigitalPetStore adalah contoh aplikasi di mana siapa pun dapat mendaftar dan menjadi pelanggan. Pelanggan dapat menambahkan hewan peliharaan untuk dijual, mencari hewan peliharaan, dan memesan. Pelanggan yang telah menambahkan hewan peliharaan dicatat sebagai pemilik hewan peliharaan. Pemilik hewan peliharaan dapat memperbarui detail hewan peliharaan, mengunggah gambar hewan peliharaan, atau menghapus daftar hewan peliharaan. Pelanggan yang telah melakukan pemesanan dicatat sebagai pemilik pesanan. Pemilik pesanan bisa mendapatkan detail pesanan atau membatalkannya. Manajer toko hewan peliharaan memiliki akses administratif.

Note

Penyimpanan kebijakan DigitalPetStoresampel tidak menyertakan templat kebijakan. Toko kebijakan PhotoFlash dan TinyTodo contoh menyertakan templat kebijakan.

- TinyTodo adalah contoh aplikasi yang memungkinkan pengguna untuk membuat taks dan daftar tugas. Pemilik daftar dapat mengelola dan membagikan daftar mereka dan menentukan siapa yang dapat melihat atau mengedit daftar mereka.
3. Namespace untuk skema penyimpanan kebijakan sampel Anda dibuat secara otomatis berdasarkan proyek sampel yang Anda pilih.

4. Pilih Buat toko kebijakan.

Toko kebijakan Anda dibuat dengan kebijakan dan skema untuk toko kebijakan sampel yang Anda pilih. Untuk informasi selengkapnya tentang kebijakan terkait templat yang dapat Anda buat untuk penyimpanan kebijakan sampel, lihat [Izin Terverifikasi Amazon contoh kebijakan terkait templat](#)

Empty policy store

Untuk membuat penyimpanan kebijakan menggunakan metode konfigurasi penyimpanan kebijakan Kosong

1. Di bagian Opsi awal, pilih Kosongkan toko kebijakan.
2. Pilih Buat toko kebijakan.

Penyimpanan kebijakan kosong dibuat tanpa skema, yang berarti kebijakan tidak divalidasi. Untuk informasi selengkapnya tentang memperbarui skema untuk toko kebijakan Anda, lihat [Skema toko kebijakan Izin Terverifikasi Amazon](#).

Untuk informasi selengkapnya tentang membuat kebijakan untuk toko kebijakan Anda, lihat [Membuat kebijakan statis Izin Terverifikasi Amazon](#) dan [Membuat kebijakan terkait templat Izin Terverifikasi Amazon](#).

AWS CLI

Untuk membuat toko kebijakan kosong dengan menggunakan file AWS CLI.

Anda dapat membuat toko kebijakan dengan menggunakan `create-policy-store` operasi.

Note

Toko kebijakan yang Anda buat dengan menggunakan kosong. AWS CLI

- Untuk menambahkan skema, lihat [Skema toko kebijakan Izin Terverifikasi Amazon](#).
- Untuk menambahkan kebijakan, lihat [Membuat kebijakan statis Izin Terverifikasi Amazon](#).
- Untuk menambahkan templat kebijakan, lihat [Membuat templat kebijakan Izin Terverifikasi Amazon](#).

```
$ aws verifiedpermissions create-policy-store \
  --validation-settings "mode=STRICT"
{
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg111111",
  "createdDate": "2023-05-16T17:41:29.103459+00:00",
  "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
  "policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

AWS SDKs

Anda dapat membuat toko kebijakan menggunakan `CreatePolicyStore` API. Untuk informasi selengkapnya, lihat [CreatePolicyStore](#) di Panduan Referensi API Izin Terverifikasi Amazon.

Menerapkan Izin Terverifikasi Amazon di Rust dengan SDK AWS

Topik ini memberikan contoh praktis penerapan Izin Terverifikasi Amazon di Rust dengan AWS SDK. Contoh ini menunjukkan bagaimana mengembangkan model otorisasi yang dapat menguji apakah pengguna dapat melihat foto. Kode sampel menggunakan [aws-sdk-verifiedpermissions](#) peti dari [AWS SDK for Rust](#), yang menawarkan seperangkat alat yang kuat untuk berinteraksi. Layanan AWS

Prasyarat

Sebelum memulai, pastikan bahwa Anda memiliki [AWS CLI](#) yang dikonfigurasi pada sistem Anda dan bahwa Anda terbiasa dengan Rust.

- Untuk petunjuk tentang menginstal AWS CLI, lihat panduan [instalasi AWS CLI](#).
- Untuk petunjuk tentang mengonfigurasi AWS CLI, lihat [Mengonfigurasi pengaturan untuk pengaturan AWS CLI dan Konfigurasi dan file kredensi](#) di AWS CLI
- Untuk informasi selengkapnya tentang Rust, lihat [rust-lang.org](#) dan [AWS SDK for Rust Developer Guide](#).

Dengan lingkungan Anda siap, mari jelajahi cara menerapkan Izin Terverifikasi di Rust.

Uji kode sampel

Kode sampel melakukan hal berikut:

- Menyiapkan klien SDK untuk berkomunikasi AWS
- Membuat [toko kebijakan](#)
- [Mendefinisikan struktur toko kebijakan dengan menambahkan skema](#)
- Menambahkan [kebijakan](#) untuk memeriksa permintaan otorisasi
- Mengirim [permintaan otorisasi](#) pengujian untuk memverifikasi semuanya sudah diatur dengan benar

Untuk menguji kode sampel

1. Buat proyek Rust.
2. Ganti kode yang ada `main.rs` dengan kode berikut:

```
use std::time::Duration;
use std::thread::sleep;
use aws_config::BehaviorVersion;
use aws_sdk_verifiedpermissions::Client;
use aws_sdk_verifiedpermissions::{
    operation::{
        create_policy::CreatePolicyOutput,
        create_policy_store::CreatePolicyStoreOutput,
        is_authorized::IsAuthorizedOutput,
        put_schema::PutSchemaOutput,
    },
    types::{
        ActionIdentifier, EntityIdentifier, PolicyDefinition, SchemaDefinition,
        StaticPolicyDefinition, ValidationSettings
    },
};

//Function that creates a policy store in the client that's passed
async fn create_policy_store(client: &Client, valid_settings: &ValidationSettings)-
> CreatePolicyStoreOutput {
    let policy_store =
    client.create_policy_store().validation_settings(valid_settings.clone()).send().await;
    return policy_store.unwrap();
}

//Function that adds a schema to the policy store in the client
async fn put_schema(client: &Client, ps_id: &str, schema: &str) -> PutSchemaOutput
{
```

```

    let schema =
    client.put_schema().definition(SchemaDefinition::CedarJson(schema.to_string())).policy_store_id(ps_id).send().await;
    return schema.unwrap();
}

//Function that creates a policy in the policy store in the client
async fn create_policy(client: &Client, ps_id: &str,
    policy_definition:&PolicyDefinition) -> CreatePolicyOutput {
    let create_policy =
    client.create_policy().definition(policy_definition.clone()).policy_store_id(ps_id).send().await;
    return create_policy.unwrap();
}

//Function that tests the authorization request to the policy store in the client
async fn authorize(client: &Client, ps_id: &str, principal: &EntityIdentifier,
    action: &ActionIdentifier, resource: &EntityIdentifier) -> IsAuthorizedOutput {
    let is_auth =
    client.is_authorized().principal(principal.to_owned()).action(action.to_owned()).resource(resource.to_owned()).send().await;
    return is_auth.unwrap();
}

#[::tokio::main]
async fn main() -> Result<(), aws_sdk_verifiedpermissions::Error> {

//Set up SDK client
    let config = aws_config::load_defaults(BehaviorVersion::latest()).await;
    let client = aws_sdk_verifiedpermissions::Client::new(&config);

//Create a policy store
    let valid_settings = ValidationSettings::builder()
        .mode({aws_sdk_verifiedpermissions::types::ValidationMode::Strict})
        .build()
        .unwrap();
    let policy_store = create_policy_store(&client, &valid_settings).await;
    println!(
        "Created Policy store with ID: {:?}",
        policy_store.policy_store_id
    );

//Add schema to policy store
    let schema= r#"
        "PhotoFlash": {
            "actions": {

```

```

        "ViewPhoto": {
            "appliesTo": {
                "context": {
                    "type": "Record",
                    "attributes": {}
                },
                "principalTypes": [
                    "User"
                ],
                "resourceTypes": [
                    "Photo"
                ]
            },
            "memberOf": []
        }
    },
    "entityTypes": {
        "Photo": {
            "memberOfTypes": [],
            "shape": {
                "type": "Record",
                "attributes": {
                    "IsPrivate": {
                        "type": "Boolean"
                    }
                }
            }
        }
    },
    "User": {
        "memberOfTypes": [],
        "shape": {
            "attributes": {},
            "type": "Record"
        }
    }
}
}
}
}";
let put_schema = put_schema(&client, &policy_store.policy_store_id,
schema).await;
println!(
    "Created Schema with Namespace: {:?}",
    put_schema.namespaces
);

```

```
//Create policy
let policy_text = r#"
    permit (
        principal in PhotoFlash::User::"alice",
        action == PhotoFlash::Action::"ViewPhoto",
        resource == PhotoFlash::Photo::"VacationPhoto94.jpg"
    );
"#;
let policy_definition =
PolicyDefinition::Static(StaticPolicyDefinition::builder().statement(policy_text).build()).
let policy = create_policy(&client, &policy_store.policy_store_id,
&policy_definition).await;
println!(
    "Created Policy with ID: {:?}",
    policy.policy_id
);

//Break to make sure the resources are created before testing authorization
sleep(Duration::new(2, 0));

//Test authorization
let principal=
EntityIdentifier::builder().entity_id("alice").entity_type("PhotoFlash::User").build().unw
let action =
ActionIdentifier::builder().action_type("PhotoFlash::Action").action_id("ViewPhoto").build
let resource =
EntityIdentifier::builder().entity_id("VacationPhoto94.jpg").entity_type("PhotoFlash::Phot
let auth = authorize(&client, &policy_store.policy_store_id, &principal,
&action, &resource).await;
println!(
    "Decision: {:?}",
    auth.decision
);
println!(
    "Policy ID: {:?}",
    auth.determining_policies
);
Ok(())
}
```

3. Jalankan kode dengan memasukkan `cargo run` di terminal.

Jika kode berjalan dengan benar, terminal akan ditampilkan `Decision: Allow` diikuti dengan ID kebijakan dari kebijakan penentu. Ini berarti Anda telah berhasil membuat toko kebijakan dan mengujinya menggunakan AWS SDK for Rust.

Pembersihan sumber daya

Setelah Anda selesai menjelajahi toko kebijakan Anda, hapus.

Untuk menghapus toko kebijakan

Anda dapat menghapus penyimpanan kebijakan dengan menggunakan `delete-policy-store` operasi, menggantinya `PSEXAMPLEabcdefgh111111` dengan ID penyimpanan kebijakan yang ingin Anda hapus.

```
$ aws verifiedpermissions delete-policy-store \  
  --policy-store-id PSEXAMPLEabcdefgh111111
```

Jika berhasil, perintah ini tidak menghasilkan output.

Toko kebijakan terkait API

Kasus penggunaan umum adalah menggunakan Izin Terverifikasi Amazon untuk mengotorisasi akses pengguna ke APIs host di Amazon API Gateway. Menggunakan wizard di AWS konsol, Anda dapat membuat kebijakan akses berbasis peran untuk pengguna yang dikelola di [Amazon Cognito](#), atau penyedia identitas OIDC (iDP) apa pun, dan AWS Lambda menerapkan Authorizer yang memanggil Izin Terverifikasi untuk mengevaluasi kebijakan ini.

Untuk menyelesaikan wizard, pilih Siapkan dengan API Gateway dan penyedia identitas saat Anda [membuat toko kebijakan baru](#) dan ikuti langkah-langkahnya.

Toko kebijakan terkait API dibuat dan menyediakan model otorisasi dan sumber daya Anda untuk permintaan otorisasi. Toko kebijakan memiliki sumber identitas dan otorisasi Lambda yang terhubung API Gateway ke Izin Terverifikasi. Setelah penyimpanan kebijakan dibuat, Anda dapat mengotorisasi permintaan API berdasarkan keanggotaan grup pengguna. Misalnya, Izin Terverifikasi hanya dapat memberikan akses kepada pengguna yang merupakan anggota `Directors` grup.

[Seiring pertumbuhan aplikasi Anda, Anda dapat menerapkan otorisasi berbutir halus dengan atribut pengguna dan cakupan OAuth 2.0 menggunakan bahasa kebijakan Cedar.](#) Misalnya, Izin

Terverifikasi hanya dapat memberikan akses kepada pengguna yang memiliki email atribut di `domainmycompany.co.uk`.

Setelah menyiapkan model otorisasi untuk API Anda, tanggung jawab Anda yang tersisa adalah mengautentikasi pengguna dan membuat permintaan API di aplikasi Anda, serta mempertahankan penyimpanan kebijakan Anda.

Untuk melihat demo, lihat [Izin Terverifikasi Amazon - Ikhtisar Mulai Cepat dan Demo](#) di Amazon Web Services YouTube saluran.

Topik

- [Bagaimana Izin Terverifikasi mengotorisasi permintaan API](#)
- [Pertimbangan untuk toko kebijakan terkait API](#)
- [Menambahkan kontrol akses berbasis atribut \(ABAC\)](#)
- [Pindah ke produksi dengan AWS CloudFormation](#)
- [Pemecahan masalah toko kebijakan terkait API](#)

Important

Penyimpanan kebijakan yang Anda buat dengan opsi Penyiapan dengan API Gateway dan sumber identitas di konsol Izin Terverifikasi tidak dimaksudkan untuk penerapan langsung ke produksi. Dengan toko kebijakan awal Anda, selesaikan model otorisasi Anda dan ekspor sumber daya penyimpanan kebijakan ke CloudFormation Menyebarkan Izin Terverifikasi untuk produksi secara terprogram dengan [AWS Cloud Development Kit \(AWS CDK\)](#) Untuk informasi selengkapnya, lihat [Pindah ke produksi dengan AWS CloudFormation](#).

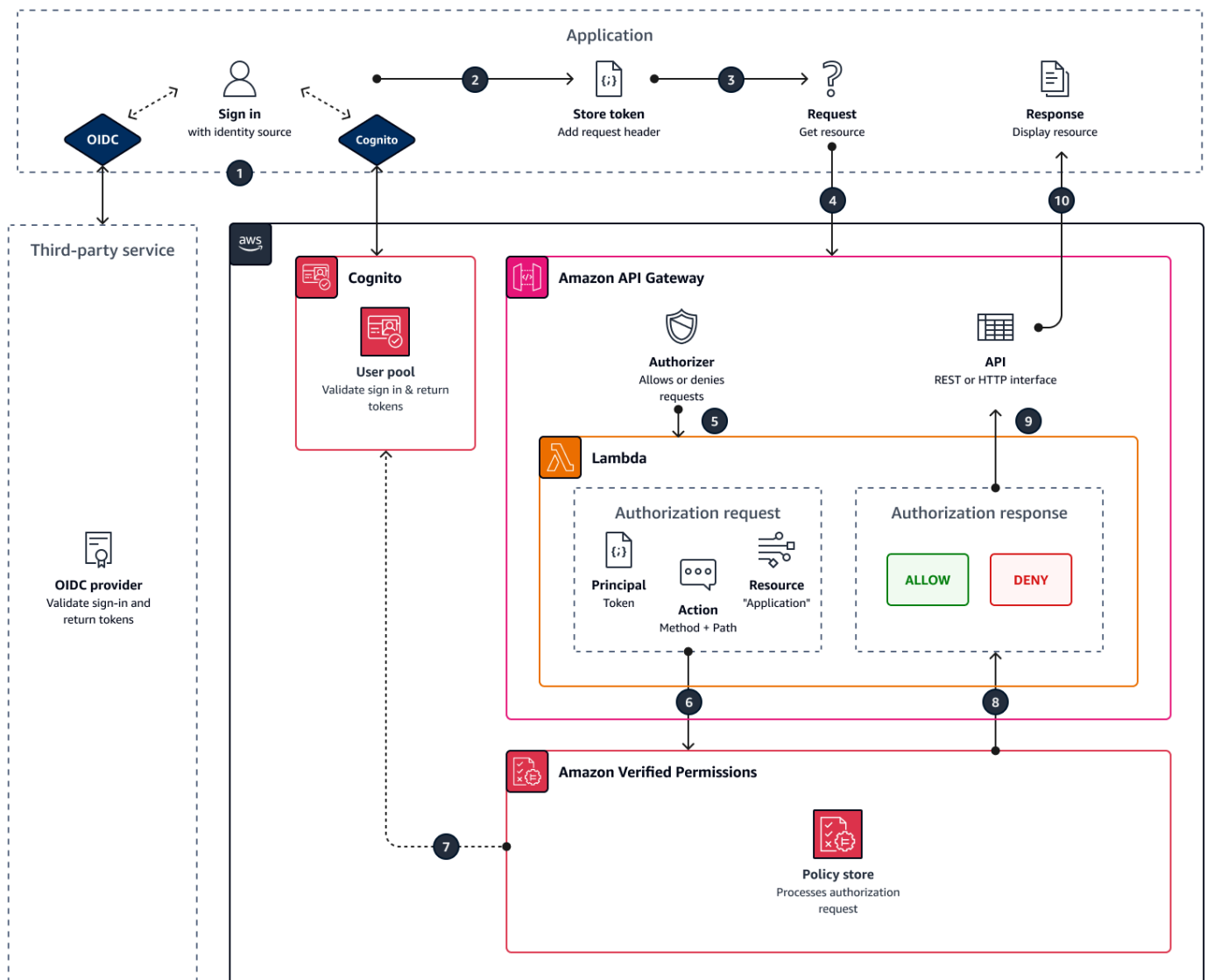
Di toko kebijakan yang ditautkan ke API dan sumber identitas, aplikasi Anda menampilkan token kumpulan pengguna di header otorisasi saat membuat permintaan ke API. Sumber identitas toko kebijakan Anda menyediakan validasi token untuk Izin Terverifikasi. Token membentuk permintaan otorisasi `principal` dalam dengan [IsAuthorizedWithToken](#) API. Izin Terverifikasi membuat kebijakan seputar keanggotaan grup pengguna Anda, seperti yang ditampilkan dalam klaim grup dalam identitas (ID) dan token akses, misalnya `cognito:groups` untuk kumpulan pengguna. API Anda memproses token dari aplikasi Anda di otorisasi Lambda dan mengirimkannya ke Izin Terverifikasi untuk keputusan otorisasi. Saat API Anda menerima keputusan otorisasi dari otorisasi Lambda, API akan meneruskan permintaan tersebut ke sumber data Anda atau menolak permintaan tersebut.

Komponen sumber identitas dan API Gateway otorisasi dengan Izin Terverifikasi

- Kumpulan [Amazon Cognito](#) pengguna atau OIDC iDP yang mengautentikasi dan mengelompokkan pengguna. Token pengguna mengisi keanggotaan grup dan prinsipal atau konteks yang dievaluasi Izin Terverifikasi di toko kebijakan Anda.
- API [API Gateway](#) REST. Izin Terverifikasi mendefinisikan tindakan dari jalur API dan metode API, misalnya. `MyAPI::Action::get /photo`
- Fungsi Lambda dan [otorisasi Lambda](#) untuk API Anda. Fungsi Lambda mengambil token pembawa dari kumpulan pengguna Anda, meminta otorisasi dari Izin Terverifikasi, dan mengembalikan keputusan ke. API Gateway Alur kerja Mengatur dengan API Gateway dan sumber identitas secara otomatis membuat otorisasi Lambda ini untuk Anda.
- Toko kebijakan Izin Terverifikasi. Sumber identitas toko kebijakan adalah kumpulan Amazon Cognito pengguna atau grup penyedia OIDC Anda. Skema penyimpanan kebijakan mencerminkan konfigurasi API Anda, dan kebijakan menautkan grup pengguna ke tindakan API yang diizinkan.
- Aplikasi yang mengautentikasi pengguna dengan IDP Anda dan menambahkan token ke permintaan API.

Bagaimana Izin Terverifikasi mengotorisasi permintaan API

Saat Anda membuat penyimpanan kebijakan baru dan memilih opsi Siapkan dengan API Gateway dan sumber identitas, Izin Terverifikasi akan membuat skema dan kebijakan penyimpanan kebijakan. Skema dan kebijakan mencerminkan tindakan API dan grup pengguna yang ingin Anda otorisasi untuk mengambil tindakan. [Izin Terverifikasi juga menciptakan fungsi dan otorisasi Lambda.](#)



1. Pengguna Anda masuk dengan aplikasi Anda melalui Amazon Cognito atau IDP OIDC lainnya. IDP mengeluarkan ID dan token akses dengan informasi pengguna.
2. Aplikasi Anda menyimpan file JWTs. Untuk informasi selengkapnya, lihat [Menggunakan token dengan kumpulan pengguna](#) di Panduan Amazon Cognito Pengembang..
3. Pengguna Anda meminta data yang harus diambil aplikasi Anda dari API eksternal.
4. Aplikasi Anda meminta data dari REST API di API Gateway. Ini menambahkan ID atau token akses sebagai header permintaan.
5. Jika API Anda memiliki cache untuk keputusan otorisasi, API akan mengembalikan respons sebelumnya. Jika caching dinonaktifkan atau API tidak memiliki cache saat ini, API Gateway meneruskan parameter permintaan ke otorisasi [Lambda berbasis token](#).

6. Fungsi Lambda mengirimkan permintaan otorisasi ke penyimpanan kebijakan Izin Terverifikasi dengan API. [IsAuthorizedWithToken](#) Fungsi Lambda melewati elemen keputusan otorisasi:
 - a. Token pengguna sebagai prinsipal.
 - b. Metode API dikombinasikan dengan jalur API, misalnya `GetPhoto`, sebagai tindakan.
 - c. Istilah `Application` sebagai sumber daya.
7. Izin Terverifikasi memvalidasi token. Untuk informasi selengkapnya tentang cara Amazon Cognito token divalidasi, lihat [Otorisasi dengan Izin Terverifikasi Amazon di Panduan Pengembang Amazon Cognito](#).
8. Izin Terverifikasi mengevaluasi permintaan otorisasi terhadap kebijakan di toko kebijakan Anda dan mengembalikan keputusan otorisasi.
9. Lambda Authorizer mengembalikan `Allow` atau `Deny` respon ke API Gateway
10. API mengembalikan data atau `ACCESS_DENIED` respons terhadap aplikasi Anda. Aplikasi Anda memproses dan menampilkan hasil permintaan API.

Pertimbangan untuk toko kebijakan terkait API

Saat membuat penyimpanan kebijakan terkait API di konsol Izin Terverifikasi, Anda membuat pengujian untuk penerapan produksi pada akhirnya. Sebelum Anda pindah ke produksi, buat konfigurasi tetap untuk API dan kumpulan pengguna Anda. Pertimbangkan faktor-faktor berikut:

API Gateway tanggapan cache

Di toko kebijakan terkait API, Izin Terverifikasi membuat otorisasi Lambda dengan TTL caching Otorisasi 120 detik. Anda dapat menyesuaikan nilai ini atau mematikan caching di otorisasi Anda. Dalam otorisasi dengan caching diaktifkan, otorisasi Anda mengembalikan respons yang sama setiap kali sampai TTL kedaluwarsa. Ini dapat memperpanjang masa efektif token kumpulan pengguna dengan durasi yang sama dengan TTL caching dari tahap yang diminta.

Amazon Cognito kelompok dapat digunakan kembali

Izin Terverifikasi Amazon menentukan keanggotaan grup untuk pengguna kumpulan pengguna dari `cognito:groups` klaim di ID pengguna atau token akses. Nilai klaim ini adalah larik nama ramah grup kumpulan pengguna yang dimiliki pengguna. Anda tidak dapat mengaitkan grup kumpulan pengguna dengan pengenalan unik.

Grup kumpulan pengguna yang Anda hapus dan buat ulang dengan nama yang sama yang ada di toko kebijakan Anda sebagai grup yang sama. Saat Anda menghapus grup dari kumpulan pengguna, hapus semua referensi ke grup dari toko kebijakan Anda.

Namespace dan skema yang diturunkan dari API adalah point-in-time

Izin Terverifikasi menangkap API Anda pada satu titik waktu: hanya menanyakan API Anda saat Anda membuat penyimpanan kebijakan. Ketika skema atau nama API Anda berubah, Anda harus memperbarui penyimpanan kebijakan dan otorisasi Lambda, atau membuat penyimpanan kebijakan terkait API baru. Izin Terverifikasi memperoleh [namespace](#) penyimpanan kebijakan dari nama API Anda.

Fungsi Lambda tidak memiliki konfigurasi VPC

Fungsi Lambda yang dibuat Izin Terverifikasi untuk otorisasi API Anda diluncurkan di VPC default. Secara default, APIs yang memiliki akses jaringan terbatas ke pribadi tidak VPCs dapat berkomunikasi dengan fungsi Lambda yang mengotorisasi permintaan akses dengan Izin Terverifikasi.

Izin Terverifikasi menyebarkan sumber daya otorisasi di CloudFormation

Untuk membuat penyimpanan kebijakan terkait API, Anda harus masuk ke AWS prinsipal yang memiliki hak istimewa tinggi ke konsol Izin Terverifikasi. Pengguna ini menyebarkan CloudFormation tumpukan yang membuat sumber daya di beberapa Layanan AWS. Prinsipal ini harus memiliki izin untuk menambah dan memodifikasi sumber daya di Izin Terverifikasi, IAM, Lambda, dan API Gateway Sebagai praktik terbaik, jangan bagikan kredensial ini dengan administrator lain di organisasi Anda.

Lihat [Pindah ke produksi dengan AWS CloudFormation](#) ikhtisar sumber daya yang dibuat oleh Izin Terverifikasi.

Menambahkan kontrol akses berbasis atribut (ABAC)

Sesi otentikasi khas dengan IDP mengembalikan ID dan token akses. Anda dapat meneruskan salah satu dari jenis token ini sebagai token pembawa dalam permintaan aplikasi ke API Anda. Bergantung pada pilihan Anda saat membuat toko kebijakan, Izin Terverifikasi mengharapkan salah satu dari dua jenis token. Kedua jenis membawa informasi tentang keanggotaan grup pengguna. Untuk informasi selengkapnya tentang jenis token Amazon Cognito, lihat [Menggunakan token dengan kumpulan pengguna](#) di Panduan Amazon Cognito Pengembang.

Setelah membuat toko kebijakan, Anda dapat menambahkan dan memperluas kebijakan. Misalnya, Anda dapat menambahkan grup baru ke kebijakan saat menambahkannya ke kumpulan pengguna. Karena toko kebijakan Anda sudah mengetahui cara kumpulan pengguna menampilkan grup dalam token, Anda dapat mengizinkan serangkaian tindakan untuk grup baru dengan kebijakan baru.

Anda mungkin juga ingin memperluas model evaluasi kebijakan berbasis grup menjadi model yang lebih tepat berdasarkan properti pengguna. Token kumpulan pengguna berisi informasi pengguna tambahan yang dapat berkontribusi pada keputusan otorisasi.

Token ID

Token ID mewakili atribut pengguna dan memiliki kontrol akses berbutir halus tingkat tinggi. Untuk mengevaluasi alamat email, nomor telepon, atau atribut khusus seperti departemen dan manajer, evaluasi token ID.

Token akses

Token akses mewakili izin pengguna dengan cakupan OAuth 2.0. Untuk menambahkan lapisan otorisasi atau mengatur permintaan sumber daya tambahan, evaluasi token akses. Misalnya, Anda dapat memvalidasi bahwa pengguna berada dalam grup yang sesuai dan membawa cakupan seperti `PetStore.read` itu umumnya mengotorisasi akses ke API. Kumpulan pengguna dapat menambahkan cakupan khusus ke token dengan [server sumber daya](#) dan dengan [kustomisasi token saat runtime](#).

Lihat [Memetakan Amazon Cognito token ke skema](#) dan [Memetakan token OIDC ke skema misalnya kebijakan yang memproses klaim dalam ID dan token](#) akses.

Pindah ke produksi dengan AWS CloudFormation

Toko kebijakan terkait API adalah cara untuk membuat model otorisasi untuk API dengan cepat. API Gateway Mereka dirancang untuk berfungsi sebagai lingkungan pengujian untuk komponen otorisasi aplikasi Anda. Setelah membuat toko kebijakan pengujian, luangkan waktu untuk menyempurnakan kebijakan, skema, dan otorisasi Lambda.

Anda dapat menyesuaikan arsitektur API Anda, yang memerlukan penyesuaian yang setara dengan skema dan kebijakan penyimpanan kebijakan Anda. Toko kebijakan yang ditautkan API tidak secara otomatis memperbarui skema mereka dari Arsitektur API—Izin Terverifikasi hanya melakukan polling API pada saat Anda membuat penyimpanan kebijakan. Jika API Anda cukup berubah, Anda mungkin harus mengulangi prosesnya dengan penyimpanan kebijakan baru.

Saat model aplikasi dan otorisasi Anda siap untuk diterapkan ke produksi, integrasikan toko kebijakan terkait API yang Anda kembangkan dengan proses otomatisasi Anda. Sebagai praktik terbaik, kami menyarankan Anda mengeksport skema penyimpanan kebijakan dan kebijakan ke dalam AWS CloudFormation templat yang dapat Anda terapkan ke yang lain Akun AWS dan Wilayah AWS.

Hasil dari proses penyimpanan kebijakan terkait API adalah penyimpanan kebijakan awal dan otorisasi Lambda. Otorisasi Lambda memiliki beberapa sumber daya yang bergantung. Izin Terverifikasi menyebarkan sumber daya ini dalam tumpukan yang dibuat secara otomatis CloudFormation. Untuk menyebarkan ke produksi, Anda harus mengumpulkan penyimpanan kebijakan dan sumber daya otorisasi Lambda ke dalam templat. Toko kebijakan terkait API dibuat dari sumber daya berikut:

1. [AWS::VerifiedPermissions::PolicyStore](#): Salin skema Anda ke SchemaDefinition objek. "Karakter melarikan diri sebagai\".
2. [AWS::VerifiedPermissions::IdentitySource](#): Salin nilai dari output [GetIdentitySource](#) dari penyimpanan kebijakan pengujian Anda dan modifikasi sesuai kebutuhan.
3. Satu atau lebih dari [AWS::VerifiedPermissions::Policy](#): Salin pernyataan kebijakan Anda ke Definition objek. "Karakter melarikan diri sebagai\".
4. [AWS::Lambda::Function](#), [AWS::Peran::IAM::Kebijakan](#), [AWS::IAM::Authorizer](#), [AWS::ApiGateway::Lambda::Permission](#)

Template berikut adalah contoh toko kebijakan. Anda dapat menambahkan sumber daya otorisasi Lambda dari tumpukan yang ada ke template ini.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyExamplePolicyStore": {
      "Type": "AWS::VerifiedPermissions::PolicyStore",
      "Properties": {
        "ValidationSettings": {
          "Mode": "STRICT"
        },
        "Description": "ApiGateway: PetStore/test",
        "Schema": {
          "CedarJson": "{\"PetStore\":{\"actions\":{\"get /pets\": {
            \"appliesTo\":{\"principalTypes\":[\"User\"],\"resourceTypes\":[\"Application\"],
            \"context\":{\"type\":\"Record\",\"attributes\":{\"}}}},\"get /\": {\"appliesTo\": {
            \"principalTypes\":[\"User\"],\"resourceTypes\":[\"Application\"],\"context\":{\"type
```

```

\":"Record","\attributes\":"{}}}},\get /pets/{petId}\":"{\appliesTo\":"{\context
\":"{\type\":"Record","\attributes\":"{}}},\resourceTypes\":"[Application]",
\principalTypes\":"[User]"}},\post /pets\":"{\appliesTo\":"{\principalTypes\":"
[User]",\resourceTypes\":"[Application]",\context\":"{\type\":"Record",
\attributes\":"{}}}}},\entityTypes\":"{\Application\":"{\shape\":"{\type\":"Record",
\attributes\":"{}}},\User\":"{\memberOfTypes\":"[UserGroup]",\shape\":"{\attributes
\":"{,\type\":"Record"}},\UserGroup\":"{\shape\":"{\type\":"Record",\attributes
\":"{}}}}}}}"
    }
  }
},
"MyExamplePolicy": {
  "Type": "AWS::VerifiedPermissions::Policy",
  "Properties": {
    "Definition": {
      "Static": {
        "Description": "Policy defining permissions for testgroup
cognito group",
        "Statement": "permit(\nprincipal in PetStore::UserGroup::
\"us-east-1_EXAMPLE|testgroup\", \naction in [\n PetStore::Action::\"get /\",
\n PetStore::Action::\"post /pets\", \n PetStore::Action::\"get /pets\", \n
PetStore::Action::\"get /pets/{petId}\" \n], \nresource);"
      }
    },
    "PolicyStoreId": {
      "Ref": "MyExamplePolicyStore"
    }
  },
  "DependsOn": [
    "MyExamplePolicyStore"
  ]
},
"MyExampleIdentitySource": {
  "Type": "AWS::VerifiedPermissions::IdentitySource",
  "Properties": {
    "Configuration": {
      "CognitoUserPoolConfiguration": {
        "ClientIds": [
          "1example23456789"
        ],
        "GroupConfiguration": {
          "GroupEntityType": "PetStore::UserGroup"
        }
      },

```

```
        "UserPoolArn": "arn:aws:cognito-idp:us-
east-1:123456789012:userpool/us-east-1_EXAMPLE"
    },
    "PolicyStoreId": {
        "Ref": "MyExamplePolicyStore"
    },
    "PrincipalEntityType": "PetStore::User"
},
"DependsOn": [
    "MyExamplePolicyStore"
]
}
}
```

Pemecahan masalah toko kebijakan terkait API

Gunakan informasi di sini untuk membantu Anda mendiagnosis dan memperbaiki masalah umum saat membuat penyimpanan kebijakan terkait API Izin Terverifikasi Amazon.

Topik

- [Saya memperbarui kebijakan saya tetapi keputusan otorisasi tidak berubah](#)
- [Saya melampirkan otorisasi Lambda ke API saya tetapi tidak menghasilkan permintaan otorisasi](#)
- [Saya menerima keputusan otorisasi yang tidak terduga dan ingin meninjau logika otorisasi](#)
- [Saya ingin menemukan log dari otorisasi Lambda saya](#)
- [Otorisasi Lambda saya tidak ada](#)
- [API saya ada di VPC pribadi dan tidak dapat memanggil otorisasi](#)
- [Saya ingin memproses atribut pengguna tambahan dalam model otorisasi saya](#)
- [Saya ingin menambahkan tindakan baru, atribut konteks tindakan, atau atribut sumber daya](#)

Saya memperbarui kebijakan saya tetapi keputusan otorisasi tidak berubah

Secara default, Izin Terverifikasi mengonfigurasi otorisasi Lambda untuk menyimpan keputusan otorisasi cache selama 120 detik. Coba lagi setelah dua menit, atau nonaktifkan cache pada otorisasi Anda. Untuk informasi selengkapnya, lihat [Mengaktifkan cache API untuk meningkatkan daya tanggap](#) di Panduan Pengembang Amazon API Gateway.

Saya melampirkan otorisasi Lambda ke API saya tetapi tidak menghasilkan permintaan otorisasi

Untuk mulai memproses permintaan, Anda harus menerapkan tahap API tempat Anda melampirkan otorisasi. Untuk informasi selengkapnya, lihat [Menerapkan REST API](#) di Panduan Pengembang Amazon API Gateway.

Saya menerima keputusan otorisasi yang tidak terduga dan ingin meninjau logika otorisasi

Proses penyimpanan kebijakan terkait API membuat fungsi Lambda untuk otorisasi Anda. Izin Terverifikasi secara otomatis membangun logika keputusan otorisasi Anda ke dalam fungsi otorisasi. Anda dapat kembali setelah membuat toko kebijakan untuk meninjau dan memperbarui logika dalam fungsi.

Untuk menemukan fungsi Lambda Anda dari AWS CloudFormation konsol, pilih tombol Periksa penerapan di halaman Ikhtisar toko kebijakan baru Anda.

Anda juga dapat menemukan fungsi Anda di AWS Lambda konsol. Arahkan ke konsol di AWS Region toko kebijakan Anda dan cari nama fungsi dengan awalan. AVPAuthorizerLambda Jika Anda telah membuat lebih dari satu toko kebijakan terkait API, gunakan Waktu modifikasi terakhir fungsi Anda untuk menghubungkannya dengan pembuatan toko kebijakan.

Saya ingin menemukan log dari otorisasi Lambda saya

Fungsi Lambda mengumpulkan metrik dan mencatat hasil pemanggilannya di Amazon. CloudWatch Untuk meninjau log Anda, [cari fungsi Anda](#) di konsol Lambda dan pilih tab Monitor. Pilih Lihat CloudWatch log dan tinjau entri dalam grup log.

Untuk informasi selengkapnya tentang log fungsi Lambda, lihat Menggunakan [CloudWatch Log Amazon dengan AWS Lambda](#) di Panduan AWS Lambda Pengembang.

Otorisasi Lambda saya tidak ada

Setelah menyelesaikan penyiapan penyimpanan kebijakan terkait API, Anda harus melampirkan otorisasi Lambda ke API Anda. Jika Anda tidak dapat menemukan otorisasi di API Gateway konsol, sumber daya tambahan untuk penyimpanan kebijakan Anda mungkin gagal atau belum diterapkan. Toko kebijakan terkait API menyebarkan sumber daya ini dalam tumpukan. CloudFormation

Izin Terverifikasi menampilkan tautan dengan label Periksa penerapan di akhir proses pembuatan. Jika Anda sudah menavigasi jauh dari layar ini, buka CloudFormation konsol dan cari tumpukan

terbaru untuk nama yang diawali. AVPAuthorizer-<policy store ID> CloudFormation menyediakan informasi pemecahan masalah yang berharga dalam output penyebaran tumpukan.

Untuk bantuan mengatasi masalah CloudFormation tumpukan, lihat [Pemecahan masalah CloudFormation](#) di Panduan Pengguna.AWS CloudFormation

API saya ada di VPC pribadi dan tidak dapat memanggil otorisasi

Izin Terverifikasi tidak mendukung akses ke otorisasi Lambda melalui titik akhir VPC. Anda harus membuka jalur jaringan antara API Anda dan fungsi Lambda yang berfungsi sebagai otorisasi Anda.

Saya ingin memproses atribut pengguna tambahan dalam model otorisasi saya

Proses penyimpanan kebijakan terkait API memperoleh kebijakan Izin Terverifikasi dari klaim grup dalam token pengguna. Untuk memperbarui model otorisasi Anda untuk mempertimbangkan atribut pengguna tambahan, integrasikan atribut tersebut dalam kebijakan Anda.

Anda dapat memetakan banyak klaim dalam ID dan token akses dari kumpulan pengguna Amazon Cognito ke pernyataan kebijakan Izin Terverifikasi. Misalnya, sebagian besar pengguna memiliki email klaim dalam token ID mereka. Untuk informasi selengkapnya tentang menambahkan klaim dari sumber identitas Anda ke kebijakan, lihat [Memetakan Amazon Cognito token ke skema](#) dan [Memetakan token OIDC ke skema](#).

Saya ingin menambahkan tindakan baru, atribut konteks tindakan, atau atribut sumber daya

Toko kebijakan terkait API dan otorisasi Lambda yang dibuatnya adalah sumber daya. point-in-time Mereka mencerminkan status API Anda pada saat pembuatan. Skema penyimpanan kebijakan tidak menetapkan atribut konteks apa pun ke tindakan, atau atribut atau induk apa pun ke sumber daya defaultApplication.

Saat menambahkan tindakan—jalur dan metode—ke API, Anda harus memperbarui penyimpanan kebijakan agar mengetahui tindakan baru tersebut. Anda juga harus memperbarui otorisasi Lambda Anda untuk memproses permintaan otorisasi untuk tindakan baru. Anda dapat [memulai lagi dengan toko kebijakan baru](#) atau Anda dapat memperbarui toko kebijakan yang ada.

Untuk memperbarui toko kebijakan yang ada, [cari fungsi Anda](#). Periksa logika dalam fungsi yang dihasilkan secara otomatis dan perbarui untuk memproses tindakan, atribut, atau konteks baru. Kemudian [edit skema Anda](#) untuk menyertakan tindakan dan atribut baru.

Menghapus toko kebijakan

Anda dapat menghapus toko kebijakan Izin Terverifikasi Amazon menggunakan Konsol Manajemen AWS atau AWS CLI. Menghapus penyimpanan kebijakan akan menghapus skema dan kebijakan serta templat kebijakan apa pun secara permanen di toko kebijakan. Alias penyimpanan kebijakan apa pun yang terkait dengan penyimpanan kebijakan yang dihapus juga akan dihapus.

Perlindungan penghapusan mencegah penghapusan toko kebijakan yang tidak disengaja.

Perlindungan penghapusan diaktifkan pada semua toko kebijakan baru yang dibuat melalui Konsol Manajemen AWS. Sebaliknya, ini dinonaktifkan untuk semua penyimpanan kebijakan yang dibuat melalui panggilan API atau SDK.

Anda mungkin ingin menghapus toko kebijakan karena alasan berikut:

- Anda telah mencapai kuota toko polis yang tersedia di Wilayah tertentu. Untuk informasi selengkapnya, lihat [Kuota untuk sumber daya](#).
- Anda tidak lagi mendukung penyewa dalam aplikasi multi-penyewa dan, oleh karena itu, tidak lagi membutuhkan toko kebijakan itu.

Konsol Manajemen AWS

Untuk menghapus toko kebijakan

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Pengaturan.
3. Pilih Hapus toko kebijakan ini.
4. `delete` Ketik kotak teks dan pilih Hapus.

Note

Jika perlindungan penghapusan diaktifkan, Anda harus menonaktifkannya sebelum dapat memilih Hapus. Untuk menonaktifkannya, pilih Nonaktifkan perlindungan penghapusan.

AWS CLI

Untuk menghapus toko kebijakan

Anda dapat menghapus penyimpanan kebijakan dengan menggunakan `delete-policy-store` operasi, menggantinya *PSEXAMPLEabcdefg111111* dengan ID penyimpanan kebijakan yang ingin Anda hapus.

```
$ aws verifiedpermissions delete-policy-store \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Jika berhasil, perintah ini tidak menghasilkan output.

Note

Jika perlindungan penghapusan diaktifkan untuk penyimpanan kebijakan ini, Anda harus terlebih dahulu menjalankan `update-policy-store` operasi dan menonaktifkan perlindungan penghapusan.

```
aws verifiedpermissions update-policy-store \  
  --deletion-protection "DISABLED" \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Alias penyimpanan kebijakan Izin Terverifikasi Amazon

Alias toko kebijakan adalah nama yang ramah untuk toko kebijakan. Misalnya, alias toko kebijakan memungkinkan Anda merujuk ke toko kebijakan menggunakan `policy-store-alias/example-policy-store` alih-alih `PSEXAMPLEabcdefg111111`. Alias penyimpanan kebijakan dapat digunakan dalam operasi Izin Terverifikasi apa pun yang menerima parameter input `policyStoreId`.

Anda dapat membuat alias penyimpanan kebijakan untuk penyimpanan kebijakan menggunakan `CreatePolicyStoreAlias` API atau dengan menggunakan `AWS::VerifiedPermissions::PolicyStoreAlias` CloudFormation sumber daya.

Amazon Verified Permissions API menyediakan kontrol penuh atas alias penyimpanan kebijakan di masing-masing Akun AWS dan Wilayah. API mencakup operasi untuk membuat alias penyimpanan kebijakan (`CreatePolicyStoreAlias`), melihat nama alias penyimpanan kebijakan dan alias penyimpanan kebijakan ARNs (`GetPolicyStoreAlias`, `ListPolicyStoreAliases`), dan menghapus alias penyimpanan kebijakan (`DeletePolicyStoreAlias`).

Topik

- [Properti alias toko kebijakan](#)
- [Membuat alias penyimpanan kebijakan Izin Terverifikasi Amazon](#)
- [Mengambil alias penyimpanan kebijakan Izin Terverifikasi Amazon](#)
- [Menghapus alias penyimpanan kebijakan Izin Terverifikasi Amazon](#)
- [Menggunakan alias penyimpanan kebijakan Izin Terverifikasi Amazon dalam operasi API](#)
- [Mengontrol akses ke alias penyimpanan kebijakan](#)

Properti alias toko kebijakan

Cara kerja alias penyimpanan kebijakan di Izin Terverifikasi Amazon.

Alias toko kebijakan adalah sumber daya independen AWS

Alias toko kebijakan bukan milik toko kebijakan. Tindakan yang Anda lakukan pada alias toko kebijakan tidak memengaruhi penyimpanan kebijakan terkait. Anda dapat menghapus alias toko kebijakan tanpa efek apa pun pada toko kebijakan terkait. Jika Anda menghapus penyimpanan

kebijakan, semua alias penyimpanan kebijakan yang terkait dengan penyimpanan kebijakan tersebut juga akan dihapus.

Setiap alias toko kebijakan memiliki Nama Sumber Daya Amazon (ARN) yang secara unik mengidentifikasi alias toko kebijakan. Jika Anda menetapkan alias penyimpanan kebijakan sebagai sumber daya dalam kebijakan IAM, kebijakan tersebut merujuk ke alias penyimpanan kebijakan, bukan ke penyimpanan kebijakan terkait.

Setiap alias toko kebijakan memiliki dua format

Saat membuat alias penyimpanan kebijakan, Anda menentukan nama alias penyimpanan kebijakan. Izin Terverifikasi Amazon membuat alias penyimpanan kebijakan ARN untuk Anda.

- Alias toko kebijakan ARN adalah Nama Sumber Daya Amazon (ARN) yang secara unik mengidentifikasi alias toko kebijakan.

```
# Alias ARN
arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/example-policy-store
```

- Nama alias toko kebijakan yang unik di Akun AWS dan Wilayah. Di Amazon Verified Permissions API, nama alias penyimpanan kebijakan selalu diawali oleh `policy-store-alias/`

```
# Alias name
policy-store-alias/example-policy-store
```

Alias toko kebijakan tidak rahasia

Alias penyimpanan kebijakan dapat ditampilkan dalam teks biasa di CloudTrail log dan output lainnya. Jangan sertakan informasi rahasia atau sensitif dalam nama alias toko kebijakan.

Setiap alias toko kebijakan dikaitkan dengan satu toko kebijakan pada satu waktu

Alias toko kebijakan dan toko kebijakan terkait harus milik yang sama Akun AWS dan Wilayah. Anda dapat mengaitkan alias toko kebijakan dengan toko kebijakan apa pun di area yang sama Akun AWS dan Wilayah.

Misalnya, `ListPolicyStoreAliases` output ini menunjukkan bahwa alias penyimpanan `example-policy-store` kebijakan dikaitkan dengan tepat satu penyimpanan kebijakan target, yang diwakili oleh `policyStoreId` properti.

```
{
  "aliasName": "policy-store-alias/example-policy-store",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store",
  "createdAt": "2024-01-15T12:30:00.000000+00:00",
  "state": "Active"
}
```

Beberapa alias dapat dikaitkan dengan toko kebijakan yang sama

Misalnya, Anda dapat mengaitkan `example-policy-store` dan `example-policy-store-2` alias dengan toko kebijakan yang sama.

```
[
  {
    "aliasName": "policy-store-alias/example-policy-store",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store",
    "createdAt": "2024-01-15T12:30:00.000000+00:00",
    "state": "Active"
  },
  {
    "aliasName": "policy-store-alias/example-policy-store-2",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store-2",
    "createdAt": "2024-01-16T09:15:00.000000+00:00",
    "state": "Active"
  }
]
```

Alias toko kebijakan harus unik di Wilayah Akun AWS dan

Misalnya, Anda hanya dapat memiliki satu alias toko kebijakan dengan nama `example-policy-store` di masing-masing Akun AWS dan Wilayah. Alias toko kebijakan peka huruf besar/kecil. Anda tidak dapat mengubah nama alias toko kebijakan. Namun, Anda dapat menghapus alias toko kebijakan dan membuat alias toko kebijakan baru dengan nama yang diinginkan setelah periode reservasi 24 jam berakhir.

Anda dapat membuat alias penyimpanan kebijakan dengan nama yang sama di Wilayah yang berbeda. Setiap alias toko kebijakan akan memiliki ARN yang unik. Jika kode Anda mengacu pada nama alias toko kebijakan seperti `policy-store-alias/example-policy-store`, Anda dapat menjalankannya di beberapa Wilayah. Di setiap Wilayah, ia menggunakan toko kebijakan yang berbeda.

Alias toko kebijakan dihapus dengan lembut

Ketika alias toko kebijakan dihapus, nama alias toko kebijakan dicadangkan untuk jangka waktu 24 jam. Jika Anda mencoba membuat alias toko kebijakan dengan nama yang sama selama periode ini, permintaan akan ditolak. Selama periode ini, `GetPolicyStoreAlias` mengembalikan alias toko kebijakan dengan `PendingDeletion` negara.

Anda dapat menggunakan alias untuk mengidentifikasi toko kebijakan

Anda dapat menggunakan alias penyimpanan kebijakan untuk mengidentifikasi penyimpanan kebijakan di semua operasi yang menerima `policyStoreId` (misalnya, `IsAuthorized`). Dalam kasus seperti itu, nama alias toko kebijakan harus diawali dengan `policy-store-alias/`. Alias penyimpanan kebijakan tidak dapat digunakan untuk mengidentifikasi penyimpanan kebijakan untuk `DeletePolicyStore` operasi.

Anda tidak dapat menggunakan nama alias toko kebijakan atau alias penyimpanan kebijakan ARN untuk mengidentifikasi penyimpanan kebijakan dalam Resource elemen kebijakan IAM. Untuk mengontrol akses ke toko kebijakan saat direferensikan melalui alias toko kebijakan, lihat [Mengontrol akses ke alias penyimpanan kebijakan](#)

Membuat alias penyimpanan kebijakan Izin Terverifikasi Amazon

Anda dapat membuat alias toko kebijakan untuk mereferensikan toko kebijakan menggunakan nama yang ramah. Nama alias toko kebijakan harus unik per Akun AWS dan Wilayah. Alias toko kebijakan hanya dapat dikaitkan dengan toko kebijakan yang dimiliki oleh yang sama Akun AWS dan aktif di Wilayah yang sama dengan alias toko kebijakan. Alias toko kebijakan adalah sumber daya terpisah dengan otorisasi mereka sendiri ARNs dan IAM.

Secara default, hanya 10 alias penyimpanan kebijakan yang dapat dikaitkan dengan penyimpanan kebijakan yang sama.

Note

`CreatePolicyStoreAlias` adalah idempoten. Jika Anda memanggil `CreatePolicyStoreAlias` operasi dengan nama alias penyimpanan kebijakan dan ID penyimpanan kebijakan yang cocok dengan alias penyimpanan kebijakan yang ada, `CreatePolicyStoreAlias` operasi akan berhasil dan mengembalikan alias penyimpanan kebijakan yang ada. Namun, jika Anda memanggil `CreatePolicyStoreAlias` operasi dengan nama alias penyimpanan kebijakan yang ada tetapi ID penyimpanan kebijakan yang berbeda, operasi akan mengembalikan nama. `ConflictException`

AWS CLI

Untuk membuat alias toko kebijakan

Anda dapat membuat alias toko kebijakan dengan menggunakan [CreatePolicyStoreAlias](#) operasi. Contoh berikut membuat alias toko kebijakan dengan nama `example-policy-store`.

```
$ aws verifiedpermissions create-policy-store-alias \
  --alias-name policy-store-alias/example-policy-store \
  --policy-store-id PSEXAMPLEabcdefg111111
{
  "aliasName": "policy-store-alias/example-policy-store",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store",
  "createdAt": "2024-01-15T12:30:00.000000+00:00"
}
```

Mengambil alias penyimpanan kebijakan Izin Terverifikasi Amazon

Anda dapat mengambil informasi tentang alias penyimpanan kebijakan menggunakan `GetPolicyStoreAlias` operasi untuk mendapatkan detail tentang alias penyimpanan kebijakan tertentu, atau `ListPolicyStoreAliases` operasi untuk mencantumkan semua alias penyimpanan kebijakan di dan Wilayah Anda. Akun AWS

Mendapatkan alias toko kebijakan

Gunakan `GetPolicyStoreAlias` operasi untuk mengambil detail tentang alias penyimpanan kebijakan tertentu, termasuk ID penyimpanan kebijakan terkait.

AWS CLI

Untuk mengambil detail tentang alias toko kebijakan

Anda dapat mengambil alias toko kebijakan dengan menggunakan operasi. [GetPolicyStoreAlias](#) Contoh berikut mengambil rincian untuk alias toko kebijakan dengan nama. `example-policy-store`

```
$ aws verifiedpermissions get-policy-store-alias \
  --alias-name policy-store-alias/example-policy-store
{
  "aliasName": "policy-store-alias/example-policy-store",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store",
  "createdAt": "2024-01-15T12:30:00.000000+00:00",
  "state": "Active"
}
```

Daftar alias toko kebijakan

Gunakan `ListPolicyStoreAliases` operasi untuk mencantumkan semua alias penyimpanan kebijakan di Wilayah Akun AWS dan Anda. Anda dapat menggunakan filter parameter untuk hanya mencantumkan alias penyimpanan kebijakan yang terkait dengan penyimpanan kebijakan tertentu.

AWS CLI

Untuk mencantumkan semua alias toko kebijakan

Anda dapat membuat daftar alias penyimpanan kebijakan dengan menggunakan [ListPolicyStoreAliases](#) operasi. Contoh berikut mencantumkan semua alias toko kebijakan yang dimiliki oleh 123456789012 Akun AWS di Wilayah us-west-2.

```
$ aws verifiedpermissions list-policy-store-aliases
{
```

```

"policyStoreAliases": [
  {
    "aliasName": "policy-store-alias/example-policy-store",
    "policyStoreId": "PSEXAMPLEEabcdefg111111",
    "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-
store-alias/example-policy-store",
    "createdAt": "2024-01-15T12:30:00.000000+00:00",
    "state": "Active"
  },
  {
    "aliasName": "policy-store-alias/example-policy-store-2",
    "policyStoreId": "PSEXAMPLEEabcdefg111111",
    "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-
store-alias/example-policy-store-2",
    "createdAt": "2024-01-16T09:15:00.000000+00:00",
    "state": "Active"
  },
  {
    "aliasName": "policy-store-alias/example-policy-store-3",
    "policyStoreId": "PSEXAMPLEEabcdefg222222",
    "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-
store-alias/example-policy-store-3",
    "createdAt": "2024-01-17T14:45:00.000000+00:00",
    "state": "Active"
  }
]
}

```

Untuk membuat daftar alias penyimpanan kebijakan untuk toko kebijakan tertentu

Gunakan filter parameter untuk mencantumkan hanya alias yang terkait dengan penyimpanan kebijakan tertentu.

```

$ aws verifiedpermissions list-policy-store-aliases \
  --filter '{"policyStoreId": "PSEXAMPLEEabcdefg111111"}'
{
  "policyStoreAliases": [
    {
      "aliasName": "policy-store-alias/example-policy-store",
      "policyStoreId": "PSEXAMPLEEabcdefg111111",
      "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-
store-alias/example-policy-store",
      "createdAt": "2024-01-15T12:30:00.000000+00:00",

```

```
        "state": "Active"
      },
      {
        "aliasName": "policy-store-alias/example-policy-store-2",
        "policyStoreId": "PSEXAMPLEEabcdefg111111",
        "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-
store-alias/example-policy-store-2",
        "createdAt": "2024-01-16T09:15:00.000000+00:00",
        "state": "Active"
      }
    ]
  }
}
```

Menghapus alias penyimpanan kebijakan Izin Terverifikasi Amazon

Anda dapat menghapus alias toko kebijakan ketika tidak lagi diperlukan. Menghapus alias penyimpanan kebijakan tidak memengaruhi penyimpanan kebijakan terkait. Menghapus toko kebijakan akan menghapus semua alias penyimpanan kebijakan yang terkait dengan penyimpanan kebijakan tersebut.

Setelah Anda menghapus alias toko kebijakan, nama alias toko kebijakan dicadangkan selama 24 jam dan tidak dapat digunakan kembali selama periode ini.

AWS CLI

Untuk menghapus alias toko kebijakan

Anda dapat menghapus alias toko kebijakan dengan menggunakan [DeletePolicyStoreAlias](#) operasi. Contoh berikut menghapus alias toko kebijakan dengan nama `example-policy-store`

```
$ aws verifiedpermissions delete-policy-store-alias \
  --alias-name policy-store-alias/example-policy-store
```

Menggunakan alias penyimpanan kebijakan Izin Terverifikasi Amazon dalam operasi API

Setiap operasi Izin Terverifikasi Amazon yang menerima `policyStoreId` parameter, seperti [IsAuthorized](#), [IsAuthorizedWithToken](#), dan [GetPolicyStore](#), dapat menerima nama alias penyimpanan kebijakan sebagai pengganti ID penyimpanan kebijakan.

Important

Bila Anda menggunakan alias penyimpanan kebijakan sebagai nilai `policyStoreId` parameter, Anda harus menyertakan `policy-store-alias/` awalan. Misalnya, gunakan `policy-store-alias/example-policy-store`, tidak `example-policy-store`.

Menggunakan alias toko Kebijakan dalam Operasi

`IsAuthorized`Perintah berikut menggunakan alias penyimpanan kebijakan dengan nama `example-policy-store` untuk mengidentifikasi toko kebijakan.

AWS CLI

```
$ aws verifiedpermissions is-authorized \  
  --policy-store-id policy-store-alias/example-policy-store \  
  --principal entityType=User,entityId=alice \  
  --action actionType=Action,actionId=view \  
  --resource entityType=Photo,entityId=photo123
```

Note

Anda tidak dapat menggunakan alias penyimpanan kebijakan sebagai pengganti `policyStoreId` bidang untuk [DeletePolicyStore](#) operasi.

Menggunakan alias toko Kebijakan Di Seluruh Wilayah AWS

Salah satu penggunaan alias yang paling kuat adalah pada aplikasi yang berjalan dalam beberapa Wilayah AWS. Misalnya, Anda mungkin memiliki aplikasi global yang menggunakan toko kebijakan berbeda di setiap Wilayah.

- Di us-east-1, Anda ingin menggunakan. PSEXAMPLEabcdefg111111
- Di eu-west-1, Anda ingin menggunakan. PSEXAMPLEabcdefg222222

Anda dapat membuat versi aplikasi yang berbeda di setiap Wilayah atau menggunakan kamus atau pernyataan switch untuk memilih penyimpanan kebijakan yang tepat untuk setiap Wilayah. Tetapi jauh lebih mudah untuk membuat alias toko kebijakan dengan nama alias toko kebijakan yang sama di setiap Wilayah. Ingat bahwa nama alias toko kebijakan peka huruf besar/kecil.

AWS CLI

```
$ aws --region us-east-1 verifiedpermissions create-policy-store-alias \
  --alias-name policy-store-alias/my-app \
  --policy-store-id PSEXAMPLEabcdefg111111

$ aws --region eu-west-1 verifiedpermissions create-policy-store-alias \
  --alias-name policy-store-alias/my-app \
  --policy-store-id PSEXAMPLEabcdefg222222
```

Kemudian, gunakan alias penyimpanan kebijakan dalam kode Anda. Ketika kode Anda berjalan di setiap Wilayah, alias penyimpanan kebijakan akan merujuk ke penyimpanan kebijakan terkait di Wilayah tersebut.

AWS CLI

```
$ aws verifiedpermissions is-authorized \
  --policy-store-id policy-store-alias/my-app \
  --principal entityType=User,entityId=alice \
  --action actionType=Action,actionId=view \
  --resource entityType=Photo,entityId=photo123
```

Namun, ada risiko bahwa alias toko kebijakan mungkin dihapus. Dalam hal ini, upaya aplikasi untuk menggunakan nama alias toko kebijakan akan gagal, dan Anda mungkin perlu membuat ulang

atau memperbarui alias penyimpanan kebijakan. Untuk mengurangi risiko ini, berhati-hatilah dalam memberikan izin kepada kepala sekolah untuk mengelola alias toko kebijakan yang Anda gunakan dalam aplikasi Anda.

Mengontrol akses ke alias penyimpanan kebijakan

Prinsipal yang mengelola alias toko kebijakan harus memiliki izin untuk berinteraksi dengan alias penyimpanan kebijakan tersebut dan, untuk beberapa operasi, penyimpanan kebijakan yang dikaitkan dengan alias penyimpanan kebijakan. Anda dapat memberikan izin ini menggunakan IAM kebijakan.

Bagian berikut menjelaskan izin yang diperlukan untuk membuat dan mengelola alias penyimpanan kebijakan.

izin terverifikasi: `CreatePolicyStoreAlias`

Untuk membuat alias penyimpanan kebijakan, prinsipal memerlukan izin berikut untuk alias penyimpanan kebijakan dan untuk penyimpanan kebijakan terkait.

- `verifiedpermissions:CreatePolicyStoreAlias` untuk alias toko kebijakan. Berikan izin ini dalam IAM kebijakan yang dilampirkan pada kepala sekolah yang diizinkan untuk membuat alias toko kebijakan.

Contoh pernyataan kebijakan berikut menentukan alias penyimpanan kebijakan tertentu dalam elemen. `Resource` Tetapi Anda dapat mencantumkan beberapa alias penyimpanan kebijakan ARNs atau menentukan pola alias toko kebijakan, seperti. `"sample*"` Anda juga dapat menentukan `Resource` nilai `"*"` untuk mengizinkan prinsipal membuat alias penyimpanan kebijakan apa pun di Wilayah Akun AWS dan.

```
{
  "Sid": "IAMPolicyForCreateAlias",
  "Effect": "Allow",
  "Action": "verifiedpermissions:CreatePolicyStoreAlias",
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/example-policy-store"
}
```

- `verifiedpermissions:CreatePolicyStoreAlias` untuk toko kebijakan terkait. Izin ini harus diberikan dalam suatu IAM kebijakan.

```
{
  "Sid": "PolicyStorePermissionForAlias",
  "Effect": "Allow",
  "Action": "verifiedpermissions:CreatePolicyStoreAlias",
  "Resource": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
}
```

izin terverifikasi: GetPolicyStoreAlias

Untuk mendapatkan rincian tentang alias toko kebijakan tertentu, kepala sekolah harus memiliki `verifiedpermissions:GetPolicyStoreAlias` izin untuk alias toko kebijakan dalam kebijakan IAM

Contoh pernyataan kebijakan berikut memberikan izin utama untuk mendapatkan alias toko kebijakan tertentu.

```
{
  "Sid": "IAMPolicyForGetAlias",
  "Effect": "Allow",
  "Action": "verifiedpermissions:GetPolicyStoreAlias",
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/
example-policy-store"
}
```

izin terverifikasi: ListPolicyStoreAliases

Untuk membuat daftar alias penyimpanan kebijakan di Wilayah Akun AWS dan, kepala sekolah harus memiliki `verifiedpermissions:ListPolicyStoreAliases` izin dalam IAM kebijakan. Karena kebijakan ini tidak terkait dengan penyimpanan kebijakan tertentu atau sumber alias penyimpanan kebijakan, nilai elemen sumber daya dalam kebijakan harus "*" .

Misalnya, pernyataan IAM kebijakan berikut memberikan izin utama untuk mencantumkan semua alias penyimpanan kebijakan di Akun AWS.

```
{
  "Sid": "IAMPolicyForListingAliases",
  "Effect": "Allow",
```

```
"Action": "verifiedpermissions:ListPolicyStoreAliases",  
"Resource": "*" } }
```

izin terverifikasi: DeletePolicyStoreAlias

Untuk menghapus alias toko kebijakan, prinsipal memerlukan izin hanya untuk alias toko kebijakan.

Note

Menghapus alias toko kebijakan tidak berpengaruh pada penyimpanan kebijakan terkait, meskipun aplikasi yang mereferensikan alias toko kebijakan akan menerima kesalahan. Jika Anda salah menghapus alias toko kebijakan, Anda dapat membuatnya kembali setelah periode reservasi 24 jam.

Kepala sekolah membutuhkan `verifiedpermissions>DeletePolicyStoreAlias` izin untuk alias toko kebijakan. Berikan izin ini dalam IAM kebijakan yang dilampirkan pada kepala sekolah yang diizinkan untuk menghapus alias toko kebijakan.

Contoh pernyataan kebijakan berikut menentukan alias penyimpanan kebijakan dalam elemen. Resource Tetapi Anda dapat mencantumkan beberapa alias penyimpanan kebijakan ARNs atau menentukan pola alias toko kebijakan, seperti. `"sample*"` Anda juga dapat menentukan Resource nilai `"*"` untuk mengizinkan prinsipal menghapus alias penyimpanan kebijakan apa pun di Wilayah Akun AWS dan.

```
{  
  "Sid": "IAMPolicyForDeleteAlias",  
  "Effect": "Allow",  
  "Action": "verifiedpermissions>DeletePolicyStoreAlias",  
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/  
example-policy-store"  
}
```

Membatasi Izin alias penyimpanan Kebijakan

Anda dapat menggunakan alias penyimpanan kebijakan untuk mereferensikan penyimpanan kebijakan dalam operasi apa pun yang menerima `policyStoreId` bidang sebagai input. Jika Anda melakukannya, Izin Terverifikasi Amazon memberikan otorisasi

`verifiedpermissions:GetPolicyStoreAlias` terhadap alias penyimpanan kebijakan dan operasi yang diminta terhadap penyimpanan kebijakan terkait.

Misalnya, jika `IsAuthorized` operasi dilakukan dengan menggunakan alias toko kebijakan, prinsipal membutuhkan keduanya:

- `verifiedpermissions:GetPolicyStoreAlias` izin untuk alias toko kebijakan
- `verifiedpermissions:IsAuthorized` izin untuk toko kebijakan terkait

Contoh kebijakan berikut memberikan izin untuk memanggil `IsAuthorized` menggunakan alias penyimpanan kebijakan tertentu.

```
{
  "Sid": "IAMPolicyForAliasUsage",
  "Effect": "Allow",
  "Action": "verifiedpermissions:GetPolicyStoreAlias",
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/example-policy-store"
},
{
  "Sid": "IAMPolicyForPolicyStoreOperation",
  "Effect": "Allow",
  "Action": "verifiedpermissions:IsAuthorized",
  "Resource": "arn:aws:verifiedpermissions::123456789012:policy-store/PSEXAMPLEabcdefg111111"
}
```

Untuk membatasi alias penyimpanan kebijakan mana yang dapat digunakan oleh prinsipal, batasi `verifiedpermissions:GetPolicyStoreAlias` izinnya. Misalnya, kebijakan berikut memungkinkan prinsipal untuk menggunakan alias toko kebijakan apa pun kecuali yang dimulai dengan `Restricted`.

```
{
  "Sid": "IAMPolicyForAliasAllow",
  "Effect": "Allow",
  "Action": "verifiedpermissions:GetPolicyStoreAlias",
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/*"
},
{
  "Sid": "IAMPolicyForAliasDeny",
  "Effect": "Deny",
  "Action": "verifiedpermissions:GetPolicyStoreAlias",
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/Restricted*"
}
```

```
"Action": "verifiedpermissions:GetPolicyStoreAlias",  
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/  
Restricted*"  
}
```

Skema toko kebijakan Izin Terverifikasi Amazon

[Skema](#) adalah deklarasi struktur jenis entitas yang didukung oleh aplikasi Anda, dan tindakan yang mungkin diberikan aplikasi Anda dalam permintaan otorisasi. Untuk melihat perbedaan antara cara Izin Terverifikasi dan Cedar menangani skema, lihat [Dukungan skema](#)

Untuk informasi lebih lanjut, lihat [Format skema Cedar di Panduan Referensi](#) bahasa kebijakan Cedar.

Note

Penggunaan skema dalam Izin Terverifikasi adalah opsional, tetapi sangat direkomendasikan untuk perangkat lunak produksi. Saat Anda membuat kebijakan baru, Izin Terverifikasi dapat menggunakan skema untuk memvalidasi entitas dan atribut yang direferensikan dalam cakupan dan kondisi untuk menghindari kesalahan ketik dan kesalahan dalam kebijakan yang dapat menyebabkan perilaku sistem yang membingungkan. Jika Anda mengaktifkan [validasi kebijakan](#), maka semua kebijakan baru harus sesuai dengan skema.

Konsol Manajemen AWS

Untuk membuat skema

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Skema.
3. Pilih Buat skema.

AWS CLI

Untuk mengirimkan skema baru, atau menimpa skema yang ada dengan menggunakan AWS CLI

Anda dapat membuat penyimpanan kebijakan dengan menjalankan AWS CLI perintah yang mirip dengan contoh berikut.

Pertimbangkan skema yang berisi konten Cedar berikut:

```
{
  "MySampleNamespace": {
    "actions": {
```

```

    "remoteAccess": {
      "appliesTo": {
        "principalTypes": [ "Employee" ]
      }
    },
    "entityTypes": {
      "Employee": {
        "shape": {
          "type": "Record",
          "attributes": {
            "jobLevel": {"type": "Long"},
            "name": {"type": "String"}
          }
        }
      }
    }
  }
}

```

Anda harus terlebih dahulu melarikan diri dari JSON ke dalam string baris tunggal, dan mengawalinya dengan deklarasi tipe datanya. cedarJson Contoh berikut menggunakan isi schema.json file berikut yang berisi versi escaped dari skema JSON.

Note

Contoh di sini adalah baris dibungkus untuk keterbacaan. Anda harus memiliki seluruh file pada satu baris agar perintah menerimanya.

```

{"cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {\"appliesTo\": {\"principalTypes\": [\"Employee\"]}}},\"entityTypes\": {\"Employee\": {\"shape\": {\"attributes\": {\"jobLevel\": {\"type\": \"Long\"},\"name\": {\"type\": \"String\"}}},\"type\": \"Record\"}}}}"}

```

```

$ aws verifiedpermissions put-schema \
  --definition file://schema.json \
  --policy-store PSEXAMPLEabcdefg111111
{

```

```
"policyStoreId": "PSEXAMPLEEabcdefg111111",
"namespaces": [
  "MySampleNamespace"
],
"createdDate": "2023-07-17T21:07:43.659196+00:00",
"lastUpdatedDate": "2023-08-16T17:03:53.081839+00:00"
}
```

AWS SDKs

Anda dapat membuat toko kebijakan menggunakan PutSchema API. Untuk informasi selengkapnya, lihat [PutSchema](#) di Panduan Referensi API Izin Terverifikasi Amazon.

Mengedit skema toko kebijakan

Saat Anda memilih Skema di konsol Izin Terverifikasi Amazon, jenis dan Tindakan Entitas yang membentuk skema Anda akan ditampilkan. Anda dapat melihat edit skema Anda dalam mode Visual atau mode JSON. Mode visual memungkinkan Anda memperbarui skema dengan menambahkan jenis dan tindakan baru menggunakan berbagai wizard. Menggunakan mode JSON, Anda dapat mulai memperbarui kode JSON skema langsung di editor JSON.

Visual Mode

Editor skema visual dimulai dengan serangkaian diagram yang menggambarkan hubungan antara entitas dalam skema Anda. Pilih Expand untuk memaksimalkan tampilan diagram Anda. Ada dua diagram yang tersedia:

- Diagram tindakan — Tampilan diagram Tindakan mencantumkan jenis Prinsipal yang telah Anda konfigurasi di toko kebijakan, Tindakan yang memenuhi syarat untuk dilakukan, dan Sumber Daya tempat mereka memenuhi syarat untuk melakukan tindakan. Garis antar entitas menunjukkan kemampuan Anda untuk membuat kebijakan yang memungkinkan prinsipal untuk mengambil tindakan pada sumber daya. Jika diagram tindakan Anda tidak menunjukkan hubungan antara dua entitas, Anda harus membuat hubungan di antara mereka sebelum Anda dapat mengizinkan atau menolaknya dalam kebijakan. Pilih entitas untuk melihat ikhtisar properti dan telusuri untuk melihat detail selengkapnya. Pilih Filter menurut [action | resource type | principal type] ini untuk melihat entitas dalam tampilan hanya dengan koneksinya sendiri.
- Diagram tipe entitas — Diagram tipe entitas berfokus pada hubungan antara prinsipal dan sumber daya. Saat Anda ingin memahami hubungan induk bersarang yang kompleks dalam

skema Anda, tinjau diagram ini. Arahkan kursor ke entitas untuk menelusuri hubungan induk yang dimilikinya.

Di bawah diagram terdapat tampilan daftar tipe Entitas dan Tindakan dalam skema Anda. Tampilan daftar berguna saat Anda ingin segera melihat detail tindakan atau jenis entitas tertentu. Pilih entitas apa pun untuk melihat detail.

Untuk mengedit skema Izin Terverifikasi dalam mode Visual

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Skema.
3. Pilih mode Visual. Tinjau diagram hubungan-entitas dan rencanakan perubahan yang ingin Anda buat pada skema Anda. Anda dapat secara opsional Memfilter oleh satu entitas untuk memeriksa koneksi individualnya ke entitas lain.
4. Pilih Edit schema (Edit skema).
5. Di bagian Detail, ketik Namespace untuk skema Anda.
6. Di bagian Entity types, pilih Add new entity type.
7. Ketik nama entitas.
8. (Opsional) Pilih Tambahkan induk untuk menambahkan entitas induk yang menjadi anggota entitas baru. Untuk menghapus induk yang telah ditambahkan ke entitas, pilih Hapus di samping nama induk.
9. Pilih Tambahkan atribut untuk menambahkan atribut ke entitas. Ketik nama Atribut dan pilih jenis Atribut untuk setiap atribut entitas. Izin Terverifikasi menggunakan nilai atribut yang ditentukan saat memverifikasi kebijakan terhadap skema. Pilih apakah setiap atribut Diperlukan. Untuk menghapus atribut yang telah ditambahkan ke entitas, pilih Hapus di sebelah atribut.
10. Pilih Tambahkan jenis entitas untuk menambahkan entitas ke skema.
11. Di bagian Tindakan, pilih Tambahkan tindakan baru.
12. Ketik nama tindakan.
13. (Opsional) Pilih Tambahkan sumber daya untuk menambahkan jenis sumber daya yang menerapkan tindakan tersebut. Untuk menghapus jenis sumber daya yang telah ditambahkan ke tindakan, pilih Hapus di samping nama jenis sumber daya.

14. (Opsional) Pilih Tambahkan prinsipal untuk menambahkan tipe utama yang berlaku untuk tindakan tersebut. Untuk menghapus tipe utama yang telah ditambahkan ke tindakan, pilih Hapus di samping nama tipe utama.
15. Pilih Tambahkan atribut untuk menambahkan atribut yang dapat ditambahkan ke konteks tindakan dalam permintaan otorisasi Anda. Masukkan nama Atribut dan pilih jenis Atribut untuk setiap atribut. Izin Terverifikasi menggunakan nilai atribut yang ditentukan saat memverifikasi kebijakan terhadap skema. Pilih apakah setiap atribut Diperlukan. Untuk menghapus atribut yang telah ditambahkan ke tindakan, pilih Hapus di sebelah atribut.
16. Pilih Tambahkan tindakan.
17. Setelah semua jenis dan tindakan entitas ditambahkan ke skema, pilih Simpan perubahan.

JSON mode

Saat melakukan pembaruan, Anda akan melihat editor JSON memvalidasi kode Anda terhadap sintaks JSON dan akan mengidentifikasi kesalahan dan peringatan saat Anda mengedit, sehingga memudahkan Anda menemukan masalah dengan cepat. Selain itu, Anda tidak perlu khawatir tentang pemformatan JSON, cukup pilih Format JSON setelah Anda membuat pembaruan dan format akan diperbarui agar sesuai dengan pemformatan JSON yang diharapkan.

Untuk mengedit skema Izin Terverifikasi dalam mode JSON

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Skema.
3. Pilih mode JSON dan kemudian pilih Edit skema.
4. Masukkan konten skema JSON Anda di bidang Isi. Anda tidak dapat menyimpan pembaruan ke skema Anda sampai Anda menyelesaikan semua kesalahan sintaks. Anda dapat memilih Format JSON untuk memformat sintaks JSON skema Anda dengan spasi dan lekukan yang disarankan.
5. Pilih Simpan perubahan.

Mengaktifkan mode validasi kebijakan Izin Terverifikasi Amazon

Anda dapat menyetel mode validasi kebijakan di Izin Terverifikasi untuk mengontrol apakah perubahan kebijakan divalidasi terhadap [skema](#) di penyimpanan kebijakan Anda.

Important

Ketika Anda mengaktifkan validasi kebijakan, semua upaya untuk membuat atau memperbarui kebijakan atau templat kebijakan divalidasi terhadap skema di penyimpanan kebijakan. Izin Terverifikasi menolak upaya permintaan jika validasi gagal. Untuk alasan ini, kami sarankan untuk menonaktifkan validasi saat Anda mengembangkan aplikasi dan menyalakannya untuk pengujian dan membiarkannya aktif saat aplikasi Anda dalam produksi.

Konsol Manajemen AWS

Untuk menyetel mode validasi kebijakan untuk penyimpanan kebijakan

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Pilih Pengaturan.
3. Di bagian Mode validasi kebijakan, pilih Ubah.
4. Lakukan salah satu hal berikut ini:
 - Untuk mengaktifkan validasi kebijakan dan menegaskan bahwa semua perubahan kebijakan harus divalidasi terhadap skema Anda, pilih tombol radio Ketat (disarankan).
 - Untuk menonaktifkan validasi kebijakan untuk perubahan kebijakan, pilih tombol Nonaktifkan radio. Ketik `confirm` untuk mengonfirmasi bahwa pembaruan kebijakan tidak akan lagi divalidasi terhadap skema Anda.
5. Pilih Simpan perubahan.

AWS CLI

Untuk menyetel mode validasi untuk penyimpanan kebijakan

Anda dapat mengubah mode validasi untuk penyimpanan kebijakan menggunakan [UpdatePolicyStore](#) operasi dan menentukan nilai yang berbeda untuk parameter [ValidationSettings](#)

```
$ aws verifiedpermissions update-policy-store \
  --validation-settings "mode=OFF",
  --policy-store-id PSEXAMPLEabcdefgh111111
{
  "createdDate": "2023-05-17T18:36:10.134448+00:00",
  "lastUpdatedDate": "2023-05-17T18:36:10.134448+00:00",
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "validationSettings": {
    "Mode": "OFF"
  }
}
```

Untuk informasi selengkapnya, lihat [Validasi kebijakan](#) dalam Panduan Referensi bahasa kebijakan Cedar.

Kebijakan Izin Terverifikasi Amazon

Kebijakan adalah pernyataan yang mengizinkan atau melarang kepala sekolah untuk mengambil satu atau lebih tindakan pada sumber daya. Setiap kebijakan dievaluasi secara independen dari setiap kebijakan lainnya. Untuk informasi lebih lanjut tentang bagaimana kebijakan Cedar disusun dan dievaluasi, lihat [Validasi kebijakan Cedar terhadap skema](#) dalam Panduan Referensi bahasa kebijakan Cedar.

Anda dapat secara opsional menetapkan nama kebijakan ke kebijakan. Nama kebijakan harus unik untuk semua kebijakan dalam toko kebijakan dan diawali dengan `name/`. Anda dapat menggunakan nama kebijakan sebagai pengganti ID kebijakan dalam operasi bidang kontrol yang menerima `policyId` parameter. Contoh berikut menggunakan nama kebijakan untuk mengambil kebijakan dengan `getPolicy`.

```
$ aws verifiedpermissions get-policy \  
  --policy-id name/example-policy \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Important

Saat Anda menulis kebijakan Cedar yang mereferensikan prinsip, sumber daya, dan tindakan, Anda dapat menentukan pengidentifikasi unik yang digunakan untuk masing-masing elemen tersebut. Kami sangat menyarankan agar Anda mengikuti praktik terbaik ini:

- Gunakan pengidentifikasi unik universal (UUIDs) untuk semua pengidentifikasi utama dan sumber daya.

Misalnya, jika pengguna `jane` meninggalkan perusahaan, dan Anda kemudian membiarkan orang lain menggunakan nama tersebut `jane`, maka pengguna baru itu secara otomatis mendapatkan akses ke semua yang diberikan oleh kebijakan yang masih mengacu `User::"jane"`. Cedar tidak dapat membedakan antara pengguna baru dan yang lama. Ini berlaku untuk pengidentifikasi utama dan sumber daya. Selalu gunakan pengidentifikasi yang dijamin unik dan tidak pernah digunakan kembali untuk memastikan bahwa Anda tidak secara tidak sengaja memberikan akses karena adanya pengenalan lama dalam kebijakan.

Jika Anda menggunakan UUID untuk entitas, kami sarankan Anda mengikutinya dengan penentu komentar//dan nama 'ramah' entitas Anda. Ini membantu membuat kebijakan

Anda lebih mudah dipahami. Misalnya: `principal == Peran: :\"a1b2c3d4-e5f6-a1b2-c3d4-\",// administrator EXAMPLE11111`

- Jangan sertakan informasi identitas pribadi, rahasia, atau sensitif sebagai bagian dari pengenal unik untuk kepala sekolah atau sumber daya Anda. Pengidentifikasi ini disertakan dalam entri log yang dibagikan di AWS CloudTrail jalur.

Topik

- [Membuat kebijakan statis Izin Terverifikasi Amazon](#)
- [Mengedit kebijakan statis Izin Terverifikasi Amazon](#)
- [Menambahkan konteks](#)
- [Menggunakan bangku uji Izin Terverifikasi Amazon](#)
- [Kebijakan contoh Izin Terverifikasi Amazon](#)

Membuat kebijakan statis Izin Terverifikasi Amazon

Anda dapat membuat kebijakan statis bagi prinsipal untuk mengizinkan atau melarang mereka melakukan tindakan tertentu pada sumber daya tertentu untuk aplikasi Anda. Kebijakan statis memiliki nilai khusus yang disertakan untuk `principal resource` dan dan siap digunakan dalam keputusan otorisasi.

Konsol Manajemen AWS

Untuk membuat kebijakan statis

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Pada panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih Buat kebijakan dan kemudian pilih Buat kebijakan statis.

Note

Jika Anda memiliki pernyataan kebijakan yang ingin Anda gunakan, lewati ke Langkah 8 dan tempelkan kebijakan tersebut ke bagian Kebijakan di halaman berikutnya.

4. Di bagian Efek kebijakan, pilih apakah kebijakan akan mengizinkan atau melarang ketika permintaan cocok dengan kebijakan. Jika Anda memilih Izin, kebijakan memungkinkan kepala sekolah untuk melakukan tindakan pada sumber daya. Sebaliknya, jika Anda memilih Forbid, kebijakan tidak mengizinkan prinsipal untuk melakukan tindakan pada sumber daya.
5. Di bidang lingkup Prinsipal, pilih ruang lingkup prinsipal yang akan diterapkan kebijakan tersebut.
 - Pilih Kepala Sekolah Khusus untuk menerapkan kebijakan ke kepala sekolah tertentu. Tentukan jenis entitas dan pengenal untuk prinsipal yang akan diizinkan atau dilarang untuk mengambil tindakan yang ditentukan dalam kebijakan.
 - Pilih Kelompok kepala sekolah untuk menerapkan kebijakan ke sekelompok kepala sekolah. Ketik nama grup utama di bidang Group of principals.
 - Pilih Semua kepala sekolah untuk menerapkan kebijakan ini ke semua kepala sekolah di toko polis Anda.
6. Di bidang Cakupan sumber daya, pilih ruang lingkup sumber daya yang akan diterapkan kebijakan tersebut.
 - Pilih Sumber daya khusus untuk menerapkan kebijakan ke sumber daya tertentu. Tentukan jenis entitas dan pengenal sumber daya yang harus diterapkan kebijakan tersebut.
 - Pilih Kelompok sumber daya untuk menerapkan kebijakan ke sekelompok sumber daya. Ketik nama grup sumber daya di bidang Kelompok sumber daya.
 - Pilih Semua sumber daya untuk menerapkan kebijakan ke semua sumber daya di toko kebijakan Anda.
7. Di bagian Lingkup tindakan, pilih ruang lingkup sumber daya yang akan diterapkan kebijakan tersebut.
 - Pilih Kumpulan tindakan tertentu untuk menerapkan kebijakan ke serangkaian tindakan. Pilih kotak centang di samping tindakan untuk menerapkan kebijakan.
 - Pilih Semua tindakan untuk menerapkan kebijakan ke semua tindakan di toko kebijakan Anda.
8. Pilih Berikutnya.
9. Di bagian Kebijakan, tinjau kebijakan Cedar Anda. Anda dapat memilih Format untuk memformat sintaks kebijakan Anda dengan spasi dan lekukan yang disarankan. Untuk informasi lebih lanjut, lihat [Konstruksi kebijakan dasar di Cedar](#) dalam Panduan Referensi bahasa kebijakan Cedar.

10. Di bagian Detail, ketikkan deskripsi opsional kebijakan.
11. Pilih Buat kebijakan.

AWS CLI

Untuk membuat kebijakan statis

Anda dapat membuat kebijakan statis dengan menggunakan [CreatePolicy](#) operasi. Contoh berikut membuat kebijakan statis sederhana.

```
$ aws verifiedpermissions create-policy \
  --definition "{ \"static\": { \"Description\": \"MyTestPolicy\", \"Statement\": \
  \"permit(principal,action,resource) when {principal.owner == resource.owner};\"}}\" \
  \
  --policy-store-id PSEXAMPLEabcdefg111111
{
  \"Arn\": \"arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEabcdefg111111/
  SPEXAMPLEabcdefg111111\",
  \"createdDate\": \"2023-05-16T20:33:01.730817+00:00\",
  \"lastUpdatedDate\": \"2023-05-16T20:33:01.730817+00:00\",
  \"policyId\": \"SPEXAMPLEabcdefg111111\",
  \"policyStoreId\": \"PSEXAMPLEabcdefg111111\",
  \"policyType\": \"STATIC\"
}
```

Untuk membuat kebijakan dengan nama kebijakan

Anda dapat menentukan nama kebijakan secara opsional saat membuat kebijakan. Nama harus unik untuk semua kebijakan dalam toko kebijakan dan diawali dengan `name/`. Anda dapat menggunakan nama sebagai pengganti ID kebijakan.

```
$ aws verifiedpermissions create-policy \
  --definition "{ \"static\": { \"Statement\": \"permit(principal, action,
  resource in Album::\\\\"public_folder\\\\");\"}}\" \
  --policy-store-id PSEXAMPLEabcdefg111111 \
  --name name/example-policy
{
  \"createdDate\": \"2023-06-12T20:33:37.382907+00:00\",
  \"lastUpdatedDate\": \"2023-06-12T20:33:37.382907+00:00\",
  \"policyId\": \"SPEXAMPLEabcdefg111111\",
  \"policyStoreId\": \"PSEXAMPLEabcdefg111111\",
}
```

```
"policyType": "STATIC",
"resource": {
  "entityId": "public_folder",
  "entityType": "Album"
}
}
```

Note

Jika Anda menentukan nama yang sudah dikaitkan dengan kebijakan lain di toko kebijakan, Anda akan menerima `ConflictException` kesalahan.

Mengedit kebijakan statis Izin Terverifikasi Amazon

Anda dapat mengedit kebijakan statis yang ada di toko kebijakan Anda. Anda hanya dapat langsung memperbarui kebijakan statis. Untuk mengubah kebijakan terkait templat, Anda harus memperbarui templat kebijakan. Untuk informasi selengkapnya, lihat [Mengedit templat kebijakan Izin Terverifikasi Amazon](#).

Anda dapat mengubah elemen berikut dari kebijakan statis:

- yang `action` direferensikan oleh kebijakan.
- Klausul kondisi, seperti `when` dan `unless`.

Anda tidak dapat mengubah elemen kebijakan statis berikut. Untuk mengubah salah satu elemen ini, Anda harus menghapus dan membuat ulang kebijakan.

- Kebijakan dari kebijakan statis hingga kebijakan terkait templat.
- Pengaruh kebijakan statis dari `permit` atau `forbid`.
- Yang `principal` direferensikan oleh kebijakan statis.
- Yang `resource` direferensikan oleh kebijakan statis.

Konsol Manajemen AWS

Untuk mengedit kebijakan statis

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.

2. Pada panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih tombol radio di samping kebijakan statis yang akan diedit, lalu pilih Edit.
4. Di bagian Badan kebijakan, perbarui klausul `action` atau kondisi kebijakan statis Anda. Anda tidak dapat memperbarui efek kebijakan `principal`, atau `resource` kebijakan.
5. Pilih Perbarui kebijakan.

Note

Jika [validasi kebijakan](#) diaktifkan di penyimpanan kebijakan, memperbarui kebijakan statis akan menyebabkan Izin Terverifikasi memvalidasi kebijakan terhadap skema di penyimpanan kebijakan. Jika kebijakan statis yang diperbarui tidak lulus validasi, operasi gagal dan pembaruan tidak disimpan.

AWS CLI

Untuk mengedit kebijakan statis

Anda dapat mengedit kebijakan statis dengan menggunakan [UpdatePolicy](#) operasi. Contoh berikut mengedit kebijakan statis sederhana.

Contoh menggunakan file `definition.txt` untuk memuat definisi kebijakan.

```
{
  "static": {
    "description": "Grant everyone of janeFriends UserGroup access to the
vacationFolder Album",
    "statement": "permit(principal in UserGroup::\\"janeFriends\\", action,
resource in Album::\\"vacationFolder\\" );"
  }
}
```

Perintah berikut referensi file itu.

```
$ aws verifiedpermissions create-policy \
  --definition file://definition.txt \
  --policy-store-id PSEXAMPLEabcdefg111111
{
```

```

"createdDate": "2023-06-12T20:33:37.382907+00:00",
"lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
"policyId": "SPEXAMPLEabcdefg111111",
"policyStoreId": "PSEXAMPLEabcdefg111111",
"policyType": "STATIC",
"principal": {
  "entityId": "janeFriends",
  "entityType": "UserGroup"
},
"resource": {
  "entityId": "vacationFolder",
  "entityType": "Album"
}
}

```

Untuk memperbarui nama kebijakan

Anda dapat menyetel atau memperbarui nama kebijakan saat memperbarui kebijakan. Nama harus unik untuk semua kebijakan dalam toko kebijakan dan diawali dengan `name/`. Jika Anda tidak menyertakan bidang nama dalam permintaan pembaruan, nama yang ada tidak berubah. Untuk menghapus nama, atur ke string kosong.

```

$ aws verifiedpermissions update-policy \
  --policy-id SPEXAMPLEabcdefg111111 \
  --policy-store-id PSEXAMPLEabcdefg111111 \
  --definition file://definition.txt \
  --name name/example-policy
{
  "createdDate": "2023-06-12T20:33:37.382907+00:00",
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
  "policyId": "SPEXAMPLEabcdefg111111",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyType": "STATIC",
  "principal": {
    "entityId": "janeFriends",
    "entityType": "UserGroup"
  },
  "resource": {
    "entityId": "vacationFolder",
    "entityType": "Album"
  }
}

```

Menambahkan konteks

Konteks adalah informasi yang relevan dengan keputusan kebijakan, tetapi bukan bagian dari identitas kepala sekolah, tindakan, atau sumber daya Anda. Klaim token akses adalah konteks. Anda mungkin ingin mengizinkan tindakan hanya dari sekumpulan alamat IP sumber, atau hanya jika pengguna Anda telah masuk dengan MFA. Aplikasi Anda memiliki akses ke data sesi kontekstual ini dan harus mengisinya ke permintaan otorisasi. Data konteks dalam permintaan otorisasi Izin Terverifikasi harus diformat JSON dalam elemen `contextMap`

Contoh yang menggambarkan konten ini berasal dari [toko kebijakan sampel](#). Untuk mengikuti, buat penyimpanan kebijakan DigitalPetStoresampel di lingkungan pengujian Anda.

Objek konteks berikut mendeklarasikan salah satu dari setiap tipe data Cedar untuk aplikasi berdasarkan penyimpanan DigitalPetStorekebijakan sampel.

```
"context": {
  "contextMap": {
    "AccountCodes": {
      "set": [
        {
          "long": 111122223333
        },
        {
          "long": 444455556666
        },
        {
          "long": 123456789012
        }
      ]
    },
    "approvedBy": {
      "entityIdentifier": {
        "entityId": "Bob",
        "entityType": "DigitalPetStore::User"
      }
    },
    "MfaAuthorized": {
      "boolean": true
    },
    "NetworkInfo": {
      "record": {
        "IPAddress": {
```

```
    "string": "192.0.2.178"
  },
  "Country": {
    "string": "United States of America"
  },
  "SSL": {
    "boolean": true
  }
},
"RequestedOrderCount": {
  "long": 4
},
"UserAgent": {
  "string": "My UserAgent 1.12"
}
}
```

Tipe data dalam konteks otorisasi

Boolean

Biner `true` atau `false` nilai. Dalam contoh, nilai boolean `true` for `MfaAuthenticated` menunjukkan bahwa pelanggan telah melakukan otentikasi multi-faktor sebelum meminta untuk melihat pesanan mereka.

Set

Kumpulan elemen konteks. Anggota set bisa semua jenis yang sama, seperti dalam contoh ini, atau tipe yang berbeda, termasuk set bersarang. Dalam contoh, pelanggan dikaitkan dengan 3 akun berbeda.

String

Urutan huruf, angka, atau simbol, terlampir dalam " karakter. Dalam contoh, `UserAgent` string mewakili browser yang digunakan pelanggan untuk meminta untuk melihat pesanan mereka.

Panjang

Sebuah bilangan bulat. Dalam contoh, `RequestedOrderCount` menunjukkan bahwa permintaan ini adalah bagian dari batch yang dihasilkan dari pelanggan yang meminta untuk melihat empat pesanan mereka sebelumnya.

Rekam

Kumpulan atribut. Anda harus mendeklarasikan atribut ini dalam konteks permintaan.

Penyimpanan kebijakan dengan skema harus menyertakan entitas ini dan atribut entitas dalam skema. Dalam contoh, `NetworkInfo` catatan berisi informasi tentang IP asal pengguna, geolokasi IP tersebut sebagaimana ditentukan oleh klien, dan enkripsi dalam perjalanan.

EntityIdentifier

Referensi ke entitas dan atribut yang dideklarasikan dalam `entities` elemen permintaan. Dalam contoh, pesanan pengguna disetujui oleh `karyawanBob`.

Untuk menguji konteks contoh ini di `DigitalPetStore` aplikasi contoh, Anda harus memperbarui permintaan `entities`, skema penyimpanan kebijakan, dan kebijakan statis dengan deskripsi Peran Pelanggan - Dapatkan Pesanan.

Memodifikasi DigitalPetStore untuk menerima konteks otorisasi

Awalnya, `DigitalPetStore` bukan toko kebijakan yang sangat kompleks. Itu tidak termasuk kebijakan atau atribut konteks yang telah dikonfigurasi sebelumnya untuk mendukung konteks yang telah kami sajikan. Untuk mengevaluasi contoh permintaan otorisasi dengan informasi konteks ini, lakukan modifikasi berikut pada toko kebijakan Anda dan permintaan otorisasi Anda. Untuk contoh konteks dengan informasi token akses sebagai konteksnya, lihat [Memetakan token Amazon Cognito akses dan Memetakan token akses OIDC](#).

Schema

Terapkan pembaruan berikut ke skema penyimpanan kebijakan Anda untuk mendukung atribut konteks baru. Perbarui `GetOrder` actions sebagai berikut.

```
"GetOrder": {
  "memberOf": [],
  "appliesTo": {
    "resourceTypes": [
      "Order"
    ],
    "context": {
      "type": "Record",
      "attributes": {
        "AccountCodes": {
```

```

    "type": "Set",
    "required": true,
    "element": {
      "type": "Long"
    }
  },
  "approvedBy": {
    "name": "User",
    "required": true,
    "type": "Entity"
  },
  "MfaAuthorized": {
    "type": "Boolean",
    "required": true
  },
  "NetworkInfo": {
    "type": "NetworkInfo",
    "required": true
  },
  "RequestedOrderCount": {
    "type": "Long",
    "required": true
  },
  "UserAgent": {
    "required": true,
    "type": "String"
  }
}
},
"principalTypes": [
  "User"
]
}
}

```

Untuk mereferensikan tipe record data yang dinamai `NetworkInfo` dalam konteks permintaan Anda, buat konstruksi [CommonType](#) dalam skema Anda dengan menambahkan berikut ini ke skema Anda sebelumnya. `actions commonTypeKonstruk` adalah kumpulan atribut bersama yang dapat Anda terapkan ke entitas yang berbeda.

```

"commonTypes": {
  "NetworkInfo": {
    "attributes": {

```

```

    "IPAddress": {
      "type": "String",
      "required": true
    },
    "SSL": {
      "required": true,
      "type": "Boolean"
    },
    "Country": {
      "required": true,
      "type": "String"
    }
  },
  "type": "Record"
}
},

```

Policy

Kebijakan berikut menetapkan kondisi yang harus dipenuhi oleh masing-masing elemen konteks yang disediakan. Ini dibangun di atas kebijakan statis yang ada dengan deskripsi Peran Pelanggan - Dapatkan Pesanan. Kebijakan ini awalnya hanya mensyaratkan bahwa kepala sekolah yang membuat permintaan adalah pemilik sumber daya.

```

permit (
  principal in DigitalPetStore::Role::"Customer",
  action in [DigitalPetStore::Action::"GetOrder"],
  resource
) when {
  principal == resource.owner &&
  context.AccountCodes.contains(111122223333) &&
  context.approvedBy in DigitalPetStore::Role::"Employee" &&
  context.MfaAuthorized == true &&
  context.NetworkInfo.Country like "*United States*" &&
  context.NetworkInfo.IPAddress like "192.0.2.*" &&
  context.NetworkInfo.SSL == true &&
  context.RequestedOrderCount <= 4 &&
  context.UserAgent like "*My UserAgent*"
};

```

Kami sekarang mengharuskan permintaan untuk mengambil pesanan memenuhi kondisi konteks tambahan yang kami tambahkan ke permintaan.

1. Pengguna harus masuk dengan MFA.
2. Browser web pengguna User-Agent harus berisi stringMy UserAgent.
3. Pengguna harus meminta untuk melihat 4 atau lebih sedikit pesanan.
4. Salah satu kode akun pengguna harus111122223333.
5. Alamat IP pengguna harus berasal dari Amerika Serikat, mereka harus pada sesi terenkripsi, dan alamat IP mereka harus dimulai dengan. 192.0.2.
6. Seorang karyawan harus menyetujui pesanan mereka. Dalam entities elemen permintaan otorisasi, kami akan mendeklarasikan pengguna Bob yang memiliki peran. Employee

Request body

Setelah mengonfigurasi penyimpanan kebijakan dengan skema dan kebijakan yang sesuai, Anda dapat menampilkan permintaan otorisasi ini ke operasi API Izin Terverifikasi. [IsAuthorized](#) Perhatikan bahwa entities segmen berisi definisiBob, pengguna dengan peranEmployee.

```
{
  "principal": {
    "entityType": "DigitalPetStore::User",
    "entityId": "Alice"
  },
  "action": {
    "actionType": "DigitalPetStore::Action",
    "actionId": "GetOrder"
  },
  "resource": {
    "entityType": "DigitalPetStore::Order",
    "entityId": "1234"
  },
  "context": {
    "contextMap": {
      "AccountCodes": {
        "set": [
          {"long": 111122223333},
          {"long": 444455556666},
          {"long": 123456789012}
        ]
      }
    },
    "approvedBy": {
      "entityIdentifier": {
        "entityId": "Bob",
```

```
    "entityType": "DigitalPetStore::User"
  }
},
"MfaAuthorized": {
  "boolean": true
},
"NetworkInfo": {
  "record": {
    "Country": {"string": "United States of America"},
    "IPAddress": {"string": "192.0.2.178"},
    "SSL": {"boolean": true}
  }
},
"RequestedOrderCount":{
  "long": 4
},
"UserAgent": {
  "string": "My UserAgent 1.12"
}
},
"entities": {
  "entityList": [
    {
      "identifier": {
        "entityType": "DigitalPetStore::User",
        "entityId": "Alice"
      },
      "attributes": {
        "memberId": {
          "string": "801b87f2-1a5c-40b3-b580-eacad506d4e6"
        }
      },
      "parents": [
        {
          "entityType": "DigitalPetStore::Role",
          "entityId": "Customer"
        }
      ]
    },
    {
      "identifier": {
        "entityType": "DigitalPetStore::User",
        "entityId": "Bob"
      }
    }
  ]
}
```

```
    },
    "attributes": {
      "memberId": {
        "string": "49d9b81e-735d-429c-989d-93bec0bcfd8b"
      }
    },
  },
  "parents": [
    {
      "entityType": "DigitalPetStore::Role",
      "entityId": "Employee"
    }
  ]
},
{
  "identifier": {
    "entityType": "DigitalPetStore::Order",
    "entityId": "1234"
  },
  "attributes": {
    "owner": {
      "entityIdentifier": {
        "entityType": "DigitalPetStore::User",
        "entityId": "Alice"
      }
    }
  },
  "parents": []
}
],
},
"policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

Menggunakan bangku uji Izin Terverifikasi Amazon

Gunakan bangku uji Izin Terverifikasi untuk menguji dan memecahkan masalah kebijakan Izin Terverifikasi dengan menjalankan permintaan [otorisasi](#) terhadapnya. Bangku tes menggunakan parameter yang Anda tentukan untuk menentukan apakah kebijakan Cedar di toko kebijakan Anda akan mengotorisasi permintaan tersebut. Anda dapat beralih antara mode Visual dan mode JSON saat menguji permintaan otorisasi. Untuk informasi lebih lanjut tentang bagaimana kebijakan Cedar

disusun dan dievaluasi, lihat [Konstruksi kebijakan dasar di Cedar dalam Panduan Referensi](#) bahasa kebijakan Cedar.

Note

Saat Anda membuat permintaan otorisasi menggunakan Izin Terverifikasi, Anda dapat memberikan daftar prinsipal dan sumber daya sebagai bagian dari permintaan di bagian Entitas tambahan. Namun, Anda tidak dapat menyertakan detail tentang tindakan tersebut. Mereka harus ditentukan dalam skema atau disimpulkan dari permintaan. Anda tidak dapat menempatkan tindakan di bagian Entitas tambahan.

Untuk ikhtisar visual dan demonstrasi bangku tes, lihat [Izin Terverifikasi Amazon - Pembuatan dan Pengujian Kebijakan \(Seri Primer #3\)](#) di saluran AWS YouTube .

Visual mode

Note

Anda harus memiliki skema yang ditentukan di toko kebijakan Anda untuk menggunakan mode Visual bangku tes.

Untuk menguji kebijakan dalam mode Visual

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Test bench.
3. Pilih mode Visual.
4. Di bagian Principal, pilih Principal mengambil tindakan dari tipe utama dalam skema Anda. Ketik pengenal untuk prinsipal di kotak teks.
5. (Opsional) Pilih Tambahkan induk untuk menambahkan entitas induk untuk prinsipal yang ditentukan. Untuk menghapus induk yang telah ditambahkan ke prinsipal, pilih Hapus di samping nama induk.
6. Tentukan nilai Atribut untuk setiap atribut dari prinsipal yang ditentukan. Bangku tes menggunakan nilai atribut yang ditentukan dalam permintaan otorisasi simulasi.
7. Di bagian Sumber Daya, pilih Sumber Daya yang ditindaklanjuti prinsipal. Ketik pengenal untuk sumber daya di kotak teks.

8. (Opsional) Pilih Tambahkan induk untuk menambahkan entitas induk untuk sumber daya yang ditentukan. Untuk menghapus induk yang telah ditambahkan ke sumber daya, pilih Hapus di samping nama induk.
9. Tentukan nilai Atribut untuk setiap atribut sumber daya yang ditentukan. Bangku tes menggunakan nilai atribut yang ditentukan dalam permintaan otorisasi simulasi.
10. Di bagian Tindakan, pilih Tindakan yang diambil prinsipal dari daftar tindakan yang valid untuk prinsipal dan sumber daya yang ditentukan.
11. Tentukan nilai Atribut untuk setiap atribut dari tindakan yang ditentukan. Bangku tes menggunakan nilai atribut yang ditentukan dalam permintaan otorisasi simulasi.
12. (Opsional) Di bagian Entitas tambahan, pilih Tambahkan entitas untuk menambahkan entitas yang akan dievaluasi untuk keputusan otorisasi.
13. Pilih Entity Identifier dari daftar dropdown dan ketik pengenal entitas.
14. (Opsional) Pilih Tambahkan induk untuk menambahkan entitas induk untuk entitas yang ditentukan. Untuk menghapus induk yang telah ditambahkan ke entitas, pilih Hapus di samping nama induk.
15. Tentukan nilai Atribut untuk setiap atribut dari entitas yang ditentukan. Bangku tes menggunakan nilai atribut yang ditentukan dalam permintaan otorisasi simulasi.
16. Pilih Konfirmasi untuk menambahkan entitas ke bangku tes.
17. Pilih Jalankan permintaan otorisasi untuk mensimulasikan permintaan otorisasi kebijakan Cedar di toko kebijakan Anda. Bangku tes menampilkan keputusan untuk mengizinkan atau menolak permintaan bersama dengan informasi tentang kebijakan yang dipenuhi atau kesalahan yang dihadapi selama evaluasi.

JSON mode

Untuk menguji kebijakan dalam mode JSON

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Test bench.
3. Pilih mode JSON.
4. Di bagian Permintaan rincian, jika Anda memiliki skema yang ditentukan, pilih Principal mengambil tindakan dari tipe utama dalam skema Anda. Ketik pengenal untuk prinsipal di kotak teks.

Jika Anda tidak memiliki skema yang ditentukan, ketikkan prinsipal di kotak teks Principal taking action.

5. Jika Anda memiliki skema yang ditentukan, pilih Sumber Daya dari jenis sumber daya dalam skema Anda. Ketik pengenal untuk sumber daya di kotak teks.

Jika Anda tidak memiliki skema yang ditentukan, ketikkan sumber daya di kotak teks Sumber daya.

6. Jika Anda memiliki skema yang ditentukan, pilih Tindakan dari daftar tindakan yang valid untuk prinsipal dan sumber daya yang ditentukan.

Jika Anda tidak memiliki skema yang ditentukan, ketikkan tindakan di kotak teks Tindakan.

7. Masukkan konteks permintaan untuk disimulasikan di bidang Konteks. Konteks permintaan adalah informasi tambahan yang dapat digunakan untuk keputusan otorisasi.
8. Di bidang Entitas, masukkan hierarki entitas dan atributnya untuk dievaluasi untuk keputusan otorisasi.
9. Pilih Jalankan permintaan otorisasi untuk mensimulasikan permintaan otorisasi kebijakan Cedar di toko kebijakan Anda. Bangku tes menampilkan keputusan untuk mengizinkan atau menolak permintaan bersama dengan informasi tentang kebijakan yang dipenuhi atau kesalahan yang dihadapi selama evaluasi.

Kebijakan contoh Izin Terverifikasi Amazon

Beberapa contoh kebijakan yang disertakan di sini adalah contoh kebijakan Cedar dasar dan beberapa khusus Izin Terverifikasi. Yang dasar terhubung ke Panduan Referensi bahasa kebijakan Cedar dan disertakan di sana. Untuk informasi selengkapnya tentang sintaks kebijakan Cedar, lihat [Konstruksi kebijakan dasar di Cedar dalam Panduan Referensi](#) bahasa kebijakan Cedar.

Contoh kebijakan

- [Memungkinkan akses ke entitas individu](#)
- [Memungkinkan akses ke grup entitas](#)
- [Memungkinkan akses untuk entitas apa pun](#)
- [Memungkinkan akses untuk atribut entitas \(ABAC\)](#)
- [Menolak akses](#)

- [Menggunakan notasi braket untuk referensi atribut token](#)
- [Menggunakan notasi titik untuk referensi atribut](#)
- [Mencerminkan atribut token Amazon Cognito ID](#)
- [Mencerminkan atribut token ID OIDC](#)
- [Mencerminkan atribut token Amazon Cognito akses](#)
- [Mencerminkan atribut token akses OIDC](#)

Menggunakan notasi braket untuk referensi atribut token

Contoh berikut ini menunjukkan bagaimana Anda dapat membuat kebijakan yang menggunakan notasi braket untuk referensi atribut token.

Untuk informasi selengkapnya tentang penggunaan atribut token dalam kebijakan di Izin Terverifikasi, lihat [Memetakan Amazon Cognito token ke skema dan Memetakan token OIDC ke skema](#).

```
permit (  
    principal in MyCorp::UserGroup::"us-west-2_EXAMPLE|MyUserGroup",  
    action,  
    resource  
) when {  
    principal["cognito:username"] == "alice" &&  
    principal["custom:employmentStoreCode"] == "petstore-dallas" &&  
    principal has email && principal.email == "alice@example.com" &&  
    context["ip-address"] like "192.0.2.*"  
};
```

Menggunakan notasi titik untuk referensi atribut

Contoh berikut ini menunjukkan bagaimana Anda dapat membuat kebijakan yang menggunakan notasi titik untuk referensi atribut.

Untuk informasi selengkapnya tentang penggunaan atribut token dalam kebijakan di Izin Terverifikasi, lihat [Memetakan Amazon Cognito token ke skema dan Memetakan token OIDC ke skema](#).

```
permit(principal, action, resource)  
when {  
    principal.cognito.username == "alice" &&
```

```
principal.custom.employmentStoreCode == "petstore-dallas" &&
principal.tenant == "x11app-tenant-1" &&
principal has email && principal.email == "alice@example.com"
};
```

Mencerminkan atribut token Amazon Cognito ID

Contoh berikut ini menunjukkan bagaimana Anda dapat membuat atribut token ID referensi kebijakan dari Amazon Cognito.

Untuk informasi selengkapnya tentang penggunaan atribut token dalam kebijakan di Izin Terverifikasi, lihat [Memetakan Amazon Cognito token ke skema dan Memetakan token OIDC ke skema](#).

```
permit (
  principal in MyCorp::UserGroup::"us-west-2_EXAMPLE|MyUserGroup",
  action,
  resource
) when {
  principal["cognito:username"] == "alice" &&
  principal["custom:employmentStoreCode"] == "petstore-dallas" &&
  principal.tenant == "x11app-tenant-1" &&
  principal has email && principal.email == "alice@example.com"
};
```

Mencerminkan atribut token ID OIDC

Contoh berikut ini menunjukkan bagaimana Anda dapat membuat atribut token ID referensi kebijakan dari penyedia OIDC.

Untuk informasi selengkapnya tentang penggunaan atribut token dalam kebijakan di Izin Terverifikasi, lihat [Memetakan Amazon Cognito token ke skema dan Memetakan token OIDC ke skema](#).

```
permit (
  principal in MyCorp::UserGroup::"MyOIDCProvider|MyUserGroup",
  action,
  resource
) when {
  principal.email_verified == true && principal.email == "alice@example.com" &&
  principal.phone_number_verified == true && principal.phone_number like "+1206*"
};
```

Mencerminkan atribut token Amazon Cognito akses

Contoh berikut ini menunjukkan bagaimana Anda dapat membuat referensi kebijakan atribut token akses dari Amazon Cognito.

Untuk informasi selengkapnya tentang penggunaan atribut token dalam kebijakan di Izin Terverifikasi, lihat [Memetakan Amazon Cognito token ke skema dan Memetakan token OIDC ke skema](#).

```
permit(principal, action in [MyApplication::Action::"Read",
  MyApplication::Action::"GetStoreInventory"], resource)
when {
  context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
  context.token.scope.contains("MyAPI/mydata.write")
};
```

Mencerminkan atribut token akses OIDC

Contoh berikut ini menunjukkan bagaimana Anda dapat membuat referensi kebijakan atribut token akses dari penyedia OIDC.

Untuk informasi selengkapnya tentang penggunaan atribut token dalam kebijakan di Izin Terverifikasi, lihat [Memetakan Amazon Cognito token ke skema dan Memetakan token OIDC ke skema](#).

```
permit(
  principal,
  action in [MyApplication::Action::"Read",
  MyApplication::Action::"GetStoreInventory"],
  resource
)
when {
  context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
  context.token.scope.contains("MyAPI-read")
};
```

Templat kebijakan Izin Terverifikasi Amazon dan kebijakan terkait templat

Di Izin Terverifikasi, templat kebijakan adalah kebijakan dengan placeholder untuk `principal`, `resource`, atau keduanya. Template kebijakan saja tidak dapat digunakan untuk menangani permintaan otorisasi. Untuk menangani permintaan otorisasi, kebijakan terkait templat harus dibuat berdasarkan templat kebijakan. Templat kebijakan memungkinkan kebijakan didefinisikan sekali dan kemudian digunakan dengan beberapa prinsip dan sumber daya. Pembaruan pada templat kebijakan tercermin di semua kebijakan yang menggunakan templat. Untuk informasi selengkapnya, lihat [Templat kebijakan Cedar](#) di Panduan Referensi bahasa kebijakan Cedar.

Anda dapat menetapkan nama templat kebijakan secara opsional ke templat kebijakan. Nama templat kebijakan harus unik di dalam toko kebijakan dan diawali dengan `name/`. Anda dapat menggunakan nama templat kebijakan sebagai pengganti ID templat kebijakan dalam operasi bidang kontrol yang menerima `policyTemplateId` parameter. Hanya `GetPolicyTemplate` dan `ListPolicyTemplates` kembalikan nama di output. Contoh berikut menggunakan nama template kebijakan untuk mengambil template kebijakan dengan `GetPolicyTemplate`.

```
$ aws verifiedpermissions get-policy-template \  
  --policy-template-id name/example-policy-template \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Misalnya, templat kebijakan berikut menyediakan `Read`, `Edit`, dan `Comment` izin untuk prinsipal dan sumber daya yang menggunakan templat kebijakan.

```
permit(  
  principal == ?principal,  
  action in [Action::"Read", Action::"Edit", Action::"Comment"],  
  resource == ?resource  
);
```

Jika Anda membuat kebijakan yang diberi nama `Editor` berdasarkan templat ini, ketika prinsipal ditetapkan sebagai editor untuk sumber daya tertentu, aplikasi Anda akan membuat kebijakan yang memberikan izin bagi prinsipal untuk membaca, mengedit, dan mengomentari sumber daya tersebut.

Tidak seperti kebijakan statis, kebijakan terkait template bersifat dinamis. Ambil contoh sebelumnya, jika Anda menghapus `Comment` tindakan dari templat kebijakan, kebijakan apa pun yang ditautkan

ke, atau berdasarkan, templat tersebut akan diperbarui sesuai dan prinsip yang ditentukan dalam kebijakan tidak akan lagi dapat mengomentari sumber daya terkait.

Untuk contoh kebijakan terkait templat lainnya, lihat. [Izin Terverifikasi Amazon contoh kebijakan terkait templat](#)

Membuat templat kebijakan Izin Terverifikasi Amazon

Anda dapat membuat templat kebijakan di Izin Terverifikasi menggunakan Konsol Manajemen AWS, templat AWS CLI, atau templat. AWS SDKs Templat kebijakan memungkinkan kebijakan didefinisikan sekali dan kemudian digunakan dengan beberapa prinsip dan sumber daya. Setelah membuat templat kebijakan, Anda dapat membuat kebijakan terkait templat untuk menggunakan templat kebijakan dengan prinsip dan sumber daya tertentu. Untuk informasi selengkapnya, lihat [Membuat kebijakan terkait templat Izin Terverifikasi Amazon](#).

Konsol Manajemen AWS

Untuk membuat templat kebijakan

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Templat kebijakan.
3. Pilih Buat templat kebijakan.
4. Di bagian Detail, ketikkan deskripsi templat Kebijakan.
5. Di bagian badan Templat kebijakan, gunakan placeholder `?principal` dan `?resource` untuk mengizinkan kebijakan yang dibuat berdasarkan templat ini untuk menyesuaikan izin yang mereka berikan. Anda dapat memilih Format untuk memformat sintaks templat kebijakan Anda dengan spasi dan lekukan yang disarankan.
6. Pilih Buat templat kebijakan.

AWS CLI

Untuk membuat templat kebijakan

Anda dapat membuat templat kebijakan dengan menggunakan [CreatePolicyTemplate](#) operasi. Contoh berikut membuat template kebijakan dengan placeholder untuk prinsipal.

File `template1.txt` berisi yang berikut ini.

```
"VacationAccess"
permit(
  principal in ?principal,
  action == Action::"view",
  resource == Photo::"VacationPhoto94.jpg"
);
```

```
$ aws verifiedpermissions create-policy-template \
  --description "Template for vacation picture access"
  --statement file://template1.txt
  --policy-store-id PSEXAMPLEEabcdefg111111
{
  "createdDate": "2023-05-18T21:17:47.284268+00:00",
  "lastUpdatedDate": "2023-05-18T21:17:47.284268+00:00",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEEabcdefg111111"
}
```

Untuk membuat templat kebijakan dengan nama templat kebijakan

Anda dapat menentukan nama templat kebijakan secara opsional saat membuat templat kebijakan. Nama harus unik untuk semua templat kebijakan dalam toko kebijakan dan diawali dengan `name/`. Anda dapat menggunakan nama sebagai pengganti ID templat kebijakan.

```
$ aws verifiedpermissions create-policy-template \
  --description "Template for vacation picture access" \
  --statement file://template1.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111 \
  --name name/example-policy-template
{
  "createdDate": "2023-06-12T20:47:42.804511+00:00",
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEEabcdefg111111"
}
```

Note

Jika Anda menentukan nama yang sudah dikaitkan dengan templat kebijakan lain di toko kebijakan, Anda akan menerima `ConflictException` kesalahan.

Membuat kebijakan terkait templat Izin Terverifikasi Amazon

Anda dapat membuat kebijakan terkait templat, atau kebijakan yang didasarkan pada templat kebijakan, menggunakan, atau kebijakan. Konsol Manajemen AWS AWS CLI AWS SDKs Kebijakan terkait templat tetap ditautkan ke templat kebijakan mereka. Jika Anda mengubah pernyataan kebijakan dalam templat kebijakan, kebijakan apa pun yang ditautkan ke templat tersebut secara otomatis menggunakan pernyataan baru untuk semua keputusan otorisasi yang dibuat sejak saat itu.

Untuk contoh kebijakan terkait templat, lihat [Izin Terverifikasi Amazon contoh kebijakan terkait templat](#)

Konsol Manajemen AWS

Untuk membuat kebijakan yang ditautkan templat dengan membuat instance template kebijakan

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Pada panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih Buat kebijakan, lalu pilih Buat kebijakan terkait templat.
4. Pilih tombol radio di sebelah templat kebijakan yang akan digunakan, lalu pilih Berikutnya.
5. Ketik Principal dan Resource yang akan digunakan untuk contoh spesifik kebijakan terkait template ini. Nilai yang ditentukan ditampilkan di bidang pratinjau pernyataan Kebijakan.

Note

Nilai Principal dan Resource harus memiliki format yang sama dengan kebijakan statis. Misalnya, untuk menentukan AdminUsers grup untuk prinsipal, ketik `Group: "AdminUsers"`. Jika Anda mengetik `AdminUsers`, kesalahan validasi ditampilkan.

6. Pilih Buat kebijakan terkait templat.

Kebijakan baru yang ditautkan templat ditampilkan di bawah Kebijakan.

AWS CLI

Untuk membuat kebijakan yang ditautkan templat dengan membuat instance template kebijakan

Anda dapat membuat kebijakan terkait templat yang mereferensikan templat kebijakan yang ada dan yang menentukan nilai untuk setiap placeholder yang digunakan oleh templat.

Contoh berikut membuat kebijakan terkait templat yang menggunakan templat dengan pernyataan berikut:

```
permit(  
  principal in ?principal,  
  action == PhotoFlash::Action::"view",  
  resource == PhotoFlash::Photo::"VacationPhoto94.jpg"  
);
```

Ini juga menggunakan `definition.txt` file berikut untuk memberikan nilai untuk `definition` parameter:

```
{  
  "templateLinked": {  
    "policyTemplateId": "PTEXAMPLEabcdefgh111111",  
    "principal": {  
      "entityType": "PhotoFlash::User",  
      "entityId": "alice"  
    }  
  }  
}
```

Output menunjukkan sumber daya, yang didapatnya dari template, dan prinsipal, yang didapatnya dari parameter definisi

```
$ aws verifiedpermissions create-policy \  
  --definition file://definition.txt  
  --policy-store-id PSEXAMPLEabcdefgh111111  
{  
  "createdDate": "2023-05-22T18:57:53.298278+00:00",  
  "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",  
  "policyId": "TPEXAMPLEabcdefgh111111",  
  "policyStoreId": "PSEXAMPLEabcdefgh111111",  
  "policyType": "TEMPLATELINKED",  
  "principal": {  
    "entityId": "alice",  
    "entityType": "PhotoFlash::User"  
  },  
  "resource": {
```

```
    "entityId": "VacationPhoto94.jpg",  
    "entityType": "PhotoFlash::Photo"  
  }  
}
```

Mengedit templat kebijakan Izin Terverifikasi Amazon

Anda dapat mengedit, atau memperbarui, templat kebijakan di Izin Terverifikasi menggunakan Konsol Manajemen AWS, AWS CLI, atau templat. AWS SDKs Mengedit templat kebijakan akan secara otomatis memperbarui kebijakan yang ditautkan ke, atau berdasarkan, templat, jadi berhati-hatilah saat mengedit templat kebijakan dan pastikan Anda tidak secara tidak sengaja memperkenalkan perubahan yang merusak aplikasi Anda.

Anda dapat mengubah elemen template kebijakan berikut:

- Yang `action` direferensikan oleh templat kebijakan
- Klausul kondisi, seperti `and` dan `unless`

Anda tidak dapat mengubah elemen templat kebijakan berikut. Untuk mengubah salah satu elemen ini, Anda harus menghapus dan membuat ulang templat kebijakan.

- Pengaruh template kebijakan dari `permit` atau `forbid`
- Yang `principal` direferensikan oleh templat kebijakan
- Yang `resource` direferensikan oleh templat kebijakan

Konsol Manajemen AWS

Untuk mengedit templat kebijakan Anda

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Templat kebijakan. Konsol menampilkan semua templat kebijakan yang Anda buat di penyimpanan kebijakan saat ini.
3. Pilih tombol radio di samping templat kebijakan untuk menampilkan detail tentang templat kebijakan, seperti saat templat kebijakan dibuat, diperbarui, dan konten templat kebijakan.
4. Pilih Edit untuk mengedit templat kebijakan Anda. Perbarui deskripsi Kebijakan dan badan Kebijakan seperlunya, lalu pilih Perbarui templat kebijakan.

5. Anda dapat menghapus templat kebijakan dengan memilih tombol radio di samping templat kebijakan, lalu memilih Hapus. Pilih OK untuk mengonfirmasi penghapusan templat kebijakan.

AWS CLI

Untuk mengedit templat kebijakan

Anda dapat membuat kebijakan statis dengan menggunakan [UpdatePolicy](#) operasi. Contoh berikut memperbarui templat kebijakan yang ditentukan dengan mengganti badan kebijakannya dengan kebijakan baru yang ditentukan dalam file.

Isi file `template1.txt`:

```
permit(  
    principal in ?principal,  
    action == Action::"view",  
    resource in ?resource)  
when {  
    principal has department && principal.department == "research"  
};
```

```
$ aws verifiedpermissions update-policy-template \  
  --policy-template-id PTEXTAMPLEabcdefg111111 \  
  --description "My updated template description" \  
  --statement file://template1.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111  
{  
  "createdDate": "2023-05-17T18:58:48.795411+00:00",  
  "lastUpdatedDate": "2023-05-17T19:18:48.870209+00:00",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "policyTemplateId": "PTEXTAMPLEabcdefg111111"  
}
```

Untuk memperbarui nama templat kebijakan

Anda dapat menyetel atau memperbarui nama templat kebijakan saat memperbarui templat kebijakan. Nama harus unik untuk semua templat kebijakan dalam toko kebijakan dan diawali dengan `name/`. Jika Anda tidak menyertakan bidang nama dalam permintaan pembaruan, nama yang ada tidak berubah. Untuk menghapus nama, atur ke string kosong.

```
$ aws verifiedpermissions update-policy-template \  
  --policy-template-id PTEXAMPLEabcdefg111111 \  
  --statement file://template1.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111 \  
  --name name/example-policy-template  
{  
  "createdDate": "2023-05-17T18:58:48.795411+00:00",  
  "lastUpdatedDate": "2023-05-17T19:18:48.870209+00:00",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "policyTemplateId": "PTEXAMPLEabcdefg111111"  
}
```

Izin Terverifikasi Amazon contoh kebijakan terkait templat

Saat Anda membuat penyimpanan kebijakan di Izin Terverifikasi menggunakan metode penyimpanan kebijakan sampel, penyimpanan kebijakan Anda dibuat dengan kebijakan, templat kebijakan, dan skema yang telah ditentukan sebelumnya untuk proyek sampel yang Anda pilih. Contoh kebijakan terkait templat Izin Terverifikasi berikut dapat digunakan dengan penyimpanan kebijakan sampel dan kebijakan, templat kebijakan, dan skema masing-masing.

PhotoFlash contoh

Contoh berikut menunjukkan cara membuat kebijakan tertaut templat yang menggunakan templat kebijakan Berikan akses terbatas ke foto bersama non-pribadi dengan pengguna dan foto individu.

Note

Bahasa kebijakan Cedar menganggap suatu entitas menjadi `in` dirinya sendiri. Oleh karena `principal in User::"Alice"` itu, setara dengan `principal == User::"Alice"`.

```
permit (  
  principal in PhotoFlash::User::"Alice",  
  action in PhotoFlash::Action::"SharePhotoLimitedAccess",  
  resource in PhotoFlash::Photo::"VacationPhoto94.jpg"  
);
```

Contoh berikut menunjukkan cara membuat kebijakan tertaut templat yang menggunakan templat kebijakan Berikan akses terbatas ke foto bersama non-pribadi dengan pengguna dan album individual.

```
permit (  
  principal in PhotoFlash::User::"Alice",  
  action in PhotoFlash::Action::"SharePhotoLimitedAccess",  
  resource in PhotoFlash::Album::"Italy2023"  
);
```

Contoh berikut menunjukkan cara membuat kebijakan tertaut templat yang menggunakan templat kebijakan Berikan akses terbatas ke foto bersama non-pribadi dengan grup teman dan foto individual.

```
permit (  
  principal in PhotoFlash::FriendGroup::"Jane::MySchoolFriends",  
  action in PhotoFlash::Action::"SharePhotoLimitedAccess",  
  resource in PhotoFlash::Photo::"VacationPhoto94.jpg"  
);
```

Contoh berikut menunjukkan cara membuat kebijakan tertaut templat yang menggunakan templat kebijakan Berikan akses terbatas ke foto bersama non-pribadi dengan grup teman dan album.

```
permit (  
  principal in PhotoFlash::FriendGroup::"Jane::MySchoolFriends",  
  action in PhotoFlash::Action::"SharePhotoLimitedAccess",  
  resource in PhotoFlash::Album::"Italy2023"  
);
```

Contoh berikut menunjukkan cara membuat kebijakan tertaut templat yang menggunakan templat kebijakan Berikan akses penuh ke foto bersama non-pribadi dengan grup teman dan foto individual.

```
permit (  
  principal in PhotoFlash::UserGroup::"Jane::MySchoolFriends",  
  action in PhotoFlash::Action::"SharePhotoFullAccess",  
  resource in PhotoFlash::Photo::"VacationPhoto94.jpg"  
);
```

Contoh berikut menunjukkan cara membuat kebijakan tertaut templat yang menggunakan templat kebijakan Memblokir pengguna dari akun.

```
forbid(  
  principal == PhotoFlash::User::"Bob",  
  action,  
  resource in PhotoFlash::Account::"Alice-account"  
);
```

DigitalPetStore contoh

Penyimpanan kebijakan DigitalPetStore sampel tidak menyertakan templat kebijakan apa pun. Anda dapat melihat kebijakan yang disertakan dengan penyimpanan kebijakan dengan memilih Kebijakan di panel navigasi di sebelah kiri setelah membuat penyimpanan kebijakan DigitalPetStore contoh.

TinyToDo contoh

Contoh berikut menunjukkan cara membuat kebijakan tertaut templat yang menggunakan templat kebijakan yang memberikan akses penampil untuk setiap pengguna dan daftar tugas.

```
permit (  
  principal == TinyToDo::User::"https://cognito-idp.us-east-1.amazonaws.com/us-east-1_h2aKCU1ts|5ae0c4b1-6de8-4dff-b52e-158188686f31|bob",  
  action in [TinyToDo::Action::"ReadList", TinyToDo::Action::"ListTasks"],  
  resource == TinyToDo::List::"1"  
);
```

Contoh berikut menunjukkan cara membuat kebijakan tertaut templat yang menggunakan templat kebijakan yang memberikan akses editor untuk pengguna individu dan daftar tugas.

```
permit (  
  principal == TinyToDo::User::"https://cognito-idp.us-east-1.amazonaws.com/us-east-1_h2aKCU1ts|5ae0c4b1-6de8-4dff-b52e-158188686f31|bob",  
  action in [  
    TinyToDo::Action::"ReadList",  
    TinyToDo::Action::"UpdateList",  
    TinyToDo::Action::"ListTasks",  
    TinyToDo::Action::"CreateTask",  
    TinyToDo::Action::"UpdateTask",  
    TinyToDo::Action::"DeleteTask"  
  ],  
  resource == TinyToDo::List::"1"  
);
```

Amankan aplikasi Anda dengan sumber identitas dan token

Amankan aplikasi Anda dengan cepat dengan membuat sumber identitas untuk mewakili penyedia identitas eksternal (iDP) di Izin Terverifikasi Amazon. Sumber identitas memberikan informasi dari pengguna yang diautentikasi dengan IDP yang memiliki hubungan kepercayaan dengan toko kebijakan Anda. Saat aplikasi Anda membuat permintaan otorisasi dengan token dari sumber identitas, toko kebijakan Anda dapat membuat keputusan otorisasi dari properti pengguna dan izin akses. Anda dapat menambahkan kumpulan pengguna Amazon Cognito atau iDP OpenID Connect (OIDC) kustom sebagai sumber identitas Anda.

Anda dapat menggunakan penyedia identitas [OpenID Connect \(OIDC\) \(\)](#) dengan Izin Terverifikasi IDPs. Aplikasi Anda dapat menghasilkan permintaan otorisasi dengan token web JSON (JWTs) yang dihasilkan oleh penyedia identitas yang sesuai dengan OIDC. Identitas pengguna dalam token dipetakan ke ID utama. Dengan token ID, Izin Terverifikasi memetakan klaim atribut ke atribut utama. Dengan token Access, klaim ini dipetakan ke [konteks](#). Dengan kedua jenis token, Anda dapat memetakan klaim seperti `groups` ke grup utama, dan membuat kebijakan yang mengevaluasi kontrol akses berbasis peran (RBAC).

Note

Izin Terverifikasi membuat keputusan otorisasi berdasarkan informasi dari token iDP tetapi tidak berinteraksi langsung dengan IDP dengan cara apa pun.

Untuk step-by-step panduan yang membangun logika otorisasi untuk Amazon API Gateway REST APIs menggunakan kumpulan Amazon Cognito pengguna atau penyedia identitas OIDC, lihat [Mengotorisasi menggunakan Izin Terverifikasi API Gateway APIs Amazon dengan Amazon Cognito atau membawa penyedia identitas Anda sendiri](#) di Blog Keamanan.AWS

Topik

- [Memilih penyedia identitas yang tepat](#)
- [Bekerja dengan sumber Amazon Cognito identitas](#)
- [Bekerja dengan sumber identitas OIDC](#)

Memilih penyedia identitas yang tepat

Meskipun Izin Terverifikasi berfungsi dengan beragam IdPs, pertimbangkan hal berikut saat memutuskan mana yang akan digunakan dalam aplikasi Anda:

Gunakan Amazon Cognito saat:

- Anda sedang membangun aplikasi baru tanpa infrastruktur identitas yang ada
- Anda ingin kumpulan pengguna AWS yang dikelola dengan fitur keamanan bawaan
- Anda membutuhkan integrasi penyedia identitas sosial
- Anda ingin manajemen token yang disederhanakan

Gunakan penyedia OIDC saat:

- Anda memiliki infrastruktur identitas yang ada (Auth0, Okta, Azure AD)
- Anda perlu mempertahankan manajemen pengguna yang terpusat
- Anda memiliki persyaratan kepatuhan untuk spesifik IdPs

Bekerja dengan sumber Amazon Cognito identitas

Izin Terverifikasi bekerja sama dengan kumpulan pengguna Amazon Cognito. Amazon Cognito JWTs memiliki struktur yang dapat diprediksi. Izin Terverifikasi mengenali struktur ini dan menarik manfaat maksimal dari informasi yang dikandungnya. Misalnya, Anda dapat menerapkan model otorisasi kontrol akses berbasis peran (RBAC) dengan token ID atau token akses.

Sumber identitas kumpulan pengguna Amazon Cognito baru memerlukan informasi berikut:

- Itu AWS Region.
- ID kolam pengguna.
- Jenis entitas utama yang ingin Anda kaitkan dengan sumber identitas Anda, misalnya `MyCorp::User`.
- Jenis entitas grup utama yang ingin Anda kaitkan dengan sumber identitas Anda, misalnya `MyCorp::UserGroup`.
- Klien IDs dari kumpulan pengguna Anda yang ingin Anda otorisasi untuk mengajukan permintaan ke toko kebijakan Anda.

Karena Izin Terverifikasi hanya berfungsi dengan kumpulan pengguna Amazon Cognito dalam Akun AWS hal yang sama, Anda tidak dapat menentukan sumber identitas di akun lain. Izin Terverifikasi menyetel awalan entitas —pengenal sumber identitas yang harus Anda referensikan dalam kebijakan yang bertindak pada prinsip kumpulan pengguna—ke ID kumpulan pengguna Anda, misalnya. `us-west-2_EXAMPLE` Dalam hal ini, Anda akan mereferensikan pengguna di kumpulan pengguna tersebut dengan ID `a1b2c3d4-5678-90ab-cdef-EXAMPLE22222` sebagai `us-west-2_EXAMPLE|a1b2c3d4-5678-90ab-cdef-EXAMPLE22222`

Klaim token kumpulan pengguna dapat berisi atribut, cakupan, grup, klien IDs, dan data khusus. [Amazon Cognito JWTs](#) memiliki kemampuan untuk memasukkan berbagai informasi yang dapat berkontribusi pada keputusan otorisasi dalam Izin Terverifikasi. Ini termasuk:

1. Nama pengguna dan klaim grup dengan `cognito: awalan`
2. [Atribut pengguna kustom](#) dengan `custom: prefix`
3. Klaim khusus ditambahkan saat runtime
4. Klaim standar OIDC seperti `dan sub email`

Kami membahas klaim ini secara rinci, dan cara mengelolanya dalam kebijakan Izin Terverifikasi, di [Memetakan Amazon Cognito token ke skema](#).

Important

Meskipun Anda dapat mencabut Amazon Cognito token sebelum kedaluwarsa, JWTs dianggap sebagai sumber daya tanpa kewarganegaraan yang mandiri dengan tanda tangan dan validitas. Layanan yang sesuai dengan [JSON Web Token RFC 7519](#) diharapkan dapat memvalidasi token dari jarak jauh dan tidak diharuskan untuk memvalidasinya dengan penerbit. Ini berarti bahwa Izin Terverifikasi dimungkinkan untuk memberikan akses berdasarkan token yang dicabut atau dikeluarkan untuk pengguna yang kemudian dihapus. Untuk mengurangi risiko ini, kami sarankan Anda membuat token dengan durasi validitas sesingkat mungkin dan mencabut token penyegaran saat Anda ingin menghapus otorisasi untuk melanjutkan sesi pengguna. Untuk informasi selengkapnya, lihat [Mengakhiri sesi pengguna dengan pencabutan token](#)

Contoh berikut ini menunjukkan cara membuat kebijakan yang mereferensikan beberapa klaim kumpulan pengguna Amazon Cognito yang terkait dengan prinsipal.

```

permit(
    principal,
    action,
    resource == ExampleCo::Photo::"VacationPhoto94.jpg"
)
when {
    principal["cognito:username"] == "alice" &&
    principal["custom:department"] == "Finance"
};

```

Contoh berikut ini menunjukkan bagaimana Anda dapat membuat kebijakan yang mereferensikan prinsipal yang merupakan pengguna di kumpulan pengguna Cognito. Perhatikan bahwa ID utama berbentuk "`<userpool-id>|<sub>`".

```

permit(
    principal == ExampleCo::User::"us-east-1_example|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    action,
    resource == ExampleCo::Photo::"VacationPhoto94.jpg"
);

```

Kebijakan cedar untuk sumber identitas kumpulan pengguna di Izin Terverifikasi menggunakan sintaks khusus untuk nama klaim yang berisi karakter selain alfanumerik dan garis bawah (`.`). `_` Ini termasuk klaim awalan kumpulan pengguna yang berisi `:` karakter, suka `cognito:username` dan `custom:department`. Untuk menulis kondisi kebijakan yang mereferensikan `cognito:username` atau `custom:department` klaim, tuliskan sebagai `principal["cognito:username"]` dan `principal["custom:department"]`, masing-masing.

Note

Jika token berisi klaim dengan `custom:` awalan `cognito:` atau dan nama klaim dengan nilai literal `cognito` atau `custom`, permintaan otorisasi dengan [IsAuthorizedWithToken](#) akan gagal dengan `ValidationException`

Untuk informasi selengkapnya tentang pemetaan klaim, lihat [Memetakan Amazon Cognito token ke skema](#). Untuk informasi selengkapnya tentang otorisasi bagi Amazon Cognito pengguna, lihat [Otorisasi dengan Izin Terverifikasi Amazon](#) di Panduan Pengembang Amazon Cognito.

Topik

- [Membuat sumber Amazon Cognito identitas Izin Terverifikasi Amazon](#)
- [Mengedit sumber Amazon Cognito identitas Izin Terverifikasi Amazon](#)
- [Memetakan Amazon Cognito token ke skema](#)
- [Validasi klien dan audiens untuk Amazon Cognito](#)

Membuat sumber Amazon Cognito identitas Izin Terverifikasi Amazon

Prosedur berikut menambahkan sumber identitas ke toko kebijakan yang ada.

Anda juga dapat membuat sumber identitas saat [membuat penyimpanan kebijakan baru](#) di konsol Izin Terverifikasi. Dalam proses ini, Anda dapat secara otomatis mengimpor klaim dalam token sumber identitas Anda ke atribut entitas. Pilih opsi Pengaturan terpandu atau Siapkan dengan API Gateway dan penyedia identitas. Opsi ini juga membuat kebijakan awal.

Note

Sumber identitas tidak tersedia di panel navigasi di sebelah kiri hingga Anda membuat toko kebijakan. Sumber identitas yang Anda buat terkait dengan penyimpanan kebijakan saat ini.

Anda dapat mengabaikan tipe entitas utama saat membuat sumber identitas dengan [create-identity-source](#) di AWS CLI atau di API [CreateIdentitySource](#) Izin Terverifikasi. Namun, tipe entitas kosong menciptakan sumber identitas dengan tipe entitas `AWS::Cognito`. Nama entitas ini tidak kompatibel dengan skema penyimpanan kebijakan. Untuk mengintegrasikan Amazon Cognito identitas dengan skema penyimpanan kebijakan, Anda harus menyetel jenis entitas utama ke entitas penyimpanan kebijakan yang didukung.

Konsol Manajemen AWS

Untuk membuat sumber identitas kumpulan pengguna Amazon Cognito

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Sumber identitas.
3. Pilih Buat sumber identitas.
4. Di detail kumpulan pengguna Cognito, pilih AWS Region dan masukkan ID kumpulan Pengguna untuk sumber identitas Anda.

5. Dalam konfigurasi Principal, untuk tipe Principal, pilih tipe entitas untuk prinsipal dari sumber ini. Identitas dari kumpulan pengguna Amazon Cognito yang terhubung akan dipetakan ke tipe utama yang dipilih.
 6. Dalam Konfigurasi grup, pilih Gunakan grup Cognito jika Anda ingin memetakan klaim kumpulan `cognito:groups` pengguna. Pilih tipe entitas yang merupakan induk dari tipe utama.
 7. Dalam validasi aplikasi Klien, pilih apakah akan memvalidasi aplikasi klien. IDs
 - Untuk memvalidasi aplikasi klien IDs, pilih Hanya terima token dengan aplikasi IDs klien yang cocok. Pilih Tambahkan ID aplikasi klien baru untuk setiap ID aplikasi klien untuk memvalidasi. Untuk menghapus ID aplikasi klien yang telah ditambahkan, pilih Hapus di sebelah ID aplikasi klien.
 - Pilih Jangan memvalidasi aplikasi klien IDs jika Anda tidak ingin memvalidasi aplikasi klien. IDs
 8. Pilih Buat sumber identitas.
 9. (Opsional) Jika toko kebijakan Anda memiliki skema, sebelum Anda dapat mereferensikan atribut yang Anda ekstrak dari identitas atau token akses dalam kebijakan Cedar Anda, Anda harus memperbarui skema Anda untuk membuat Cedar mengetahui jenis prinsipal yang dibuat oleh sumber identitas Anda. Penambahan skema itu harus menyertakan atribut yang ingin Anda referensikan dalam kebijakan Cedar Anda. Untuk informasi selengkapnya tentang pemetaan atribut Amazon Cognito token ke atribut utama Cedar, lihat. [Memetakan Amazon Cognito token ke skema](#)
-  **Note**

Saat Anda membuat [penyimpanan kebijakan terkait API atau menggunakan Penyiapan](#) dengan API Gateway dan penyedia identitas saat membuat penyimpanan kebijakan, Izin Terverifikasi akan menanyakan kumpulan pengguna Anda untuk atribut pengguna dan membuat skema tempat tipe utama Anda diisi dengan atribut kumpulan pengguna.
10. Buat kebijakan yang menggunakan informasi dari token untuk membuat keputusan otorisasi. Untuk informasi selengkapnya, lihat [Membuat kebijakan statis Izin Terverifikasi Amazon](#).

Setelah Anda membuat sumber identitas, memperbarui skema, dan membuat kebijakan, gunakan `IsAuthorizedWithToken` agar Izin Terverifikasi membuat keputusan otorisasi. Untuk informasi selengkapnya, lihat [IsAuthorizedWithToken](#) di panduan referensi API Izin Terverifikasi Amazon.

AWS CLI

Untuk membuat sumber identitas kumpulan pengguna Amazon Cognito

Anda dapat membuat sumber identitas dengan menggunakan [CreateIdentitySource](#) operasi. Contoh berikut membuat sumber identitas yang dapat mengakses identitas yang diautentikasi dari kumpulan Amazon Cognito pengguna.

1. Buat `config.txt` file yang berisi rincian berikut dari kumpulan Amazon Cognito pengguna untuk digunakan oleh `--configuration` parameter dalam `create-identity-source` perintah.


```
{
  "cognitoUserPoolConfiguration": {
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_1a2b3c4d5",
    "clientIds": ["a1b2c3d4e5f6g7h8i9j0kalbmc"],
    "groupConfiguration": {
      "groupEntityType": "MyCorp::UserGroup"
    }
  }
}
```

2. Jalankan perintah berikut untuk membuat sumber Amazon Cognito identitas.

```
$ aws verifiedpermissions create-identity-source \
  --configuration file://config.txt \
  --principal-entity-type "User" \
  --policy-store-id 123456789012
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEabcdefg111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

3. (Opsional) Jika toko kebijakan Anda memiliki skema, sebelum Anda dapat mereferensikan atribut yang Anda ekstrak dari identitas atau token akses dalam kebijakan Cedar Anda, Anda

harus memperbarui skema Anda untuk membuat Cedar mengetahui jenis prinsipal yang dibuat oleh sumber identitas Anda. Penambahan skema itu harus menyertakan atribut yang ingin Anda referensikan dalam kebijakan Cedar Anda. Untuk informasi selengkapnya tentang pemetaan atribut Amazon Cognito token ke atribut utama Cedar, lihat [Memetakan Amazon Cognito token ke skema](#)

 Note

Saat Anda membuat [penyimpanan kebijakan terkait API atau menggunakan Penyiapan](#) dengan API Gateway dan penyedia identitas saat membuat penyimpanan kebijakan, Izin Terverifikasi akan menanyakan kumpulan pengguna Anda untuk atribut pengguna dan membuat skema tempat tipe utama Anda diisi dengan atribut kumpulan pengguna.

4. Buat kebijakan yang menggunakan informasi dari token untuk membuat keputusan otorisasi. Untuk informasi selengkapnya, lihat [Membuat kebijakan statis Izin Terverifikasi Amazon](#).

Setelah Anda membuat sumber identitas, memperbarui skema, dan membuat kebijakan, gunakan `IsAuthorizedWithToken` agar Izin Terverifikasi membuat keputusan otorisasi. Untuk informasi selengkapnya, lihat [IsAuthorizedWithToken](#) di panduan referensi API Izin Terverifikasi Amazon.

Untuk informasi selengkapnya tentang penggunaan akses Amazon Cognito dan token identitas untuk pengguna yang diautentikasi di Izin Terverifikasi, lihat Otorisasi [dengan Izin Terverifikasi Amazon di Panduan Pengembang Amazon](#) Cognito.

Mengedit sumber Amazon Cognito identitas Izin Terverifikasi Amazon

Anda dapat mengedit beberapa parameter sumber identitas Anda setelah Anda membuatnya. Anda tidak dapat mengubah jenis sumber identitas, Anda harus menghapus sumber identitas dan membuat yang baru untuk beralih dari ke OIDC atau OIDC Amazon Cognito ke. Amazon Cognito Jika skema penyimpanan kebijakan Anda cocok dengan atribut sumber identitas Anda, perhatikan bahwa Anda harus memperbarui skema secara terpisah untuk mencerminkan perubahan yang Anda buat pada sumber identitas Anda.

Konsol Manajemen AWS

Untuk memperbarui sumber Amazon Cognito identitas

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Sumber identitas.
3. Pilih ID sumber identitas yang akan diedit.
4. Pilih Edit.
5. Di detail kumpulan pengguna Cognito, pilih AWS Region dan ketik ID kumpulan Pengguna untuk sumber identitas Anda.
6. Dalam detail Principal, Anda dapat memperbarui tipe Principal untuk sumber identitas. Identitas dari kumpulan pengguna Amazon Cognito yang terhubung akan dipetakan ke tipe utama yang dipilih.
7. Dalam Konfigurasi grup, pilih Gunakan grup Cognito jika Anda ingin memetakan klaim kumpulan `cognito:groups` pengguna. Pilih tipe entitas yang merupakan induk dari tipe utama.
8. Dalam validasi aplikasi Klien, pilih apakah akan memvalidasi aplikasi klien. IDs
 - Untuk memvalidasi aplikasi klien IDs, pilih Hanya terima token dengan aplikasi IDs klien yang cocok. Pilih Tambahkan ID aplikasi klien baru untuk setiap ID aplikasi klien untuk memvalidasi. Untuk menghapus ID aplikasi klien yang telah ditambahkan, pilih Hapus di sebelah ID aplikasi klien.
 - Pilih Jangan memvalidasi aplikasi klien IDs jika Anda tidak ingin memvalidasi aplikasi klien. IDs
9. Pilih Simpan perubahan.
10. Jika Anda mengubah tipe utama untuk sumber identitas, Anda harus memperbarui skema Anda untuk mencerminkan tipe utama yang diperbarui dengan benar.

Anda dapat menghapus sumber identitas dengan memilih tombol radio di sebelah sumber identitas dan kemudian memilih Hapus sumber identitas. Ketik `delete` kotak teks dan kemudian pilih Hapus sumber identitas untuk mengonfirmasi penghapusan sumber identitas.

AWS CLI

Untuk memperbarui sumber Amazon Cognito identitas

Anda dapat memperbarui sumber identitas dengan menggunakan [UpdateIdentitySource](#) operasi. Contoh berikut memperbarui sumber identitas yang ditentukan untuk menggunakan kumpulan Amazon Cognito pengguna yang berbeda.

1. Buat `config.txt` file yang berisi rincian berikut dari kumpulan Amazon Cognito pengguna untuk digunakan oleh `--configuration` parameter dalam `update-identity-source` perintah.

```
{
  "cognitoUserPoolConfiguration": {
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_1a2b3c4d5",
    "clientIds": ["a1b2c3d4e5f6g7h8i9j0kalbmc"],
    "groupConfiguration": {
      "groupEntityType": "MyCorp::UserGroup"
    }
  }
}
```

2. Jalankan perintah berikut untuk memperbarui sumber Amazon Cognito identitas.

```
$ aws verifiedpermissions update-identity-source \
  --update-configuration file://config.txt \
  --policy-store-id 123456789012
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEabcdefg111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Note

Jika Anda mengubah tipe utama untuk sumber identitas, Anda harus memperbarui skema Anda untuk mencerminkan tipe utama yang diperbarui dengan benar.

Memetakan Amazon Cognito token ke skema

Anda mungkin menemukan bahwa Anda ingin menambahkan sumber identitas ke toko kebijakan dan klaim penyedia peta, atau token, ke skema toko kebijakan Anda. Anda dapat mengotomatiskan proses ini, dengan menggunakan [Penyiapan terpandu](#) untuk membuat penyimpanan kebijakan dengan sumber identitas, atau memperbarui skema secara manual setelah penyimpanan kebijakan dibuat. Setelah Anda memetakan token ke skema, Anda dapat membuat kebijakan yang mereferensikannya.

Bagian panduan pengguna ini memiliki informasi berikut:

- Bila Anda dapat secara otomatis mengisi atribut ke skema penyimpanan kebijakan
- Cara menggunakan klaim Amazon Cognito token dalam kebijakan Izin Terverifikasi
- Cara membuat skema untuk sumber identitas secara manual

Penyimpanan [kebijakan terkait API dan penyimpanan](#) kebijakan dengan sumber identitas yang dibuat melalui [penyiapan Terpandu](#) tidak memerlukan pemetaan manual atribut token identitas (ID) ke skema. Anda dapat memberikan Izin Terverifikasi dengan atribut di kumpulan pengguna dan membuat skema yang diisi dengan atribut pengguna. Dalam otorisasi token ID, Izin Terverifikasi memetakan klaim ke atribut entitas utama. Anda mungkin perlu memetakan Amazon Cognito token secara manual ke skema Anda dalam kondisi berikut:

- Anda membuat toko kebijakan kosong atau penyimpanan kebijakan dari sampel.
- Anda ingin memperluas penggunaan token akses di luar kontrol akses berbasis peran (RBAC).
- Anda membuat penyimpanan kebijakan dengan REST API Izin Terverifikasi, AWS SDK, atau AWS CDK

Untuk digunakan Amazon Cognito sebagai sumber identitas di toko kebijakan Izin Terverifikasi, Anda harus memiliki atribut penyedia dalam skema Anda. Skema diperbaiki dan harus sesuai dengan entitas yang dibuat oleh token penyedia [IsAuthorizedWithToken](#) atau permintaan [BatchIsAuthorizedWithToken](#) API. Jika Anda membuat toko kebijakan dengan cara yang secara otomatis mengisi skema Anda dari informasi penyedia dalam token ID, Anda siap untuk menulis kebijakan. Jika Anda membuat penyimpanan kebijakan tanpa skema untuk sumber identitas, Anda harus menambahkan atribut penyedia ke skema yang cocok dengan entitas yang dibuat menggunakan permintaan API. Kemudian Anda dapat menulis kebijakan menggunakan atribut dari token penyedia.

Untuk informasi selengkapnya tentang menggunakan ID Amazon Cognito dan token akses untuk pengguna yang diautentikasi di Izin Terverifikasi, lihat Otorisasi [dengan Izin Terverifikasi Amazon di Panduan Pengembang Amazon Cognito](#).

Topik

- [Memetakan token ID ke skema](#)
- [Memetakan token akses](#)
- [Notasi alternatif untuk klaim yang dibatasi titik Amazon Cognito dua](#)
- [Hal-hal yang perlu diketahui tentang pemetaan skema](#)

Memetakan token ID ke skema

Izin Terverifikasi memproses klaim token ID sebagai atribut pengguna: nama dan judul mereka, keanggotaan grup mereka, informasi kontak mereka. Token ID paling berguna dalam model otorisasi kontrol akses berbasis atribut (ABAC). Jika Anda ingin Izin Terverifikasi menganalisis akses ke sumber daya berdasarkan siapa yang membuat permintaan, pilih token ID untuk sumber identitas Anda.

Amazon Cognito Token ID bekerja dengan sebagian besar pustaka [relying-party OIDC](#). Mereka memperluas fitur OIDC dengan klaim tambahan. Aplikasi Anda dapat mengautentikasi pengguna dengan operasi API autentikasi kumpulan pengguna Amazon Cognito, atau dengan UI yang dihosting kumpulan pengguna. Untuk informasi selengkapnya, lihat [Menggunakan API dan titik akhir](#) di Panduan Amazon Cognito Pengembang.

Klaim yang berguna dalam token Amazon Cognito ID

cognito:username dan *preferred_username*

Varian dari nama pengguna pengguna.

sub

Pengenal pengguna unik pengguna (UUID)

Klaim dengan *custom:* awalan

Awalan untuk atribut kumpulan pengguna kustom seperti *custom:employmentStoreCode*.

Klaim standar

Standar OIDC mengklaim seperti email dan phone_number. Untuk informasi selengkapnya, lihat [Klaim standar](#) di OpenID Connect Core 1.0 yang menggabungkan errata set 2.

cognito:groups

Keanggotaan grup pengguna. Dalam model otorisasi berdasarkan kontrol akses berbasis peran (RBAC), klaim ini menyajikan peran yang dapat Anda evaluasi dalam kebijakan Anda.

Klaim sementara

Klaim yang bukan milik pengguna, tetapi ditambahkan saat runtime oleh kumpulan pengguna [Pre token generation Lambda trigger](#). Klaim transien menyerupai klaim standar tetapi berada di luar standar, misalnya tenant atau department.

Dalam kebijakan yang mereferensikan Amazon Cognito atribut yang memiliki pemisah, rujuk atribut dalam format `principal["cognito:username"]`. Klaim peran `cognito:groups` adalah pengecualian untuk aturan ini. Izin Terverifikasi memetakan konten klaim ini ke entitas induk entitas pengguna.

Untuk informasi selengkapnya tentang struktur token ID dari kumpulan pengguna Amazon Cognito, lihat [Menggunakan token ID di](#) Panduan Amazon Cognito Pengembang.

Contoh ID token berikut memiliki masing-masing dari empat jenis atribut. Ini termasuk klaim Amazon Cognito-spesifik `cognito:username`, klaim khusus `custom:employmentStoreCode`, klaim standar `email`, dan klaim sementara `tenant`.

```
{
  "sub": "91eb4550-XXX",
  "cognito:groups": [
    "Store-Owner-Role",
    "Customer"
  ],
  "email_verified": true,
  "clearance": "confidential",
  "iss": "https://cognito-idp.us-east-2.amazonaws.com/us-east-2_EXAMPLE",
  "cognito:username": "alice",
  "custom:employmentStoreCode": "petstore-dallas",
  "origin_jti": "5b9f50a3-05da-454a-8b99-b79c2349de77",
  "aud": "1example23456789",
  "event_id": "0ed5ad5c-7182-4ecf-XXX",
```

```
"token_use": "id",
"auth_time": 1687885407,
"department": "engineering",
"exp": 1687889006,
"iat": 1687885407,
"tenant": "x11app-tenant-1",
"jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111",
"email": "alice@example.com"
}
```

Saat membuat sumber identitas dengan kumpulan Amazon Cognito pengguna, Anda menentukan jenis entitas utama yang dihasilkan Izin Terverifikasi dalam permintaan otorisasi. `IsAuthorizedWithToken` Kebijakan Anda kemudian dapat menguji atribut prinsipal tersebut sebagai bagian dari evaluasi permintaan tersebut. Skema Anda mendefinisikan jenis dan atribut utama untuk sumber identitas, dan kemudian Anda dapat mereferensikannya dalam kebijakan Cedar Anda.

Anda juga menentukan jenis entitas grup yang ingin Anda peroleh dari klaim grup token ID. Dalam permintaan otorisasi, Izin Terverifikasi memetakan setiap anggota grup yang diklaim ke jenis entitas grup tersebut. Dalam kebijakan, Anda dapat mereferensikan entitas grup tersebut sebagai prinsipal.

Contoh berikut menunjukkan cara mencerminkan atribut dari token identitas contoh dalam skema Izin Terverifikasi Anda. Untuk informasi selengkapnya tentang mengedit skema Anda, lihat [Mengedit skema toko kebijakan](#). Jika konfigurasi sumber identitas Anda menentukan tipe utama `User`, maka Anda dapat menyertakan sesuatu yang mirip dengan contoh berikut untuk membuat atribut tersebut tersedia untuk Cedar.

```
"User": {
  "shape": {
    "type": "Record",
    "attributes": {
      "cognito:username": {
        "type": "String",
        "required": false
      },
      "custom:employmentStoreCode": {
        "type": "String",
        "required": false
      },
      "email": {
        "type": "String"
      }
    }
  }
}
```

```
    },
    "tenant": {
      "type": "String",
      "required": true
    }
  }
}
```

Untuk contoh kebijakan yang akan memvalidasi skema ini, lihat. [Mencerminkan atribut token Amazon Cognito ID](#)

Memetakan token akses

Izin Terverifikasi memproses klaim token akses selain klaim grup sebagai atribut tindakan, atau atribut konteks. Seiring dengan keanggotaan grup, token akses dari IDP Anda mungkin berisi informasi tentang akses API. Token akses berguna dalam model otorisasi yang menggunakan kontrol akses berbasis peran (RBAC). Model otorisasi yang mengandalkan klaim token akses selain keanggotaan grup memerlukan upaya tambahan dalam konfigurasi skema.

Amazon Cognito token akses memiliki klaim yang dapat digunakan untuk otorisasi:

Klaim yang berguna dalam token Amazon Cognito akses

client_id

ID aplikasi klien dari pihak yang mengandalkan OIDC. Dengan ID klien, Izin Terverifikasi dapat memverifikasi bahwa permintaan otorisasi berasal dari klien yang diizinkan untuk penyimpanan kebijakan. Dalam otorisasi machine-to-machine (M2M), sistem permintaan mengotorisasi permintaan dengan rahasia klien dan memberikan ID klien dan cakupan sebagai bukti otorisasi.

scope

[Cakupan OAuth 2.0](#) yang mewakili izin akses pembawa token.

cognito:groups

Keanggotaan grup pengguna. Dalam model otorisasi berdasarkan kontrol akses berbasis peran (RBAC), klaim ini menyajikan peran yang dapat Anda evaluasi dalam kebijakan Anda.

Klaim sementara

Klaim yang bukan merupakan izin akses, tetapi ditambahkan saat runtime oleh kumpulan pengguna [Pre token generation Lambda trigger](#). Klaim transien menyerupai klaim standar tetapi

berada di luar standar, misalnya tenant atau. department Kustomisasi token akses menambah biaya pada AWS tagihan Anda.

Untuk informasi selengkapnya tentang struktur token akses dari kumpulan pengguna Amazon Cognito, lihat [Menggunakan token akses di](#) Panduan Amazon Cognito Pengembang.

Token Amazon Cognito akses dipetakan ke objek konteks saat diteruskan ke Izin Terverifikasi. Atribut token akses dapat direferensikan menggunakan `context.token.attribute_name`. Contoh token akses berikut mencakup scope klaim `client_id` dan klaim.

```
{
  "sub": "91eb4550-9091-708c-a7a6-9758ef8b6b1e",
  "cognito:groups": [
    "Store-Owner-Role",
    "Customer"
  ],
  "iss": "https://cognito-idp.us-east-2.amazonaws.com/us-east-2_EXAMPLE",
  "client_id": "1example23456789",
  "origin_jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111",
  "event_id": "bda909cb-3e29-4bb8-83e3-ce6808f49011",
  "token_use": "access",
  "scope": "MyAPI/mydata.write",
  "auth_time": 1688092966,
  "exp": 1688096566,
  "iat": 1688092966,
  "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN22222222",
  "username": "alice"
}
```

Contoh berikut menunjukkan cara mencerminkan atribut dari token akses contoh dalam skema Izin Terverifikasi Anda. Untuk informasi selengkapnya tentang mengedit skema Anda, lihat [Mengedit skema toko kebijakan](#).

```
{
  "MyApplication": {
    "actions": {
      "Read": {
        "appliesTo": {
          "context": {
            "type": "ReusedContext"
          }
        }
      }
    }
  }
}
```


referensi untuk `cognito:groups` menjadi `principal.cognito.groups` dalam kebijakan Anda. Meskipun Anda dapat terus menggunakan format ini, kami sarankan Anda membuat skema dan kebijakan Anda dengan notasi [braket](#). Dalam format ini, referensi untuk `cognito:groups` menjadi `principal["cognito:groups"]` dalam kebijakan Anda. Skema yang dibuat secara otomatis untuk token ID kumpulan pengguna dari konsol Izin Terverifikasi menggunakan notasi braket.

Anda dapat terus menggunakan notasi titik dalam skema dan kebijakan yang dibuat secara manual untuk sumber identitas. Amazon Cognito Anda tidak dapat menggunakan notasi titik dengan `:` atau karakter non-alfanumerik lainnya dalam skema atau kebijakan untuk jenis OIDC IDP lainnya.

Skema untuk notasi titik bersarang setiap instance `:` karakter sebagai anak dari frasa custom awal `cognito` atau, seperti yang ditunjukkan pada contoh berikut:

```
"CognitoUser": {
  "shape": {
    "type": "Record",
    "attributes": {
      "cognito": {
        "type": "Record",
        "required": true,
        "attributes": {
          "username": {
            "type": "String",
            "required": true
          }
        }
      },
      "custom": {
        "type": "Record",
        "required": true,
        "attributes": {
          "employmentStoreCode": {
            "type": "String",
            "required": true
          }
        }
      },
      "email": {
        "type": "String"
      },
      "tenant": {
        "type": "String",
```

```
        "required": true
      }
    }
  }
}
```

Untuk contoh kebijakan yang akan memvalidasi skema ini dan menggunakan notasi titik, lihat.

[Menggunakan notasi titik untuk referensi atribut](#)

Hal-hal yang perlu diketahui tentang pemetaan skema

Pemetaan atribut berbeda antara jenis token

[Dalam otorisasi token akses, Izin Terverifikasi memetakan klaim ke konteks.](#) Dalam otorisasi token ID, Izin Terverifikasi memetakan klaim ke atribut utama. Untuk penyimpanan kebijakan yang Anda buat di konsol Izin Terverifikasi, hanya penyimpanan kebijakan kosong dan sampel yang tidak memiliki sumber identitas dan mengharuskan Anda mengisi skema Anda dengan atribut kumpulan pengguna untuk otorisasi token ID. Otorisasi token akses didasarkan pada kontrol akses berbasis peran (RBAC) dengan klaim keanggotaan grup dan tidak secara otomatis memetakan klaim lain ke skema penyimpanan kebijakan.

Atribut sumber identitas tidak diperlukan

Saat Anda membuat sumber identitas di konsol Izin Terverifikasi, tidak ada atribut yang ditandai sebagai wajib. Ini mencegah klaim yang hilang menyebabkan kesalahan validasi dalam permintaan otorisasi. Anda dapat mengatur atribut ke `required` sesuai kebutuhan, tetapi atribut tersebut harus ada di semua permintaan otorisasi.

RBAC tidak memerlukan atribut dalam skema

Skema untuk sumber identitas bergantung pada asosiasi entitas yang Anda buat saat menambahkan sumber identitas. Sumber identitas memetakan satu klaim ke jenis entitas pengguna, dan satu klaim ke jenis entitas grup. Pemetaan entitas ini adalah inti dari konfigurasi sumber identitas. Dengan informasi minimum ini, Anda dapat menulis kebijakan yang melakukan tindakan otorisasi untuk pengguna tertentu dan grup tertentu yang mungkin menjadi anggota pengguna, dalam model kontrol akses berbasis peran (RBAC). Penambahan klaim token ke skema memperluas cakupan otorisasi toko kebijakan Anda. Atribut pengguna dari token ID memiliki informasi tentang pengguna yang dapat berkontribusi pada otorisasi kontrol akses berbasis atribut (ABAC). Atribut konteks dari token akses memiliki informasi seperti cakupan OAuth 2.0 yang dapat menyumbangkan informasi kontrol akses tambahan dari penyedia Anda, tetapi memerlukan modifikasi skema tambahan.

Opsi Pengaturan dengan API Gateway dan penyedia identitas serta Penyiapan terpandu di konsol Izin Terverifikasi menetapkan klaim token ID ke skema. Ini tidak berlaku untuk klaim token akses. [Untuk menambahkan klaim token akses non-grup ke skema Anda, Anda harus mengedit skema Anda dalam mode JSON dan menambahkan atribut CommonTypes.](#) Untuk informasi selengkapnya, lihat [Memetakan token akses.](#)

Pilih jenis token

Cara penyimpanan kebijakan Anda bekerja dengan sumber identitas Anda bergantung pada keputusan kunci dalam konfigurasi sumber identitas: apakah Anda akan memproses ID atau token akses. Dengan penyedia Amazon Cognito identitas, Anda memiliki pilihan jenis token saat membuat toko kebijakan terkait API. Saat membuat [penyimpanan kebijakan terkait API](#), Anda harus memilih apakah Anda ingin menyiapkan otorisasi untuk ID atau token akses. Informasi ini memengaruhi atribut skema yang diterapkan Izin Terverifikasi untuk penyimpanan kebijakan Anda, dan sintaks otorisasi Lambda untuk API Anda. API Gateway Terutama jika Anda ingin mendapatkan keuntungan dari pemetaan otomatis klaim token ID ke atribut di konsol Izin Terverifikasi, putuskan lebih awal tentang jenis token yang ingin Anda proses sebelum Anda membuat sumber identitas Anda. Mengubah jenis token membutuhkan upaya yang signifikan untuk memfaktorkan ulang kebijakan dan skema Anda. Topik berikut menjelaskan penggunaan ID dan token akses dengan toko kebijakan.

Parser cedar membutuhkan tanda kurung untuk beberapa karakter

Kebijakan biasanya referensi atribut skema dalam format seperti `principal.username`.

Dalam kasus sebagian besar karakter non-alfanumerik seperti `:`, `.`, atau `/` yang mungkin muncul di nama klaim token, Izin Terverifikasi tidak dapat mengurai nilai kondisi seperti `atau.principal.cognito:username context.ip-address` Anda harus memformat kondisi ini dengan notasi braknet dalam format `principal["cognito:username"]` atau `context["ip-address"]`, masing-masing. Karakter garis bawah `_` adalah karakter yang valid dalam nama klaim, dan satu-satunya pengecualian non-alfanumerik untuk persyaratan ini.

Contoh skema parsi untuk atribut utama jenis ini terlihat seperti berikut:

```
"User": {
  "shape": {
    "type": "Record",
    "attributes": {
      "cognito:username": {
        "type": "String",
        "required": true
      }
    }
  },
```

```
    "custom:employmentStoreCode": {
      "type": "String",
      "required": true,
    },
    "email": {
      "type": "String",
      "required": false
    }
  }
}
```

Contoh sebagian skema untuk atribut konteks jenis ini terlihat seperti berikut:

```
"GetOrder": {
  "memberOf": [],
  "appliesTo": {
    "resourceTypes": [
      "Order"
    ],
    "context": {
      "type": "Record",
      "attributes": {
        "ip-address": {
          "required": false,
          "type": "String"
        }
      }
    }
  },
  "principalTypes": [
    "User"
  ]
}
```

Untuk contoh kebijakan yang akan memvalidasi skema ini, lihat. [Menggunakan notasi braket untuk referensi atribut token](#)

Validasi klien dan audiens untuk Amazon Cognito

Saat Anda menambahkan sumber identitas ke penyimpanan kebijakan, Izin Terverifikasi memiliki opsi konfigurasi yang memverifikasi bahwa ID dan token akses digunakan sebagaimana

dimaksud. Validasi ini terjadi dalam pemrosesan permintaan `IsAuthorizedWithToken` dan `BatchIsAuthorizedWithToken` API. Perilaku berbeda antara ID dan token akses, dan antara Amazon Cognito dan sumber identitas OIDC. Dengan penyedia kumpulan pengguna Amazon Cognito, Izin Terverifikasi dapat memvalidasi ID klien di ID dan token akses. Dengan penyedia OIDC, Izin Terverifikasi dapat memvalidasi ID klien dalam token ID, dan audiens dalam token akses.

ID klien adalah pengenal yang terkait dengan instance penyedia identitas yang digunakan aplikasi Anda, misalnya `example23456789`. Audiens adalah jalur URL yang terkait dengan pihak yang mengandalkan, atau tujuan, dari token akses, misalnya `https://mytoken.example.com`. Saat menggunakan token akses, aud klaim selalu dikaitkan dengan audiens.

Amazon Cognito Token ID memiliki aud klaim yang berisi ID [klien aplikasi](#). Token akses memiliki `client_id` klaim yang juga berisi ID klien aplikasi.

Saat Anda memasukkan satu atau beberapa nilai untuk validasi aplikasi Klien di sumber identitas Anda, Izin Terverifikasi membandingkan daftar klien aplikasi ini IDs dengan aud klaim token ID atau klaim token akses. `client_id` Izin Terverifikasi tidak memvalidasi URL audiens pihak terkait untuk sumber identitas. Amazon Cognito

Otorisasi sisi klien untuk JWTs

Anda mungkin ingin memproses token web JSON di aplikasi Anda dan meneruskan klaimnya ke Izin Terverifikasi tanpa menggunakan sumber identitas toko kebijakan. Anda dapat mengekstrak atribut entitas Anda dari JSON Web Token (JWT) dan menguraikannya menjadi Izin Terverifikasi.

Contoh ini menunjukkan bagaimana Anda dapat memanggil Izin Terverifikasi dari aplikasi menggunakan JWT.¹

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

    let principalEntity = {
        entityType: "PhotoFlash::User", // the application needs to fill in the
relevant user type
        entityId: payload["sub"], // the application need to use the claim that
represents the user-id
    };
    let resourceEntity = {
        entityType: "PhotoFlash::Photo", //the application needs to fill in the
relevant resource type
```

```
    entityId: "jane_photo_123.jpg", // the application needs to fill in the
relevant resource id
  };
  let action = {
    actionType: "PhotoFlash::Action", //the application needs to fill in the
relevant action id
    actionId: "GetPhoto", //the application needs to fill in the relevant action
type
  };
  let entities = {
    entityList: [],
  };
  entities.entityList.push(...getUserEntitiesFromToken(payload));
  let policyStoreId = "PSEXAMPLEabcdefghijklmnop111111"; // set your own policy store id

  const authResult = await client
    .isAuthorized({
      policyStoreId: policyStoreId,
      principal: principalEntity,
      resource: resourceEntity,
      action: action,
      entities,
    })
    .promise();

  return authResult;
}

function getUserEntitiesFromToken(payload) {
  let attributes = {};
  let claimsNotPassedInEntities = ['aud', 'sub', 'exp', 'jti', 'iss'];
  Object.entries(payload).forEach(([key, value]) => {
    if (claimsNotPassedInEntities.includes(key)) {
      return;
    }
    if (Array.isArray(value)) {
      var attributeItem = [];
      value.forEach((item) => {
        attributeItem.push({
          string: item,
        });
      });
      attributes[key] = {
```

```
        set: attributeItem,
    };
    } else if (typeof value === 'string') {
        attributes[key] = {
            string: value,
        }
    } else if (typeof value === 'bigint' || typeof value === 'number') {
        attributes[key] = {
            long: value,
        }
    } else if (typeof value === 'boolean') {
        attributes[key] = {
            boolean: value,
        }
    }
});

let entityItem = {
    attributes: attributes,
    identifier: {
        entityType: "PhotoFlash::User",
        entityId: payload["sub"], // the application needs to use the claim that
represents the user-id
    }
};
return [entityItem];
}
```

¹ Contoh kode ini menggunakan [aws-jwt-verify](#) pustaka untuk memverifikasi JWTs ditandatangani oleh IdPs OIDC-kompatibel.

Bekerja dengan sumber identitas OIDC

Anda juga dapat mengonfigurasi IdP OpenID Connect (OIDC) yang sesuai sebagai sumber identitas penyimpanan kebijakan. Penyedia OIDC mirip dengan kumpulan pengguna Amazon Cognito: mereka JWTs menghasilkan sebagai produk otentikasi. Untuk menambahkan penyedia OIDC, Anda harus memberikan URL penerbit

Sumber identitas OIDC baru memerlukan informasi berikut:

- URL penerbit. Izin Terverifikasi harus dapat menemukan `.well-known/openid-configuration` titik akhir di URL ini.
- Catatan CNAME yang tidak termasuk kartu liar. Misalnya, tidak `a.example.com` dapat dipetakan ke `*.example.net`. Sebaliknya, tidak `*.example.com` dapat dipetakan ke `a.example.net`
- Jenis token yang ingin Anda gunakan dalam permintaan otorisasi. Dalam hal ini, Anda memilih token Identity.
- Jenis entitas pengguna yang ingin Anda kaitkan dengan sumber identitas Anda, misalnya `MyCorp::User`.
- Jenis entitas grup yang ingin Anda kaitkan dengan sumber identitas Anda, misalnya `MyCorp::UserGroup`.
- Contoh token ID, atau definisi klaim dalam token ID.
- Awalan yang ingin Anda terapkan ke entitas IDs pengguna dan grup. Di CLI dan API, Anda dapat memilih awalan ini. Di penyimpanan kebijakan yang Anda buat dengan opsi Penyiapan dengan API Gateway dan penyedia identitas atau Penyiapan terpandu, Izin Terverifikasi menetapkan awalan nama penerbit dikurangi `https://`, misalnya `MyCorp::User::"auth.example.com|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"`

Untuk informasi selengkapnya tentang penggunaan operasi API untuk mengotorisasi permintaan dari sumber OIDC, lihat [Operasi API yang tersedia untuk otorisasi](#)

Contoh berikut ini menunjukkan bagaimana Anda dapat membuat kebijakan yang memungkinkan akses ke laporan akhir tahun untuk karyawan di departemen akuntansi, memiliki klasifikasi rahasia, dan tidak berada di kantor satelit. Izin Terverifikasi memperoleh atribut ini dari klaim dalam token ID prinsipal.

Perhatikan bahwa saat mereferensikan grup di prinsipal, Anda harus menggunakan `in` operator agar kebijakan dapat dievaluasi dengan benar.

```
permit(  
    principal in MyCorp::UserGroup::"MyOIDCProvider|Accounting",  
    action,  
    resource in MyCorp::Folder::"YearEnd2024"  
) when {  
    principal.jobClassification == "Confidential" &&  
    !(principal.location like "SatelliteOffice*")  
};
```

Topik

- [Membuat Izin Terverifikasi Amazon Sumber identitas OIDC](#)
- [Mengedit Izin Terverifikasi Amazon Sumber identitas OIDC](#)
- [Memetakan token OIDC ke skema](#)
- [Validasi klien dan audiens untuk penyedia OIDC](#)

Membuat Izin Terverifikasi Amazon Sumber identitas OIDC

Prosedur berikut menambahkan sumber identitas ke toko kebijakan yang ada.

Anda juga dapat membuat sumber identitas saat [membuat penyimpanan kebijakan baru](#) di konsol Izin Terverifikasi. Dalam proses ini, Anda dapat secara otomatis mengimpor klaim dalam token sumber identitas Anda ke atribut entitas. Pilih opsi Pengaturan terpandu atau Siapkan dengan API Gateway dan penyedia identitas. Opsi ini juga membuat kebijakan awal.

Note

Sumber identitas tidak tersedia di panel navigasi di sebelah kiri hingga Anda membuat toko kebijakan. Sumber identitas yang Anda buat terkait dengan penyimpanan kebijakan saat ini.

Anda dapat mengabaikan tipe entitas utama saat membuat sumber identitas dengan [create-identity-sourcedi](#) AWS CLI atau di API [CreateIdentitySource](#)Izin Terverifikasi. Namun, tipe entitas kosong menciptakan sumber identitas dengan tipe entitas `AWS::Cognito`. Nama entitas ini tidak kompatibel dengan skema penyimpanan kebijakan. Untuk mengintegrasikan Amazon Cognito identitas dengan skema penyimpanan kebijakan, Anda harus menyetel jenis entitas utama ke entitas penyimpanan kebijakan yang didukung.

Konsol Manajemen AWS

Untuk membuat sumber identitas OpenID Connect (OIDC)

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Sumber identitas.
3. Pilih Buat sumber identitas.
4. Pilih penyedia OIDC eksternal.

5. Di URL Penerbit, masukkan URL penerbit OIDC Anda. Ini adalah titik akhir layanan yang menyediakan server otorisasi, kunci penandatanganan, dan informasi lain tentang penyedia Anda, misalnya. <https://auth.example.com> URL penerbit Anda harus meng-host dokumen penemuan OIDC di `/.well-known/openid-configuration`
6. Pada tipe Token, pilih jenis OIDC JWT yang Anda ingin aplikasi Anda kirimkan untuk otorisasi. Untuk informasi selengkapnya, lihat [Memetakan token OIDC ke skema](#).
7. Dalam klaim token Peta ke entitas skema, pilih entitas Pengguna dan klaim Pengguna untuk sumber identitas. Entitas Pengguna adalah entitas di toko kebijakan yang ingin Anda rujuk ke pengguna dari penyedia OIDC Anda. Klaim Pengguna adalah klaim, biasanya `sub`, dari ID atau token akses Anda yang memegang pengenal unik untuk entitas yang akan dievaluasi. Identitas dari IdP OIDC yang terhubung akan dipetakan ke tipe utama yang dipilih.
8. (Opsional) Dalam klaim token Peta ke entitas skema, pilih entitas Grup dan klaim Grup untuk sumber identitas. Entitas Grup adalah [induk](#) dari entitas Pengguna. Klaim grup dipetakan ke entitas ini. Klaim Grup adalah klaim, biasanya `groups`, dari ID atau token akses Anda yang berisi string, JSON, atau string nama grup pengguna yang dibatasi spasi untuk entitas yang akan dievaluasi. Identitas dari IdP OIDC yang terhubung akan dipetakan ke tipe utama yang dipilih.
9. Dalam validasi - opsional, masukkan klien IDs atau audiens URLs yang Anda ingin toko kebijakan Anda terima dalam permintaan otorisasi, jika ada.
10. Pilih Buat sumber identitas.
11. (Opsional) Jika toko kebijakan Anda memiliki skema, sebelum Anda dapat mereferensikan atribut yang Anda ekstrak dari identitas atau token akses dalam kebijakan Cedar Anda, Anda harus memperbarui skema Anda untuk membuat Cedar mengetahui jenis prinsipal yang dibuat oleh sumber identitas Anda. Penambahan skema itu harus menyertakan atribut yang ingin Anda referensikan dalam kebijakan Cedar Anda. Untuk informasi selengkapnya tentang pemetaan atribut token OIDC ke atribut utama Cedar, lihat [Memetakan token OIDC ke skema](#)
12. Buat kebijakan yang menggunakan informasi dari token untuk membuat keputusan otorisasi. Untuk informasi selengkapnya, lihat [Membuat kebijakan statis Izin Terverifikasi Amazon](#).

Setelah Anda membuat sumber identitas, memperbarui skema, dan membuat kebijakan, gunakan `IsAuthorizedWithToken` agar Izin Terverifikasi membuat keputusan otorisasi. Untuk informasi selengkapnya, lihat [IsAuthorizedWithToken](#) di panduan referensi API Izin Terverifikasi Amazon.

AWS CLI

Untuk membuat sumber identitas OIDC

Anda dapat membuat sumber identitas dengan menggunakan [CreateIdentitySource](#) operasi. Contoh berikut membuat sumber identitas yang dapat mengakses identitas yang diautentikasi dari penyedia identitas OIDC (iDP).

1. Buat `config.txt` file yang berisi rincian berikut dari IDP OIDC untuk digunakan oleh `--configuration` parameter perintah. `create-identity-source`

```
{
  "openIdConnectConfiguration": {
    "issuer": "https://auth.example.com",
    "tokenSelection": {
      "identityTokenOnly": {
        "clientIds": ["1example23456789"],
        "principalIdClaim": "sub"
      },
    },
    "entityIdPrefix": "MyOIDCProvider",
    "groupConfiguration": {
      "groupClaim": "groups",
      "groupEntityType": "MyCorp::UserGroup"
    }
  }
}
```

2. Jalankan perintah berikut untuk membuat sumber identitas OIDC.

```
$ aws verifiedpermissions create-identity-source \
  --configuration file://config.txt \
  --principal-entity-type "User" \
  --policy-store-id 123456789012
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEabcdefg111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

3. (Opsional) Jika toko kebijakan Anda memiliki skema, sebelum Anda dapat mereferensikan atribut yang Anda ekstrak dari identitas atau token akses dalam kebijakan Cedar Anda,

Anda harus memperbarui skema Anda untuk membuat Cedar mengetahui jenis prinsipal yang dibuat oleh sumber identitas Anda. Penambahan skema itu harus menyertakan atribut yang ingin Anda referensikan dalam kebijakan Cedar Anda. Untuk informasi selengkapnya tentang pemetaan atribut token OIDC ke atribut utama Cedar, lihat [Memetakan token OIDC ke skema](#)

4. Buat kebijakan yang menggunakan informasi dari token untuk membuat keputusan otorisasi. Untuk informasi selengkapnya, lihat [Membuat kebijakan statis Izin Terverifikasi Amazon](#).

Setelah Anda membuat sumber identitas, memperbarui skema, dan membuat kebijakan, gunakan `IsAuthorizedWithToken` agar Izin Terverifikasi membuat keputusan otorisasi. Untuk informasi selengkapnya, lihat [IsAuthorizedWithToken](#) di panduan referensi API Izin Terverifikasi Amazon.

Mengedit Izin Terverifikasi Amazon Sumber identitas OIDC

Anda dapat mengedit beberapa parameter sumber identitas Anda setelah Anda membuatnya. Anda tidak dapat mengubah jenis sumber identitas, Anda harus menghapus sumber identitas dan membuat yang baru untuk beralih dari ke OIDC atau OIDC Amazon Cognito ke. Amazon Cognito Jika skema penyimpanan kebijakan Anda cocok dengan atribut sumber identitas Anda, perhatikan bahwa Anda harus memperbarui skema secara terpisah untuk mencerminkan perubahan yang Anda buat pada sumber identitas Anda.

Konsol Manajemen AWS

Untuk memperbarui sumber identitas OIDC

1. Buka [konsol Izin Terverifikasi](#). Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Sumber identitas.
3. Pilih ID sumber identitas yang akan diedit.
4. Pilih Edit.
5. Dalam detail penyedia OIDC, ubah URL Penerbit sesuai kebutuhan.
6. Dalam klaim token Peta ke atribut skema, ubah asosiasi antara klaim pengguna dan grup serta jenis entitas penyimpanan kebijakan, sesuai kebutuhan. Setelah mengubah jenis entitas, Anda harus memperbarui kebijakan dan atribut skema agar diterapkan pada tipe entitas baru.
7. Dalam validasi Audiens, tambahkan atau hapus nilai audiens yang ingin Anda terapkan.

8. Pilih Simpan perubahan.

Anda dapat menghapus sumber identitas dengan memilih tombol radio di sebelah sumber identitas dan kemudian memilih Hapus sumber identitas. Ketik delete kotak teks dan kemudian pilih Hapus sumber identitas untuk mengonfirmasi penghapusan sumber identitas.

AWS CLI

Untuk memperbarui sumber identitas OIDC

Anda dapat memperbarui sumber identitas dengan menggunakan [UpdateIdentitySource](#) operasi. Contoh berikut memperbarui sumber identitas yang ditentukan untuk menggunakan penyedia OIDC yang berbeda.

1. Buat `config.txt` file yang berisi rincian berikut dari IDP OIDC untuk digunakan oleh `--configuration` parameter perintah. `update-identity-source`

```
{
  "openIdConnectConfiguration": {
    "issuer": "https://auth2.example.com",
    "tokenSelection": {
      "identityTokenOnly": {
        "clientIds": ["2example10111213"],
        "principalIdClaim": "sub"
      },
    },
    "entityIdPrefix": "MyOIDCProvider",
    "groupConfiguration": {
      "groupClaim": "groups",
      "groupEntityType": "MyCorp::UserGroup"
    }
  }
}
```

2. Jalankan perintah berikut untuk memperbarui sumber identitas OIDC.

```
$ aws verifiedpermissions update-identity-source \
  --update-configuration file://config.txt \
  --policy-store-id 123456789012
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEabcdefg111111",
```

```
"lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",  
"policyStoreId": "PSEXAMPLEabcdefghijklmnop111111"  
}
```

Note

Jika Anda mengubah tipe utama untuk sumber identitas, Anda harus memperbarui skema Anda untuk mencerminkan tipe utama yang diperbarui dengan benar.

Memetakan token OIDC ke skema

Anda mungkin menemukan bahwa Anda ingin menambahkan sumber identitas ke toko kebijakan dan klaim penyedia peta, atau token, ke skema toko kebijakan Anda. Anda dapat mengotomatiskan proses ini, dengan menggunakan [Pengaturan terpandu](#) untuk membuat penyimpanan kebijakan Anda dengan sumber identitas, atau memperbarui skema Anda secara manual setelah penyimpanan kebijakan dibuat. Setelah Anda memetakan token ke skema, Anda dapat membuat kebijakan yang merujuknya.

Bagian panduan pengguna ini memiliki informasi berikut:

- Bila Anda dapat secara otomatis mengisi atribut ke skema penyimpanan kebijakan
- Cara membuat skema untuk sumber identitas secara manual

Penyimpanan [kebijakan terkait API dan penyimpanan](#) kebijakan dengan sumber identitas yang dibuat melalui [penyiapan Terpandu](#) tidak memerlukan pemetaan manual atribut token identitas (ID) ke skema. Anda dapat memberikan Izin Terverifikasi dengan atribut di kumpulan pengguna dan membuat skema yang diisi dengan atribut pengguna. Dalam otorisasi token ID, Izin Terverifikasi memetakan klaim ke atribut entitas utama.

Untuk menggunakan penyedia identitas OIDC (iDP) sebagai sumber identitas di toko kebijakan Izin Terverifikasi, Anda harus memiliki atribut penyedia dalam skema Anda. Skema diperbaiki dan harus sesuai dengan entitas yang dibuat oleh token penyedia [IsAuthorizedWithToken](#) atau permintaan [BatchIsAuthorizedWithToken](#) API. Jika Anda membuat toko kebijakan dengan cara yang secara otomatis mengisi skema Anda dari informasi penyedia dalam token ID, Anda siap untuk menulis kebijakan. Jika Anda membuat penyimpanan kebijakan tanpa skema untuk sumber identitas, Anda harus menambahkan atribut penyedia ke skema yang cocok dengan entitas yang dibuat

menggunakan permintaan API. Kemudian Anda dapat menulis kebijakan menggunakan atribut dari token penyedia.

Topik

- [Memetakan token ID ke skema](#)
- [Memetakan token akses](#)
- [Hal-hal yang perlu diketahui tentang pemetaan skema](#)

Memetakan token ID ke skema

Izin Terverifikasi memproses klaim token ID sebagai atribut pengguna: nama dan judul mereka, keanggotaan grup mereka, informasi kontak mereka. Token ID paling berguna dalam model otorisasi kontrol akses berbasis atribut (ABAC). Jika Anda ingin Izin Terverifikasi menganalisis akses ke sumber daya berdasarkan siapa yang membuat permintaan, pilih token ID untuk sumber identitas Anda.

Bekerja dengan token ID dari penyedia OIDC hampir sama dengan bekerja dengan token Amazon Cognito ID. Perbedaannya ada pada klaim. IDP Anda mungkin menampilkan [atribut OIDC standar](#), atau memiliki skema khusus. Saat membuat penyimpanan kebijakan baru di konsol Izin Terverifikasi, Anda dapat menambahkan sumber identitas OIDC dengan token ID contoh, atau Anda dapat memetakan klaim token secara manual ke atribut pengguna. Karena Izin Terverifikasi tidak mengetahui skema atribut IDP Anda, Anda harus memberikan informasi ini.

Untuk informasi selengkapnya, lihat [Membuat toko kebijakan Izin Terverifikasi](#).

Berikut ini adalah contoh skema untuk toko kebijakan dengan sumber identitas OIDC.

```
"User": {
  "shape": {
    "type": "Record",
    "attributes": {
      "email": {
        "type": "String"
      },
      "email_verified": {
        "type": "Boolean"
      },
      "name": {
        "type": "String",
        "required": true
      }
    }
  }
}
```

```
    },
    "phone_number": {
      "type": "String"
    },
    "phone_number_verified": {
      "type": "Boolean"
    }
  }
}
```

Untuk contoh kebijakan yang akan memvalidasi skema ini, lihat. [Mencerminkan atribut token ID OIDC](#)

Memetakan token akses

Izin Terverifikasi memproses klaim token akses selain klaim grup sebagai atribut tindakan, atau atribut konteks. Seiring dengan keanggotaan grup, token akses dari IDP Anda mungkin berisi informasi tentang akses API. Token akses berguna dalam model otorisasi yang menggunakan kontrol akses berbasis peran (RBAC). Model otorisasi yang mengandalkan klaim token akses selain keanggotaan grup memerlukan upaya tambahan dalam konfigurasi skema.

Sebagian besar token akses dari penyedia OIDC eksternal sejajar erat dengan Amazon Cognito token akses. Token akses OIDC dipetakan ke objek konteks saat diteruskan ke Izin Terverifikasi. Atribut token akses dapat direferensikan menggunakan `context.token.attribute_name`. Contoh token akses OIDC berikut mencakup contoh klaim dasar.

```
{
  "sub": "91eb4550-9091-708c-a7a6-9758ef8b6b1e",
  "groups": [
    "Store-Owner-Role",
    "Customer"
  ],
  "iss": "https://auth.example.com",
  "client_id": "1example23456789",
  "aud": "https://myapplication.example.com"
  "scope": "MyAPI-Read",
  "exp": 1688096566,
  "iat": 1688092966,
  "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN2222222",
  "username": "alice"
}
```

Contoh berikut menunjukkan cara mencerminkan atribut dari token akses contoh dalam skema Izin Terverifikasi Anda. Untuk informasi selengkapnya tentang mengedit skema Anda, lihat [Mengedit skema toko kebijakan](#).

```
{
  "MyApplication": {
    "actions": {
      "Read": {
        "appliesTo": {
          "context": {
            "type": "ReusedContext"
          },
          "resourceTypes": [
            "Application"
          ],
          "principalTypes": [
            "User"
          ]
        }
      }
    },
    ...
    ...
    "commonTypes": {
      "ReusedContext": {
        "attributes": {
          "token": {
            "type": "Record",
            "attributes": {
              "scope": {
                "type": "Set",
                "element": {
                  "type": "String"
                }
              },
              "client_id": {
                "type": "String"
              }
            }
          }
        }
      },
      "type": "Record"
    }
  }
}
```

```
    }  
  }  
}
```

Untuk contoh kebijakan yang akan memvalidasi skema ini, lihat. [Mencerminkan atribut token akses OIDC](#)

Hal-hal yang perlu diketahui tentang pemetaan skema

Pemetaan atribut berbeda antara jenis token

[Dalam otorisasi token akses, Izin Terverifikasi memetakan klaim ke konteks.](#) Dalam otorisasi token ID, Izin Terverifikasi memetakan klaim ke atribut utama. Untuk penyimpanan kebijakan yang Anda buat di konsol Izin Terverifikasi, hanya penyimpanan kebijakan kosong dan sampel yang tidak memiliki sumber identitas dan mengharuskan Anda mengisi skema Anda dengan atribut kumpulan pengguna untuk otorisasi token ID. Otorisasi token akses didasarkan pada kontrol akses berbasis peran (RBAC) dengan klaim keanggotaan grup dan tidak secara otomatis memetakan klaim lain ke skema penyimpanan kebijakan.

Atribut sumber identitas tidak diperlukan

Saat Anda membuat sumber identitas di konsol Izin Terverifikasi, tidak ada atribut yang ditandai sebagai wajib. Ini mencegah klaim yang hilang menyebabkan kesalahan validasi dalam permintaan otorisasi. Anda dapat mengatur atribut ke required sesuai kebutuhan, tetapi atribut tersebut harus ada di semua permintaan otorisasi.

RBAC tidak memerlukan atribut dalam skema

Skema untuk sumber identitas bergantung pada asosiasi entitas yang Anda buat saat menambahkan sumber identitas. Sumber identitas memetakan satu klaim ke jenis entitas pengguna, dan satu klaim ke jenis entitas grup. Pemetaan entitas ini adalah inti dari konfigurasi sumber identitas. Dengan informasi minimum ini, Anda dapat menulis kebijakan yang melakukan tindakan otorisasi untuk pengguna tertentu dan grup tertentu yang mungkin menjadi anggota pengguna, dalam model kontrol akses berbasis peran (RBAC). Penambahan klaim token ke skema memperluas cakupan otorisasi toko kebijakan Anda. Atribut pengguna dari token ID memiliki informasi tentang pengguna yang dapat berkontribusi pada otorisasi kontrol akses berbasis atribut (ABAC). Atribut konteks dari token akses memiliki informasi seperti cakupan OAuth 2.0 yang dapat menyumbangkan informasi kontrol akses tambahan dari penyedia Anda, tetapi memerlukan modifikasi skema tambahan.

Opsi Pengaturan dengan API Gateway dan penyedia identitas serta Penyiapan terpandu di konsol Izin Terverifikasi menetapkan klaim token ID ke skema. Ini tidak berlaku untuk klaim token akses.

[Untuk menambahkan klaim token akses non-grup ke skema Anda, Anda harus mengedit skema Anda dalam mode JSON dan menambahkan atribut `CommonTypes`.](#) Untuk informasi selengkapnya, lihat [Memetakan token akses](#).

Klaim grup OIDC mendukung berbagai format

Saat menambahkan penyedia OIDC, Anda dapat memilih nama klaim grup di ID atau token akses yang ingin dipetakan ke keanggotaan grup pengguna di toko kebijakan Anda. Izin terverifikasi mengenali klaim grup dalam format berikut:

1. String tanpa spasi: `"groups": "MyGroup"`
2. Daftar yang dibatasi ruang: `"groups": "MyGroup1 MyGroup2 MyGroup3"` Setiap string adalah grup.
3. Daftar JSON (dibatasi koma): `"groups": ["MyGroup1", "MyGroup2", "MyGroup3"]`

Note

Izin Terverifikasi menafsirkan setiap string dalam klaim grup yang dipisahkan spasi sebagai grup terpisah. Untuk menafsirkan nama grup dengan karakter spasi sebagai grup tunggal, ganti atau hapus spasi dalam klaim. Misalnya, format grup bernama `My Group` sebagai `MyGroup`.

Pilih jenis token

Cara penyimpanan kebijakan Anda bekerja dengan sumber identitas Anda bergantung pada keputusan kunci dalam konfigurasi sumber identitas: apakah Anda akan memproses ID atau token akses. Dengan penyedia OIDC, Anda harus memilih jenis token saat menambahkan sumber identitas. Anda dapat memilih ID atau token akses, dan pilihan Anda mengecualikan jenis token yang tidak dipilih untuk diproses di toko kebijakan Anda. Terutama jika Anda ingin mendapatkan keuntungan dari pemetaan otomatis klaim token ID ke atribut di konsol Izin Terverifikasi, putuskan lebih awal tentang jenis token yang ingin Anda proses sebelum Anda membuat sumber identitas Anda. Mengubah jenis token membutuhkan upaya yang signifikan untuk memfaktorkan ulang kebijakan dan skema Anda. Topik berikut menjelaskan penggunaan ID dan token akses dengan toko kebijakan.

Parser cedar membutuhkan tanda kurung untuk beberapa karakter

Kebijakan biasanya referensi atribut skema dalam format seperti `principal.username`. Dalam kasus sebagian besar karakter non-alfanumerik seperti `:`, `.`, atau `/` yang mungkin muncul di nama klaim token, Izin Terverifikasi tidak dapat mengurai nilai kondisi seperti atau `principal.cognito:username context.ip-address` Anda harus memformat kondisi ini dengan notasi braket dalam format `principal["cognito:username"]` atau `context["ip-address"]`, masing-masing. Karakter garis bawah `_` adalah karakter yang valid dalam nama klaim, dan satu-satunya pengecualian non-alfanumerik untuk persyaratan ini.

Contoh sebagian skema untuk atribut utama dari jenis ini terlihat seperti berikut:

```
"User": {
  "shape": {
    "type": "Record",
    "attributes": {
      "cognito:username": {
        "type": "String",
        "required": true
      },
      "custom:employmentStoreCode": {
        "type": "String",
        "required": true,
      },
      "email": {
        "type": "String",
        "required": false
      }
    }
  }
}
```

Contoh sebagian skema untuk atribut konteks jenis ini terlihat seperti berikut:

```
"GetOrder": {
  "memberOf": [],
  "appliesTo": {
    "resourceTypes": [
      "Order"
    ],
  },
  "context": {
    "type": "Record",
    "attributes": {
      "ip-address": {
```

```
        "required": false,  
        "type": "String"  
    }  
},  
    "principalTypes": [  
        "User"  
    ]  
}  
}
```

Untuk contoh kebijakan yang akan memvalidasi skema ini, lihat. [Menggunakan notasi braket untuk referensi atribut token](#)

Validasi klien dan audiens untuk penyedia OIDC

Saat Anda menambahkan sumber identitas ke penyimpanan kebijakan, Izin Terverifikasi memiliki opsi konfigurasi yang memverifikasi bahwa ID dan token akses digunakan sebagaimana dimaksud. Validasi ini terjadi dalam pemrosesan permintaan `IsAuthorizedWithToken` dan `BatchIsAuthorizedWithToken` API. Perilaku berbeda antara ID dan token akses, dan antara Amazon Cognito dan sumber identitas OIDC. Dengan penyedia kumpulan pengguna Amazon Cognito, Izin Terverifikasi dapat memvalidasi ID klien di ID dan token akses. Dengan penyedia OIDC, Izin Terverifikasi dapat memvalidasi ID klien dalam token ID, dan audiens dalam token akses.

ID klien adalah pengidentifikasi yang terkait dengan instance penyedia identitas yang digunakan aplikasi Anda, misalnya `1example23456789`. Audiens adalah jalur URL yang terkait dengan pihak yang mengandalkan, atau tujuan, dari token akses, misalnya `https://mytoken.example.com`. Saat menggunakan token akses, aud klaim selalu dikaitkan dengan audiens.

Token ID OIDC memiliki aud klaim yang berisi klien IDs, seperti. `1example23456789`

Token Akses OIDC memiliki aud klaim yang berisi URL pemirsa untuk token, seperti `https://myapplication.example.com`, dan `client_id` klaim yang berisi klien IDs, seperti. `1example23456789`

Saat menyiapkan penyimpanan kebijakan, masukkan satu atau beberapa nilai untuk validasi Audiens yang digunakan oleh kebijakan Anda untuk memvalidasi pemirsa token.

- Token ID — Izin Terverifikasi memvalidasi ID klien dengan memeriksa bahwa setidaknya satu anggota klien IDs dalam aud klaim cocok dengan nilai validasi audiens.

- Token akses — Izin Terverifikasi memvalidasi audiens dengan memeriksa apakah URL dalam aud klaim cocok dengan nilai validasi audiens. Jika tidak ada aud klaim, audiens dapat divalidasi menggunakan `client_id` klaim `cid` atau. Periksa dengan penyedia identitas Anda untuk klaim dan format audiens yang benar.

Otorisasi sisi klien untuk JWTs

Anda mungkin ingin memproses token web JSON di aplikasi Anda dan meneruskan klaimnya ke Izin Terverifikasi tanpa menggunakan sumber identitas toko kebijakan. Anda dapat mengekstrak atribut entitas Anda dari JSON Web Token (JWT) dan menguraikannya menjadi Izin Terverifikasi.

Contoh ini menunjukkan bagaimana Anda dapat memanggil Izin Terverifikasi dari aplikasi menggunakan JWT.¹

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

    let principalEntity = {
        entityType: "PhotoFlash::User", // the application needs to fill in the
relevant user type
        entityId: payload["sub"], // the application need to use the claim that
represents the user-id
    };
    let resourceEntity = {
        entityType: "PhotoFlash::Photo", //the application needs to fill in the
relevant resource type
        entityId: "jane_photo_123.jpg", // the application needs to fill in the
relevant resource id
    };
    let action = {
        actionType: "PhotoFlash::Action", //the application needs to fill in the
relevant action id
        actionId: "GetPhoto", //the application needs to fill in the relevant action
type
    };
    let entities = {
        entityList: [],
    };
    entities.entityList.push(...getUserEntitiesFromToken(payload));
    let policyStoreId = "PSEXAMPLEabcdefg111111"; // set your own policy store id
```

```
const authResult = await client
  .isAuthorized({
    policyStoreId: policyStoreId,
    principal: principalEntity,
    resource: resourceEntity,
    action: action,
    entities,
  })
  .promise();

return authResult;
}

function getUserEntitiesFromToken(payload) {
  let attributes = {};
  let claimsNotPassedInEntities = ['aud', 'sub', 'exp', 'jti', 'iss'];
  Object.entries(payload).forEach(([key, value]) => {
    if (claimsNotPassedInEntities.includes(key)) {
      return;
    }
    if (Array.isArray(value)) {
      var attributeItem = [];
      value.forEach((item) => {
        attributeItem.push({
          string: item,
        });
      });
      attributes[key] = {
        set: attributeItem,
      };
    } else if (typeof value === 'string') {
      attributes[key] = {
        string: value,
      }
    } else if (typeof value === 'bigint' || typeof value === 'number') {
      attributes[key] = {
        long: value,
      }
    } else if (typeof value === 'boolean') {
      attributes[key] = {
        boolean: value,
      }
    }
  })
}
```

```
});

let entityItem = {
  attributes: attributes,
  identifier: {
    entityType: "PhotoFlash::User",
    entityId: payload["sub"], // the application needs to use the claim that
    represents the user-id
  }
};
return [entityItem];
}
```

¹ Contoh kode ini menggunakan [aws-jwt-verify](#) pustaka untuk memverifikasi JWTs ditandatangani oleh IdPs OIDC-kompatibel.

Integrasi untuk Izin Terverifikasi Amazon

Integrasi Izin Terverifikasi Amazon membantu Anda menerapkan otorisasi berbutir halus dalam aplikasi sambil meminimalkan kode dan mengikuti praktik terbaik khusus kerangka kerja. Integrasi ini menyediakan komponen dan utilitas middleware yang menghubungkan aplikasi Anda dengan Izin Terverifikasi dengan mulus.

Dengan integrasi, Anda dapat:

- Menerapkan otorisasi dalam hitungan menit
- Ikuti pola dan konvensi khusus kerangka kerja
- Kurangi overhead perawatan
- Meminimalkan potensi kesalahan implementasi keamanan
- Fokus pada logika bisnis daripada kode otorisasi

Ketika ditambahkan ke aplikasi Anda, integrasi melakukan hal berikut:

1. Mencegat permintaan masuk melalui middleware khusus kerangka kerja
2. Ekstrak konteks otorisasi yang relevan dari permintaan
3. Tentukan keputusan otorisasi menggunakan Izin Terverifikasi
4. Menegakkan kontrol akses berdasarkan hasil otorisasi

Izin Terverifikasi saat ini mendukung kerangka kerja berikut:

- [Express.js untuk aplikasi Node.js](#)

Mengintegrasikan Express dengan Izin Terverifikasi Amazon

Integrasi Verified Permissions Express menyediakan pendekatan berbasis middleware untuk menerapkan otorisasi dalam aplikasi Express.js Anda. Dengan integrasi ini, Anda dapat melindungi titik akhir API menggunakan kebijakan otorisasi berbutir halus tanpa mengubah penanganan rute yang ada. Integrasi menangani pemeriksaan otorisasi secara otomatis dengan mencegat permintaan, mengevaluasinya terhadap kebijakan yang Anda tetapkan, dan memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses sumber daya yang dilindungi.

Topik ini memandu Anda melalui pengaturan integrasi Express, mulai dari membuat toko kebijakan hingga menerapkan dan menguji middleware otorisasi. Dengan mengikuti langkah-langkah ini, Anda dapat menambahkan kontrol otorisasi yang kuat ke aplikasi Express Anda dengan sedikit perubahan kode.

Repo berikut GitHub direferensikan di seluruh topik ini:

- [cedar-policy/ authorization-for-expressjs](#) - Middleware otorisasi Cedar untuk Express.js
- [verifiedpermissions/ authorization-clients-js](#) - Klien otorisasi Izin Terverifikasi untuk JavaScript
- [verifiedpermissions/examples/express-petstore](#) - Contoh implementasi menggunakan middleware Express.js

Prasyarat

Sebelum Anda menerapkan integrasi Express, pastikan Anda memiliki:

- [AWS Akun](#) dengan akses ke Izin Terverifikasi
- [Node.js](#) dan [npm diinstal](#)
- Aplikasi [Express.js](#)
- Penyedia identitas OpenID Connect (OIDC) (seperti [Amazon Cognito](#))
- [AWS CLI](#) di konfigurasi dengan izin yang sesuai

Menyiapkan integrasi

Langkah 1: Buat toko kebijakan

Buat toko kebijakan menggunakan AWS CLI:

```
aws verifiedpermissions create-policy-store --validation-settings "mode=STRICT"
```

Note

Simpan ID penyimpanan kebijakan yang dikembalikan dalam respons untuk digunakan pada langkah selanjutnya.

Langkah 2: Instal dependensi

Instal paket yang diperlukan di aplikasi Express Anda:

```
npm i --save @verifiedpermissions/authorization-clients-js
npm i --save @cedar-policy/authorization-for-expressjs
```

Mengkonfigurasi otorisasi

Langkah 1: Buat dan unggah skema Cedar

Skema mendefinisikan model otorisasi untuk aplikasi, termasuk jenis entitas dalam aplikasi dan tindakan yang diizinkan untuk dilakukan pengguna. Kami merekomendasikan mendefinisikan [namespace](#) untuk skema Anda. Dalam contoh ini, kami menggunakan `YourNamespace`. Anda melampirkan skema Anda ke penyimpanan kebijakan Izin Terverifikasi, dan ketika kebijakan ditambahkan atau diubah, layanan secara otomatis memvalidasi kebijakan terhadap skema.

`@cedar-policy/authorization-for-expressjs` Paket ini dapat menganalisis [spesifikasi OpenAPI](#) aplikasi Anda dan menghasilkan skema Cedar. Secara khusus, objek jalur diperlukan dalam spesifikasi Anda.

Jika Anda tidak memiliki spesifikasi OpenAPI, Anda dapat mengikuti instruksi cepat [express-openapi-generator](#) paket untuk menghasilkan spesifikasi OpenAPI.

Buat skema dari spesifikasi OpenAPI Anda:

```
npx @cedar-policy/authorization-for-expressjs generate-schema --api-spec schemas/openapi.json --namespace YourNamespace --mapping-type SimpleRest
```

Selanjutnya, format skema Cedar untuk digunakan dengan AWS CLI Untuk informasi selengkapnya tentang format spesifik yang diperlukan, lihat [Skema toko kebijakan](#). Jika Anda memerlukan bantuan untuk memformat skema, ada skrip yang dipanggil `prepare-cedar-schema.sh` di repo [GitHubverifiedpermissions/examples](#). Berikut ini adalah contoh panggilan ke skrip yang mengeluarkan skema diformat Izin Terverifikasi dalam file `v2.cedarschema.forAVP.json`

```
./scripts/prepare-cedar-schema.sh v2.cedarschema.json v2.cedarschema.forAVP.json
```

Unggah skema yang diformat ke toko kebijakan Anda, ganti `policy-store-id` dengan ID toko kebijakan Anda:

```
aws verifiedpermissions put-schema \  
  --definition file://v2.cedarschema.forAVP.json \  
  --policy-store-id policy-store-id
```

Langkah 2: Buat kebijakan otorisasi

Jika tidak ada kebijakan yang dikonfigurasi, Cedar menolak semua permintaan otorisasi. Integrasi kerangka kerja Express membantu mem-bootstrap proses ini dengan menghasilkan contoh kebijakan berdasarkan skema yang dihasilkan sebelumnya.

Saat menggunakan integrasi ini dalam aplikasi produksi Anda, sebaiknya buat kebijakan baru menggunakan alat infrastruktur sebagai kode (iAAC). Untuk informasi selengkapnya, lihat [Bekerja dengan AWS CloudFormation](#).

Hasilkan contoh kebijakan Cedar:

```
npx @cedar-policy/authorization-for-expressjs generate-policies --schema  
  v2.cedarschema.json
```

Ini akan menghasilkan kebijakan sampel di /policies direktori. Anda kemudian dapat menyesuaikan kebijakan ini berdasarkan kasus penggunaan Anda. Contoh:

```
// Defines permitted administrator user group actions  
permit (  
  principal in YourNamespace::UserGroup::"<userPoolId>|administrator",  
  action,  
  resource  
);  
  
// Defines permitted employee user group actions  
permit (  
  principal in YourNamespace::UserGroup::"<userPoolId>|employee",  
  action in  
    [YourNamespace::Action::"GET /resources",  
     YourNamespace::Action::"POST /resources",  
     YourNamespace::Action::"GET /resources/{resourceId}",  
     YourNamespace::Action::"PUT /resources/{resourceId}"],  
  resource  
);
```

Format kebijakan untuk digunakan dengan AWS CLI. Untuk informasi lebih lanjut tentang format yang diperlukan, lihat [create-policy dalam referensi](#). AWS CLI Jika Anda memerlukan bantuan untuk memformat kebijakan, ada skrip yang dipanggil `convert_cedar_policies.sh` di repo [verifiedpermissions/examples](#) GitHub. Berikut ini adalah panggilan ke skrip itu:

```
./scripts/convert_cedar_policies.sh
```

Unggah kebijakan yang diformat ke Izin Terverifikasi, ganti `policy_1.json` dengan jalur dan nama file kebijakan Anda dan `policy-store-id` dengan ID penyimpanan kebijakan Anda:

```
aws verifiedpermissions create-policy \  
  --definition file://policies/json/policy_1.json \  
  --policy-store-id policy-store-id
```

Langkah 3: Hubungkan penyedia identitas

Secara default, middleware otorisasi Izin Terverifikasi membaca Token Web JSON (JWT) yang disediakan dalam header otorisasi permintaan API untuk mendapatkan informasi pengguna. Izin Terverifikasi dapat memvalidasi token selain melakukan evaluasi kebijakan otorisasi.

Buat file konfigurasi sumber identitas bernama `identity-source-configuration.txt` yang terlihat seperti berikut dengan file Anda `userPoolArn` dan `clientId`:

```
{  
  "cognitoUserPoolConfiguration": {  
    "userPoolArn": "arn:aws:cognito-idp:region:account:userpool/pool-id",  
    "clientIds": ["client-id"],  
    "groupConfiguration": {  
      "groupEntityType": "YourNamespace::UserGroup"  
    }  
  }  
}
```

Buat sumber identitas dengan menjalankan AWS CLI perintah berikut, ganti `policy-store-id` dengan ID toko kebijakan Anda:

```
aws verifiedpermissions create-identity-source \  
  --configuration file://identity-source-configuration.txt \  
  --policy-store-id policy-store-id \  
  --principal-entity-type YourNamespace::User
```

Menerapkan middleware otorisasi

Perbarui aplikasi Express Anda untuk menyertakan middleware otorisasi. Dalam contoh ini kami menggunakan token identitas, tetapi Anda juga dapat menggunakan token akses. Untuk informasi lebih lanjut, lihat [authorization-for-expressjs](#) di GitHub.

```
const { ExpressAuthorizationMiddleware } = require('@cedar-policy/authorization-for-expressjs');

const { AVPAuthorizationEngine } = require('@verifiedpermissions/authorization-clients');

const avpAuthorizationEngine = new AVPAuthorizationEngine({
  policyStoreId: 'policy-store-id',
  callType: 'identityToken'
});

const expressAuthorization = new ExpressAuthorizationMiddleware({
  schema: {
    type: 'jsonString',
    schema: fs.readFileSync(path.join(__dirname, '../v4.cedarschema.json'),
      'utf8'),
  },
  authorizationEngine: avpAuthorizationEngine,
  principalConfiguration: { type: 'identityToken' },
  skippedEndpoints: [],
  logger: {
    debug: (s) => console.log(s),
    log: (s) => console.log(s),
  }
});

// Add the middleware to your Express application
app.use(expressAuthorization.middleware);
```

Menguji integrasi

Anda dapat menguji implementasi otorisasi Anda dengan membuat permintaan ke titik akhir API Anda dengan token pengguna yang berbeda. Middleware otorisasi akan secara otomatis mengevaluasi setiap permintaan terhadap kebijakan yang Anda tetapkan.

Misalnya, jika Anda telah menyiapkan grup pengguna yang berbeda dengan izin yang berbeda:

- Administrator: Akses penuh ke semua sumber daya dan fungsi manajemen
- Karyawan: Dapat melihat, membuat, dan memperbarui sumber daya
- Pelanggan: Hanya dapat melihat sumber daya

Anda dapat memvalidasi bahwa kebijakan izin berfungsi seperti yang diharapkan dengan masuk ke pengguna yang berbeda dan mencoba berbagai operasi. Di terminal untuk aplikasi Express, Anda dapat melihat output log yang memberikan rincian tambahan tentang keputusan otorisasi.

Pemecahan masalah

Jika Anda mengalami kegagalan otorisasi, coba yang berikut ini:

- Verifikasi ID toko kebijakan Anda benar
- Pastikan sumber identitas Anda dikonfigurasi dengan benar
- Periksa apakah kebijakan Anda telah diformat dengan benar
- Validasi bahwa token JWT Anda valid

Langkah selanjutnya

Setelah menerapkan integrasi dasar, pertimbangkan:

- Menerapkan pemetaan khusus untuk skenario otorisasi tertentu
- Menyiapkan pemantauan dan pencatatan untuk keputusan otorisasi
- Membuat kebijakan tambahan untuk peran pengguna yang berbeda

Menerapkan otorisasi di Izin Terverifikasi Amazon

Setelah membuat toko kebijakan, kebijakan, templat, skema, dan model otorisasi, Anda siap untuk mulai mengotorisasi permintaan menggunakan Izin Terverifikasi Amazon. Untuk menerapkan otorisasi Izin Terverifikasi, Anda harus menggabungkan konfigurasi kebijakan otorisasi AWS dengan integrasi dalam aplikasi. Untuk mengintegrasikan Izin Terverifikasi dengan aplikasi Anda, tambahkan AWS SDK dan terapkan metode yang memanggil API Izin Terverifikasi dan buat keputusan otorisasi terhadap penyimpanan kebijakan Anda.

Otorisasi dengan Izin Terverifikasi berguna untuk izin UX dan izin API di aplikasi Anda.

Izin UX

Kontrol akses pengguna ke UX aplikasi Anda. Anda dapat mengizinkan pengguna untuk melihat hanya bentuk, tombol, grafik, dan sumber daya lain yang tepat yang perlu mereka akses. Misalnya, saat pengguna masuk, Anda mungkin ingin menentukan apakah tombol “Transfer dana” terlihat di akun mereka. Anda juga dapat mengontrol tindakan yang dapat dilakukan pengguna. Misalnya, di aplikasi perbankan yang sama, Anda mungkin ingin menentukan apakah pengguna Anda diizinkan untuk mengubah kategori transaksi.

Izin API:

Kontrol akses pengguna ke data. Aplikasi sering menjadi bagian dari sistem terdistribusi dan membawa informasi dari eksternal APIs. Dalam contoh aplikasi perbankan di mana Izin Terverifikasi telah mengizinkan tampilan tombol “Transfer dana”, keputusan otorisasi yang lebih kompleks harus dibuat saat pengguna Anda memulai transfer. Izin Terverifikasi dapat mengotorisasi permintaan API yang mencantumkan akun tujuan yang merupakan target transfer yang memenuhi syarat, dan kemudian permintaan untuk mendorong transfer ke akun lain.

Contoh yang menggambarkan konten ini berasal dari [toko kebijakan sampel](#). Untuk mengikuti, buat penyimpanan kebijakan DigitalPetStoresampel di lingkungan pengujian Anda.

Untuk contoh aplikasi ujung ke ujung yang mengimplementasikan izin UX menggunakan otorisasi batch, lihat Menggunakan Izin [Terverifikasi Amazon untuk otorisasi berbutir halus](#) pada skala besar di Blog Keamanan.AWS

Topik

- [Operasi API yang tersedia untuk otorisasi](#)

- [Menguji model otorisasi Anda](#)
- [Mengintegrasikan model otorisasi Anda dengan aplikasi](#)

Operasi API yang tersedia untuk otorisasi

API Izin Terverifikasi memiliki operasi otorisasi berikut.

[IsAuthorized](#)

Operasi `IsAuthorized` API adalah titik masuk ke permintaan otorisasi dengan Izin Terverifikasi. Anda harus mengirimkan elemen pokok, tindakan, sumber daya, konteks, dan entitas. Izin Terverifikasi mengevaluasi permintaan Anda terhadap semua kebijakan di toko kebijakan yang diminta yang berlaku untuk entitas dalam permintaan.

[IsAuthorizedWithToken](#)

`IsAuthorizedWithToken` Operasi menghasilkan permintaan otorisasi dari data pengguna di token web JSON (JWTs). Izin Terverifikasi berfungsi langsung dengan penyedia OIDC seperti Amazon Cognito sebagai sumber identitas di toko kebijakan Anda. Izin Terverifikasi mengisi semua atribut ke prinsipal dalam permintaan Anda dari klaim di ID pengguna atau token akses. Anda dapat mengotorisasi tindakan dan sumber daya dari atribut pengguna atau keanggotaan grup dalam sumber identitas.

Anda tidak dapat menyertakan informasi tentang grup atau tipe utama pengguna dalam `IsAuthorizedWithToken` permintaan. Anda harus mengisi semua data utama ke JWT yang Anda berikan.

[BatchIsAuthorized](#)

`BatchIsAuthorized` Operasi memproses beberapa keputusan otorisasi untuk satu prinsipal atau sumber daya dalam satu permintaan API. Operasi ini mengelompokkan permintaan ke dalam satu operasi batch yang meminimalkan [penggunaan kuota](#) dan mengembalikan keputusan otorisasi untuk masing-masing hingga 30 tindakan bersarang yang kompleks. Dengan otorisasi batch untuk satu sumber daya, Anda dapat memfilter tindakan yang dapat dilakukan pengguna pada sumber daya. Dengan otorisasi batch untuk satu prinsipal, Anda dapat memfilter sumber daya yang dapat diambil tindakan pengguna.

[BatchIsAuthorizedWithToken](#)

`BatchIsAuthorizedWithToken` Operasi memproses beberapa keputusan otorisasi untuk satu prinsipal dalam satu permintaan API. Prinsipal disediakan oleh sumber identitas toko polis

Anda dalam ID atau token akses. Operasi ini mengelompokkan permintaan ke dalam satu operasi batch yang meminimalkan [penggunaan kuota](#) dan mengembalikan keputusan otorisasi untuk masing-masing hingga 30 permintaan untuk tindakan dan sumber daya. Dalam kebijakan Anda, Anda dapat mengotorisasi akses mereka dari atribut atau keanggotaan grup mereka di direktori pengguna.

Seperti halnya `IsAuthorizedWithToken`, Anda tidak dapat menyertakan informasi tentang tipe utama grup atau pengguna dalam `BatchIsAuthorizedWithToken` permintaan. Anda harus mengisi semua data utama ke JWT yang Anda berikan.

Menguji model otorisasi Anda

Untuk memahami pengaruh keputusan otorisasi Izin Terverifikasi Amazon saat menerapkan aplikasi, Anda dapat mengevaluasi kebijakan saat mengembangkannya dengan [Menggunakan bangku uji Izin Terverifikasi Amazon](#) dan dengan permintaan HTTPS REST API ke Izin Terverifikasi. Bangku tes adalah alat Konsol Manajemen AWS untuk mengevaluasi permintaan otorisasi dan tanggapan di toko kebijakan Anda.

API REST Izin Terverifikasi adalah langkah selanjutnya dalam pengembangan Anda saat Anda beralih dari pemahaman konseptual ke desain aplikasi. [API Izin Terverifikasi menerima permintaan otorisasi dengan `IsAuthorized`, `IsAuthorizedWithToken`, dan `BatchIsAuthorized` sebagai permintaan AWS API yang ditandatangani ke titik akhir layanan Regional](#). Untuk menguji model otorisasi, Anda dapat membuat permintaan dengan klien API apa pun dan memverifikasi bahwa kebijakan Anda mengembalikan keputusan otorisasi seperti yang diharapkan.

Misalnya, Anda dapat menguji `IsAuthorized` di toko kebijakan sampel dengan prosedur berikut.

Test bench

1. Buka konsol Izin Terverifikasi di konsol [Izin Terverifikasi](#). Buat toko kebijakan dari toko kebijakan Sample dengan nama `DigitalPetStore`.
2. Pilih bangku tes di toko kebijakan baru Anda.
3. Isi permintaan bangku pengujian Anda dari [IsAuthorized](#) referensi API Izin Terverifikasi. Rincian berikut mereplikasi kondisi dalam Contoh 4 yang mereferensikan `DigitalPetStore` sampel.
 - a. Tetapkan Alice sebagai kepala sekolah. Untuk Principal mengambil tindakan, pilih `DigitalPetStore::User` dan masukkan `Alice`.

- b. Tetapkan peran Alice sebagai pelanggan. Pilih Tambahkan induk, pilih `DigitalPetStore::Role`, dan masukkan Pelanggan.
 - c. Atur sumber daya sebagai urutan "1234." Untuk Sumber Daya tempat kepala sekolah bertindak, pilih `DigitalPetStore::Order` dan masukkan 1234.
 - d. Sumber `DigitalPetStore::Order` daya membutuhkan `owner` atribut. Tetapkan Alice sebagai pemilik pesanan. Pilih `DigitalPetStore::User` dan masukkan Alice
 - e. Alice meminta untuk melihat pesanan. Untuk tindakan yang diambil kepala sekolah, pilih `DigitalPetStore::Action::"GetOrder"`.
4. Pilih Jalankan permintaan otorisasi. Di toko kebijakan yang tidak dimodifikasi, permintaan ini menghasilkan `ALLOW` keputusan. Perhatikan kebijakan Puas yang mengembalikan keputusan.
 5. Pilih Kebijakan dari bilah navigasi kiri. Tinjau kebijakan statis dengan deskripsi Peran Pelanggan - Dapatkan Pesanan.
 6. Perhatikan bahwa Izin Terverifikasi mengizinkan permintaan karena prinsipal berada dalam peran pelanggan dan merupakan pemilik sumber daya.

REST API

1. Buka konsol Izin Terverifikasi di konsol [Izin Terverifikasi](#). Buat toko kebijakan dari toko kebijakan Sample dengan nama `DigitalPetStore`.
2. Perhatikan ID toko Kebijakan toko kebijakan baru Anda.
3. Dari [IsAuthorized](#) referensi API Izin Terverifikasi, salin badan permintaan Contoh 4 yang mereferensikan `DigitalPetStoresampel`.
4. Buka klien API Anda dan buat permintaan ke titik akhir layanan Regional untuk toko kebijakan Anda. [Isi header seperti yang ditunjukkan pada contoh.](#)
5. Tempel di badan permintaan sampel dan ubah nilainya `policyStoreId` ke ID penyimpanan kebijakan yang Anda catat sebelumnya.
6. Kirim permintaan dan tinjau hasilnya. Di penyimpanan `DigitalPetStore` kebijakan default, permintaan ini mengembalikan `ALLOW` keputusan.

Anda dapat membuat perubahan pada kebijakan, skema, dan permintaan di lingkungan pengujian Anda untuk mengubah hasil dan menghasilkan keputusan yang lebih kompleks.

1. Ubah permintaan dengan cara yang mengubah keputusan dari Izin Terverifikasi. Misalnya, ubah peran Alice menjadi `Employee` atau ubah `owner` atribut urutan 1234 menjadi Bob.
2. Ubah kebijakan dengan cara yang memengaruhi keputusan otorisasi. Misalnya, ubah kebijakan dengan deskripsi Peran Pelanggan - Dapatkan Pesanan untuk menghapus kondisi bahwa `User` harus menjadi pemilik `Resource` dan memodifikasi permintaan sehingga Bob ingin melihat pesanan.
3. Ubah skema untuk memungkinkan kebijakan membuat keputusan yang lebih kompleks. Perbarui entitas permintaan sehingga Alice dapat memenuhi persyaratan baru. Misalnya, edit skema `User` untuk memungkinkan menjadi anggota `ActiveUsers` atau `InactiveUsers`. Perbarui kebijakan sehingga hanya pengguna aktif yang dapat melihat pesanan mereka sendiri. Perbarui entitas permintaan sehingga Alice adalah pengguna aktif atau tidak aktif.

Mengintegrasikan model otorisasi Anda dengan aplikasi

Untuk menerapkan Izin Terverifikasi Amazon di aplikasi Anda, Anda harus menentukan kebijakan dan skema yang ingin diterapkan oleh aplikasi Anda. Dengan model otorisasi Anda di tempat dan diuji, langkah Anda selanjutnya adalah mulai membuat permintaan API dari titik penegakan hukum. Untuk melakukan ini, Anda harus mengatur logika aplikasi untuk mengumpulkan data pengguna dan mengisinya ke permintaan otorisasi.

Cara aplikasi mengotorisasi permintaan dengan Izin Terverifikasi

1. Kumpulkan informasi tentang pengguna saat ini. Biasanya, detail pengguna disediakan dalam rincian sesi yang diautentikasi, seperti JWT atau cookie sesi web. Data pengguna ini mungkin berasal dari [sumber Amazon Cognito identitas](#) yang ditautkan ke penyimpanan kebijakan Anda atau dari penyedia [OpenID Connect \(OIDC\)](#) lain.
2. Kumpulkan informasi tentang sumber daya yang ingin diakses pengguna. Biasanya, aplikasi Anda akan menerima informasi tentang sumber daya saat pengguna membuat pilihan yang mengharuskan aplikasi Anda memuat aset baru.
3. Tentukan tindakan yang ingin diambil pengguna Anda.
4. Buat permintaan otorisasi ke Izin Terverifikasi dengan prinsipal, tindakan, sumber daya, dan entitas untuk upaya operasi pengguna Anda. Izin Terverifikasi mengevaluasi permintaan terhadap kebijakan di toko kebijakan Anda dan mengembalikan keputusan otorisasi.
5. Aplikasi Anda membaca respons izinkan atau penolakan dari Izin Terverifikasi dan memberlakukan keputusan atas permintaan pengguna.

Operasi API Izin Terverifikasi dibangun ke dalam AWS SDKs. Untuk menyertakan Izin Terverifikasi dalam aplikasi, integrasikan AWS SDK untuk bahasa pilihan Anda ke dalam paket aplikasi.

Untuk mempelajari lebih lanjut dan mengunduh AWS SDKs, lihat [Alat untuk Amazon Web Services](#).

Berikut ini adalah tautan ke dokumentasi untuk sumber daya Izin Terverifikasi di berbagai AWS SDKs sumber.

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto\)](#)
- [AWS SDK for Ruby](#)
- [AWS SDK for Rust](#)

AWS SDK for JavaScript Contoh berikut `IsAuthorized` berasal dari [Sederhanakan otorisasi berbutir halus dengan Izin Terverifikasi Amazon dan Amazon Cognito](#).

```
const authResult = await avp.isAuthorized({
  principal: 'User::"alice"',
  action: 'Action::"view"',
  resource: 'Photo::"VacationPhoto94.jpg"',
  // whenever our policy references attributes of the entity,
  // isAuthorized needs an entity argument that provides
  // those attributes
  entities: {
    entityList: [
      {
        "identifier": {
          "entityType": "User",
          "entityId": "alice"
        },
        "attributes": {
          "location": {
            "String": "USA"
          }
        }
      }
    ]
  }
});
```

```
    }  
  }  
] }  
}  
});
```

Lebih banyak sumber daya pengembang

- [Lokakarya Izin Terverifikasi Amazon](#)
- [Izin Terverifikasi Amazon - Sumber Daya](#)
- [Menerapkan penyedia kebijakan otorisasi khusus untuk aplikasi ASP.NET Core menggunakan Izin Terverifikasi Amazon](#)
- [Membangun layanan hak untuk aplikasi bisnis menggunakan Izin Terverifikasi Amazon](#)
- [Sederhanakan otorisasi halus dengan Izin Terverifikasi Amazon dan Amazon Cognito](#)

Keamanan di Izin Terverifikasi Amazon

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 Izin Terverifikasi Amazon, lihat [AWS Layanan dalam Lingkup berdasarkan 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 Izin Terverifikasi. Topik berikut menunjukkan cara mengonfigurasi Izin Terverifikasi untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Izin Terverifikasi.

Topik

- [Perlindungan data di Izin Terverifikasi Amazon](#)
- [Manajemen identitas dan akses untuk Izin Terverifikasi Amazon](#)
- [Validasi kepatuhan untuk Izin Terverifikasi Amazon](#)
- [Ketahanan dalam Izin Terverifikasi Amazon](#)

Perlindungan data di Izin Terverifikasi Amazon

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di Izin Terverifikasi Amazon. Sebagaimana diuraikan dalam model ini, AWS 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. Konten ini mencakup konfigurasi keamanan dan tugas manajemen untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

- 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 menyarankan Anda mengamankan data Anda dengan cara berikut:
 - Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
 - Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2.
 - Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail.
 - Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
 - Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu dalam menemukan dan mengamankan data sensitif yang disimpan. Amazon S3
 - Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [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 Izin Terverifikasi atau lainnya Layanan AWS 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. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.
- Nama tindakan Anda tidak boleh menyertakan informasi sensitif apa pun.
- Kami juga sangat menyarankan agar Anda selalu menggunakan pengidentifikasi unik, tidak dapat diubah, dan tidak dapat digunakan kembali untuk entitas Anda (sumber daya dan prinsipal). Di lingkungan pengujian, Anda dapat memilih untuk menggunakan pengidentifikasi entitas sederhana,

seperti `jane` atau `bob` untuk nama entitas bertipe `User`. Namun, dalam sistem produksi, sangat penting untuk alasan keamanan bahwa Anda menggunakan nilai unik yang tidak dapat digunakan kembali. Kami menyarankan Anda menggunakan nilai seperti pengidentifikasi unik universal (UUIDs). Misalnya, perhatikan pengguna `jane` yang meninggalkan perusahaan. Kemudian, Anda membiarkan orang lain menggunakan nama itu `jane`. Pengguna baru itu mendapatkan akses secara otomatis ke semua yang diberikan oleh kebijakan yang masih merujuk `User : : "jane"`. Izin Terverifikasi dan Cedar tidak dapat membedakan antara pengguna baru dan pengguna sebelumnya.

Panduan ini berlaku untuk pengidentifikasi utama dan sumber daya. Selalu gunakan pengidentifikasi yang dijamin unik dan tidak pernah digunakan kembali untuk memastikan bahwa Anda tidak memberikan akses secara tidak sengaja karena adanya pengenalan lama dalam kebijakan.

- Pastikan bahwa string yang Anda berikan untuk menentukan `Long` dan `Decimal` nilai berada dalam rentang yang valid dari setiap jenis. Selain itu, pastikan bahwa penggunaan operator aritmatika tidak menghasilkan nilai di luar rentang yang valid. Jika rentang terlampaui, operasi menghasilkan pengecualian luapan. Kebijakan yang mengakibatkan kesalahan diabaikan, artinya kebijakan Izin mungkin gagal mengizinkan akses secara tak terduga, atau kebijakan Larangan mungkin gagal memblokir akses secara tak terduga.

Enkripsi data

Izin Terverifikasi Amazon secara otomatis mengenkripsi semua data pelanggan seperti kebijakan dengan file. Kunci yang dikelola AWS Izin Terverifikasi Amazon juga memungkinkan pelanggan memanfaatkan kunci yang dikelola pelanggan untuk mengenkripsi data mereka.

Untuk informasi rinci tentang penggunaan kunci terkelola pelanggan untuk enkripsi, lihat [the section called "Kunci yang dikelola pelanggan"](#).

Mengenkripsi Sumber Daya di Izin Terverifikasi Amazon

Izin Terverifikasi Amazon menyediakan enkripsi secara default untuk melindungi data pelanggan sensitif saat istirahat menggunakan kunci enkripsi yang AWS dimiliki. Sebagai lapisan perlindungan tambahan, Izin Terverifikasi Amazon memungkinkan Anda mengenkripsi toko kebijakan menggunakan AWS Key Management Service (AWS KMS) kunci yang dikelola pelanggan (CMK).

Fungsionalitas ini memastikan perlindungan data sensitif melalui enkripsi saat istirahat, yang membantu Anda:

- Mengurangi beban operasional di ujung aplikasi Anda untuk melindungi data sensitif
- Pertahankan kontrol atas siapa yang dapat melihat detail kebijakan otorisasi Anda melalui kebijakan Anda sendiri AWS KMS kunci yang dikelola pelanggan
- Membangun aplikasi sensitif keamanan yang memenuhi kepatuhan enkripsi yang ketat dan persyaratan peraturan

Bagian berikut menjelaskan cara mengonfigurasi enkripsi untuk penyimpanan kebijakan baru dan mengelola kunci enkripsi Anda.

AWS KMS Jenis Kunci untuk Izin Terverifikasi Amazon

Izin Terverifikasi Amazon terintegrasi dengan AWS KMS untuk mengelola kunci enkripsi yang digunakan untuk data encrypting/decrypting pelanggan. Untuk mempelajari lebih lanjut tentang jenis dan status kunci, lihat [AWS Key Management Service konsep](#) di Panduan AWS KMS Pengembang. Saat membuat penyimpanan kebijakan baru, Anda dapat memilih dari jenis AWS KMS kunci berikut untuk mengenkripsi data Anda:

AWS Kunci yang Dimiliki

Jenis enkripsi default. Izin Terverifikasi Amazon memiliki kunci tanpa biaya tambahan kepada Anda dan mengenkripsi data sumber daya saat dibuat. Tidak ada konfigurasi tambahan yang diperlukan dalam kode atau aplikasi Anda ke data encrypt/decrypt Anda menggunakan kunci yang dimiliki oleh Izin Terverifikasi.

Kunci yang Dikelola Pelanggan

Anda membuat, memiliki, dan mengelola kunci di AWS akun Anda. Anda memiliki kendali penuh atas AWS KMS kunci. AWS KMS biaya berlaku untuk kunci yang dikelola pelanggan s. Untuk informasi lebih lanjut, lihat halaman [AWS KMS Harga](#). Untuk informasi selengkapnya tentang jenis kunci, lihat [Kunci yang dikelola pelanggan](#) di Panduan AWS KMS Pengembang.

Saat Anda menentukan enkripsi kunci yang dikelola pelanggan untuk sumber daya tingkat atas (yaitu penyimpanan kebijakan), Izin Terverifikasi mengenkripsi sumber daya, serta sumber daya turunannya, dengan kunci tersebut. Untuk mengenkripsi penyimpanan kebijakan menggunakan kunci yang dikelola pelanggan, Anda harus memberikan akses ke Izin Terverifikasi dalam kebijakan

utama Anda. Kebijakan utama adalah [kebijakan berbasis sumber daya](#) yang Anda lampirkan untuk mengontrol akses ke kunci yang dikelola pelanggan sana. Lihat [the section called “Mengotorisasi penggunaan AWS KMS kunci Anda untuk Izin Terverifikasi Amazon”](#) untuk detail selengkapnya.

Selain itu, untuk membuat penyimpanan kebijakan terenkripsi dengan kunci yang dikelola pelanggan, atau untuk membuat panggilan API ke toko kebijakan yang dienkripsi oleh a kunci yang dikelola pelanggan, pengguna atau peran IAM yang membuat panggilan juga harus memiliki akses ke kunci. Jika Izin Terverifikasi tidak dapat mengakses kunci, keputusan otorisasi apa pun yang melibatkan sumber daya yang dienkripsi oleh kunci tersebut mungkin basi atau tidak akurat. Ketika Anda tidak memiliki akses ke kunci, Anda tidak akan dapat read/update/delete sumber daya yang dienkripsi oleh kunci itu, dan setiap panggilan buat untuk memanfaatkan kunci untuk enkripsi akan gagal.

Note

Enkripsi Izin Terverifikasi saat istirahat tersedia di semua AWS Wilayah di mana Izin Terverifikasi tersedia.

Important

Setelah digunakan untuk mengenkripsi penyimpanan kebijakan, Anda **TIDAK DAPAT** memperbarui sumber daya untuk menggunakan kunci yang berbeda untuk enkripsi atau menghapus kunci dari penyimpanan kebijakan tersebut. kunci yang dikelola pelanggan

Menggunakan AWS KMS dan kunci data dengan Izin Terverifikasi Amazon

Fitur enkripsi Izin Terverifikasi Amazon saat istirahat menggunakan AWS KMS kunci dan hierarki kunci data untuk melindungi data sumber daya Anda.

Note

Izin Terverifikasi Amazon hanya mendukung kunci simetris AWS KMS . Anda tidak dapat menggunakan AWS KMS kunci asimetris untuk mengenkripsi sumber daya Izin Terverifikasi Amazon.

Menggunakan Kunci AWS Milik

Izin Terverifikasi Amazon mengenkripsi semua sumber daya secara default dengan AWS kunci yang dimiliki. Kunci ini gratis untuk digunakan dan diputar setiap tahun untuk melindungi sumber daya akun Anda. Anda tidak perlu melihat, mengelola, menggunakan, atau mengaudit kunci ini, sehingga tidak ada tindakan yang diperlukan untuk perlindungan data. Untuk informasi selengkapnya tentang AWS kunci yang [AWS dimiliki, lihat kunci](#) yang dimiliki di Panduan AWS KMS Pengembang.

Menggunakan Kunci yang Dikelola Pelanggan

Memilih enkripsi kunci yang dikelola pelanggan untuk memberikan manfaat berikut:

- Anda membuat dan mengelola AWS KMS kunci, termasuk menyetel kebijakan dan IAM kebijakan utama untuk mengontrol akses ke AWS KMS kunci. Anda dapat mengaktifkan dan menonaktifkan AWS KMS kunci, mengaktifkan dan menonaktifkan rotasi tombol otomatis, dan menghapus AWS KMS kunci ketika tidak lagi digunakan.
- Anda dapat menggunakan bahan kunci kunci yang dikelola pelanggan yang diimpor atau kunci yang dikelola pelanggan di toko kunci khusus yang Anda miliki dan kelola.
- Anda dapat mengaudit enkripsi dan dekripsi sumber daya Izin Terverifikasi dengan memeriksa panggilan API Izin Terverifikasi Amazon ke dalam log. AWS KMS AWS CloudTrail

Untuk Izin Terverifikasi Amazon untuk menggunakan kunci yang dikelola pelanggan s Anda untuk encryption/decryption, you will need to add specific key policies to allow Amazon Verified Permissions to encrypt/decrypt sumber daya atas nama Anda.

Mengotorisasi penggunaan AWS KMS kunci Anda untuk Izin Terverifikasi Amazon

Minimal, Izin Terverifikasi Amazon memerlukan izin berikut pada: kunci yang dikelola pelanggan

- kms:Encrypt
- kms:GenerateDataKeyWithoutPlaintext
- kms:DescribeKey
- kms:ReEncryptTo
- kms:ReEncryptFrom
- kms:Decrypt

Contoh kebijakan kunci dapat dilihat di bawah ini:

```
{
  "Sid": "Enable AVP to use the KMS key for encrypting project J.A.K. policy
resources",
  "Effect": "Allow",
  "Principal": {
    "Service": "verifiedpermissions.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKeyWithoutPlaintext",
    "kms:Encrypt",
    "kms:ReEncryptFrom",
    "kms:ReEncryptTo",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Memahami Konteks Sumber

Konteks sumber memberikan informasi tentang pemanggil sumber yang mencoba melakukan AWS KMS tindakan terhadap kunci yang diberikan. Ini mencegah kebingungan atau penyalahgunaan data terenkripsi dengan mengikat konteks ke sumber data.

Pelanggan dapat menggunakan konteks sumber sebagai ketentuan tambahan pada kebijakan utama mereka seperti pernyataan kebijakan utama berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable this account full access to this key",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Enable AVP to retrieve this key's metadata",
      "Effect": "Allow",
      "Principal": {
```

```

        "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
            "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
        }
    }
},
{
    "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
    "Effect": "Allow",
    "Principal": {
        "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Encrypt"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
            "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
        }
    }
}
]
}

```

Kebijakan kunci ini memungkinkan Izin Terverifikasi untuk melakukan AWS KMS panggilan atas nama Anda, jika akun sumber sama dengan akun tempat AWS KMS kunci ini berada. Nilai-nilai ini

harus dapat diverifikasi saat memeriksa log AWS CloudTrail audit untuk kunci CMK. Untuk informasi selengkapnya tentang kunci AWS kondisi global, lihat [Menggunakan aws:SourceArn atau tombol aws:SourceAccount kondisi](#).

Memahami Konteks Enkripsi

Konteks enkripsi adalah sekumpulan pasangan nilai kunci yang berisi data tambahan yang diautentikasi untuk pemeriksaan integritas enkripsi. Ketika Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, secara AWS KMS kriptografis mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda harus melewati konteks enkripsi yang sama.

Izin Terverifikasi Amazon menggunakan konteks enkripsi yang sama di semua operasi AWS KMS kriptografi dan dapat diverifikasi dalam AWS CloudTrail log saat Izin Terverifikasi melakukan AWS KMS panggilan atas nama Anda untuk proses. encryption/decryption Secara default, Izin Terverifikasi menggunakan pasangan nilai kunci konteks enkripsi berikut saat mengenkripsi sumber daya Anda:

```
{
  "aws:verifiedpermissions:policy-store-arn":
  "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
}
```

Izin Terverifikasi Amazon juga memungkinkan Anda menambahkan konteks enkripsi khusus sebagai bagian dari metadata tambahan yang ingin Anda sertakan selama proses. encryption/decryption Ini berarti kebijakan utama Anda dapat lebih terperinci dalam memberikan izin seperti contoh di bawah ini:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable this account full access to this key",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Enable AVP to retrieve this key's metadata",
      "Effect": "Allow",
```

```

    "Principal": {
      "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
  },
  {
    "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
    "Effect": "Allow",
    "Principal": {
      "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": [
      "kms:Decrypt",
      "kms:ReEncryptTo",
      "kms:ReEncryptFrom",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:Encrypt"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/*",
        "kms:EncryptionContext:policy_owner": "Tim"
      }
    }
  }
]
}

```

Kebijakan kunci ini memungkinkan Izin Terverifikasi untuk melakukan AWS KMS panggilan atas nama Anda, jika peta konteks enkripsi berisi kunci `aws:verifiedpermissions:policy-store-arn`, yang nilainya mengikuti format `arn:aws:verifiedpermissions::111122223333:policy-store/*` dan juga berisi pasangan nilai kunci. `"policy_owner": "Tim"` Lihat [the section called “Membuat toko Kebijakan Terenkripsi”](#) cara mengatur konteks enkripsi kustom.

Note

Disarankan agar kebijakan utama dengan kondisi berdasarkan konteks enkripsi menjadi subset dari peta konteks enkripsi, daripada memeriksa setiap pasangan nilai kunci. Layanan

dan dependensinya di hulu dapat menambahkan pasangan nilai kunci tambahan yang tidak terlihat oleh Anda, dan dapat memengaruhi akses kunci Izin Terverifikasi jika kebijakan kunci memungkinkan secara kondisional berdasarkan tampilan yang tepat dari peta konteks enkripsi.

Memahami kms:ViaService

Kunci `kms:ViaService` kondisi membatasi penggunaan AWS KMS kunci untuk permintaan dari AWS layanan tertentu. Kunci kondisi ini hanya berlaku untuk [sesi akses Teruskan](#) (FAS). Untuk informasi selengkapnya `kms:ViaService`, lihat [kms: ViaService](#) di Panduan AWS KMS Pengembang.

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:ViaService` kondisi untuk mengizinkan a digunakan [kunci yang dikelola pelanggan](#) untuk tindakan yang ditentukan hanya jika permintaan berasal dari Izin Terverifikasi Amazon di wilayah AS Timur (Virginia Utara) atas nama `BrentRole`

```
{
  "Sid": "Enable AVP to encrypt/decrypt resources using credentials of BrentRole",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/BrentRole"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKeyWithoutPlaintext",
    "kms:Encrypt",
    "kms:ReEncryptFrom",
    "kms:ReEncryptTo",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "verifiedpermissions.us-east-1.amazonaws.com"
      ]
    }
  }
}
```

Hal ini diperlukan agar Izin Terverifikasi dapat meneruskan identitas, izin, dan atribut sesi Anda saat Izin Terverifikasi mengajukan permintaan atas nama Anda untuk AWS KMS enkripsi/dekripsi. Untuk informasi selengkapnya tentang permintaan FAS, lihat [Teruskan Sesi Akses](#) di Panduan IAM Pengguna.

Kebijakan AWS KMS Kunci Lengkap

Berdasarkan konsep di bagian sebelumnya, ini adalah contoh kebijakan kunci yang memungkinkan Izin Terverifikasi Amazon menggunakan CMK untuk enkripsi/dekripsi:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable this account full access to this key",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Enable AVP to retrieve this key's metadata",
      "Effect": "Allow",
      "Principal": {
        "Service": "verifiedpermissions.amazonaws.com"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
        }
      }
    },
    {
      "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
      "Effect": "Allow",
```

```

    "Principal": {
      "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": [
      "kms:Decrypt",
      "kms:ReEncryptTo",
      "kms:ReEncryptFrom",
      "kms:Encrypt",
      "kms:GenerateDataKeyWithoutPlaintext"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/*",
        "kms:EncryptionContext:policy_owner": "Tim",
        "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  },
  {
    "Sid": "Enable AVP to encrypt/decrypt resources using credentials of
BrentRole",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/BrentRole"
    },
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:Encrypt",
      "kms:ReEncryptFrom",
      "kms:ReEncryptTo",
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": [
          "verifiedpermissions.us-east-1.amazonaws.com"

```

```

    ]
  },
  "StringLike": {
    "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/*",
    "kms:EncryptionContext:policy_owner": "Tim"
  }
}
]
}

```

Warning

Berhati-hatilah saat memodifikasi kebijakan AWS KMS kunci untuk kunci yang sudah digunakan oleh Izin Terverifikasi Amazon. Meskipun Izin Terverifikasi memvalidasi izin enkripsi dan dekripsi saat pertama kali mengonfigurasi AWS KMS kunci selama pembuatan sumber daya tingkat atas, kunci tersebut tidak dapat memverifikasi perubahan kebijakan berikutnya sesuai permintaan. Menghapus izin yang diperlukan secara tidak sengaja dapat mengganggu keputusan otorisasi Anda dan alur layanan Izin Terverifikasi reguler. Untuk panduan pemecahan masalah kesalahan umum yang terkait dengan kunci yang dikelola pelanggan s di Izin Terverifikasi Amazon, lihat [the section called “Memecahkan Masalah Kunci yang Dikelola Pelanggan di Izin Terverifikasi Amazon”](#)

IAM Kebijakan yang Diperlukan untuk Sumber Daya Terenkripsi

Pelanggan yang memanggil Izin Terverifikasi melalui IAM peran dalam akun mereka perlu memastikan bahwa IAM kebijakan terkait memiliki izin yang tepat untuk memanfaatkan enkripsi dan dekripsi sumber daya. kunci yang dikelola pelanggan

Untuk membuat penyimpanan kebijakan yang dienkripsi oleh a kunci yang dikelola pelanggan, IAM kebijakan berikut menggambarkan tindakan minimum yang diperlukan AWS KMS dan Izin Terverifikasi yang diperlukan untuk melakukannya:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

    "Action": "verifiedpermissions:CreatePolicyStore",
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "kms:Decrypt",
      "kms:Encrypt",
      "kms:ReEncryptTo",
      "kms:ReEncryptFrom",
      "kms:DescribeKey",
      "kms:GenerateDataKeyWithoutPlaintext"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }
]
}

```

Note

Untuk mengambil (operasi Get* dan List*) dan menghapus penyimpanan kebijakan yang dienkripsi oleh a kunci yang dikelola pelanggan, tidak diperlukan izin tambahan.

Untuk memperbarui penyimpanan kebijakan yang dienkripsi oleh kunci yang dikelola pelanggan, mengambil (operasi Get* dan List*), memperbarui, dan menghapus sumber daya turunan dari penyimpanan kebijakan yang dienkripsi oleh a kunci yang dikelola pelanggan, IAM kebijakan berikut menggambarkan tindakan Izin Terverifikasi minimum yang diperlukan dan dilakukan untuk melakukannya: AWS KMS

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "verifiedpermissions:*",
      "Resource": "*",
      "Effect": "Allow"
    },
    {

```

```
        "Action": [
            "kms:Decrypt"
        ],
        "Resource": "*",
        "Effect": "Allow"
    }
]
}
```

Sebagai IAM kebijakan tunggal, pelanggan dapat dengan mudah menambahkan yang berikut ini ke kebijakan IAM peran mereka:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "verifiedpermissions:*",
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom",
        "kms:DescribeKey",
        "kms:GenerateDataKeyWithoutPlaintext"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

Mengelola toko Kebijakan Terenkripsi

Toko kebijakan adalah wadah entry-level yang akan berisi semua sumber daya kebijakan terkait. Untuk informasi selengkapnya tentang penyimpanan kebijakan dan hierarki sumber daya anak, lihat [penyimpanan kebijakan Izin Terverifikasi Amazon](#) di Panduan Pengguna Izin Terverifikasi Amazon.

Saat membuat penyimpanan kebijakan di Izin Terverifikasi, Anda dapat mengaktifkan enkripsi saat istirahat menggunakan AWS KMS kunci. Ini memastikan bahwa:

- Semua operasi baca, perbarui, dan hapus di toko kebijakan, dan sumber daya anak mereka, akan memanfaatkan yang disediakan kunci yang dikelola pelanggan untuk proses dekripsi
- Setiap panggilan keputusan otorisasi (yaitu `IsAuthorized`, `BatchIsAuthorized`, `IsAuthorizedWithToken`, dll.) akan menggunakan yang disediakan kunci yang dikelola pelanggan untuk proses dekripsi

Membuat toko Kebijakan Terenkripsi

Sebelum membuat penyimpanan kebijakan terenkripsi, pastikan bahwa yang kunci yang dikelola pelanggan Anda gunakan memiliki pernyataan kebijakan kunci yang tepat yang ditetapkan untuk Izin Terverifikasi Amazon untuk menggunakan kunci untuk enkripsi/dekripsi. Lihat [the section called “Mengotorisasi penggunaan AWS KMS kunci Anda untuk Izin Terverifikasi Amazon”](#) izin apa yang diperlukan.

Menggunakan AWS CLI:

```
aws verifiedpermissions create-policy-store --region us-east-1 --encryption-settings
file://encrypted.json --validation-settings "{\"mode\": \"OFF\"}"
```

Dimana `encrypted.json` terlihat seperti:

```
{
  "kmsEncryptionSettings": {
    "key": "arn:aws:kms:us-east-1:111122223333:key/12345678-90ab-cdef-ghij-
klmnopqrstuv",
    "encryptionContext": {
      "<ENCRYPTION_CONTEXT_KEY_1>": "<ENCRYPTION_CONTEXT_VALUE_1>",
      "<ENCRYPTION_CONTEXT_KEY_2>": "<ENCRYPTION_CONTEXT_VALUE_2>",
      ...
    }
  }
}
```

```
}
```

Pastikan untuk mengganti key dengan kunci yang dikelola pelanggan ARN Anda dan mengganti <ENCRYPTION_CONTEXT_KEY> dan <ENCRYPTION_CONTEXT_VALUE> memasangkan dengan pasangan kunci-nilai yang diinginkan `encryptionContext`. `encryptionContext` dapat dihilangkan sepenuhnya jika tidak ada penambahan pasangan nilai kunci yang diinginkan.

Important

Jangan sertakan pasangan kunci-nilai `aws:verifiedpermissions:policy-store-arn` dalam konteks enkripsi kustom Anda. Ini ditambahkan secara otomatis dan akan menghasilkan kesalahan validasi jika itu adalah bagian dari pasangan nilai kunci konteks enkripsi kustom Anda yang diteruskan.

Untuk informasi selengkapnya tentang sumber daya anak yang tersedia APIs di penyimpanan kebijakan, lihat [Tindakan](#) di Panduan Referensi API Izin Terverifikasi Amazon.

Note

Jika sumber daya AWS KMS kunci yang dikelola pelanggan Izin Terverifikasi Amazon Anda dihapus, dinonaktifkan, atau tidak dapat diakses karena kebijakan AWS KMS kunci yang salah, dekripsi sumber daya akan gagal, sehingga menghasilkan keputusan otorisasi yang basi. Hilangnya akses dapat bersifat sementara (kebijakan kunci dapat diperbaiki) atau permanen (kunci yang dihapus tidak dapat dipulihkan) tergantung pada keadaan. Kami menyarankan Anda [membatasi akses](#) ke operasi penting, seperti menghapus atau menonaktifkan kunci. AWS KMS Selain itu, kami menyarankan agar organisasi Anda menyiapkan [prosedur akses AWS break-glass](#) untuk memastikan pengguna istimewa Anda dapat mengakses jika AWS Izin Terverifikasi Amazon tidak dapat diakses.

Memantau Interaksi Izin Terverifikasi Amazon dengan AWS KMS

Anda dapat memantau penggunaan Izin Terverifikasi Amazon atas Anda kunci yang dikelola pelanggan . AWS CloudTrail Setiap permintaan AWS KMS melalui Izin Terverifikasi mencakup konteks enkripsi dan ARN kunci yang digunakan (kunci yang dikelola pelanggan Anda) dalam parameter permintaan:

Contoh entri AWS CloudTrail log untuk `GenerateDataKeyWithoutPlaintext`:

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "verifiedpermissions.amazonaws.com"
  },
  "eventTime": "2025-09-28T16:51:04Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "verifiedpermissions.amazonaws.com",
  "userAgent": "verifiedpermissions.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwxyz",
    "encryptionContext": {
      "aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012",
      "policy_store_editor": "Janus"
    },
    ...
  },
  ...
}
```

Contoh entri AWS CloudTrail log untuk Decrypt:

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "verifiedpermissions.amazonaws.com"
  },
  "eventTime": "2025-09-28T16:53:21Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "verifiedpermissions.amazonaws.com",
  "userAgent": "verifiedpermissions.amazonaws.com",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwxyz",
  }
}
```

```

    "encryptionContext": {
      "aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012",
      "policy_store_owner": "Elias"
    }
  },
  ...
}

```

Contoh entri AWS CloudTrail log untuk ReEncrypt:

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "verifiedpermissions.amazonaws.com"
  },
  "eventTime": "2025-09-28T16:51:04Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ReEncrypt",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "verifiedpermissions.amazonaws.com",
  "userAgent": "verifiedpermissions.amazonaws.com",
  "requestParameters": {
    "sourceKeyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwxyz",
    "destinationEncryptionContext": {
      "aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
    },
    "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "destinationKeyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-
ijkl-4567-mnopqrstuvwxyz",
    "sourceEncryptionContext": {
      "aws:verifiedpermissions:policy_store_arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
    },
    "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
    ...
  },
  ...
}

```

Perhatikan bahwa entri log mencakup `invokedBy` referensi utama Izin Terverifikasi Amazon, dan disertakan dalam `encryptionContext/sourceEncryptionContext/destinationEncryptionContext` peta. `requestParameters`

Contoh entri AWS CloudTrail log untuk `DescribeKey`:

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "verifiedpermissions.amazonaws.com"
  },
  "eventTime": "2025-09-28T16:51:02Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "verifiedpermissions.amazonaws.com",
  "userAgent": "verifiedpermissions.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwxyz"
  },
  ...
}
```

Perhatikan bahwa entri log mencakup `invokedBy` referensi utama Izin Terverifikasi Amazon.

Untuk informasi selengkapnya tentang entri AWS CloudTrail log, lihat [Memahami AWS CloudTrail peristiwa](#) di Panduan AWS CloudTrail Pengguna.

Batasan

Topik ini menjelaskan batasan Izin Terverifikasi saat ini dan penggunaan kunci yang dikelola pelanggan untuk enkripsi sumber daya.

- Anda tidak dapat menonaktifkan enkripsi untuk penyimpanan kebijakan setelah diaktifkan
- Setelah Anda membuat penyimpanan kebijakan tanpa enkripsi, Anda tidak dapat memperbarui penyimpanan kebijakan yang akan dienkripsi oleh kunci yang dikelola pelanggan
- Setelah Anda mencabut akses Izin Terverifikasi ke penyimpanan kebijakan terenkripsi yang ada, ada potensi keputusan otorisasi basis kunci yang dikelola pelanggan

- Setelah Anda membuat penyimpanan kebijakan dengan kunci yang dikelola pelanggan, Anda tidak dapat mengubah nilai konteks enkripsi kustom; nilai tersebut adalah nilai statis yang ditetapkan selama pembuatan penyimpanan kebijakan terenkripsi

Memecahkan Masalah Kunci yang Dikelola Pelanggan di Izin Terverifikasi Amazon

Topik ini menjelaskan kesalahan umum kunci yang dikelola pelanggan terkait yang mungkin Anda temui saat menggunakan Izin Terverifikasi Amazon dan menyediakan langkah-langkah pemecahan masalah untuk mengatasinya.

Akses Ditolak: Masalah AWS KMS Izin

Kesalahan: “Layanan atau pemanggil tidak diizinkan untuk menggunakan AWS KMS kunci yang disediakan, karena sumber daya tidak ada di Wilayah ini, tidak ada kebijakan berbasis sumber daya yang mengizinkan akses, atau kebijakan berbasis sumber daya secara eksplisit menolak akses”

Ini bisa berarti bahwa layanan atau pemanggil tidak memiliki izin kms : * tindakan yang diperlukan dalam kebijakan atau IAM kebijakan kunci mereka atau bahwa AWS KMS kunci yang direferensikan tidak ada atau tidak ada lagi.

Pemecahan masalah dengan: AWS CloudTrail

- Cari kms . amazonaws . com acara di AWS CloudTrail
- Cari nama acara AWS KMS operasi yang diidentifikasi tidak diizinkan (yaitu Decrypt,, ReEncrypt, GenerateDataKeyWithoutPlaintextDescribeKey, dll.)
- Tinjau errorCode dan errorMessage bidang
- Periksa userIdentity untuk mengonfirmasi kepala sekolah mana yang mencoba operasi

Untuk mengatasi masalah ini, berikan pengguna atau IAM prinsipal izin akses AWS KMS operasi yang tepat dalam IAM kebijakan dan kebijakan AWS KMS utama mereka. Untuk informasi selengkapnya, lihat [the section called “Kebijakan AWS KMS Kunci Lengkap”](#).

Pengecualian Validasi: Konfigurasi AWS KMS Kunci

Kesalahan: “ AWS KMS Kunci yang dikonfigurasi tidak memiliki konfigurasi yang valid”

Ini berarti bahwa kunci yang direferensikan tidak dapat digunakan oleh layanan untuk kunci yang dikelola pelanggan enkripsi karena konfigurasinya saat ini. Alasan mungkin termasuk kunci yang

dinonaktifkan, kunci tidak didukung EncryptionAlgorithm, atau kunci memiliki tipe yang tidak didukung KeyUsage .

Pengecualian Throttling: AWS KMS Batas Nilai

Kesalahan: “Anda telah melampaui tingkat di mana Anda dapat menelepon AWS KMS”

[Kesalahan ini berarti bahwa Anda telah melampaui AWS KMS batas untuk operasi kriptografi untuk kunci Anda: https://docs.aws.amazon.com/kms/latest/developerguide/requests-per-second.html.](https://docs.aws.amazon.com/kms/latest/developerguide/requests-per-second.html)

Informasi Terkait

- [Mengelola toko Kebijakan Izin Terverifikasi](#)
- [AWS KMS Praktik Terbaik](#)
- [AWS KMS Konteks Enkripsi](#)
- [AWS CloudTrail Integrasi](#)
- [AWS CloudTrail Contoh Entri Log](#)

Manajemen identitas dan akses untuk Izin Terverifikasi Amazon

AWS Identity and Access Management (IAM) adalah Layanan AWS 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 Izin Terverifikasi. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Cara kerja Izin Terverifikasi Amazon IAM](#)
- [IAM kebijakan untuk Izin Terverifikasi](#)
- [Contoh kebijakan berbasis identitas untuk Izin Terverifikasi Amazon](#)
- [AWS kebijakan terkelola untuk Izin Terverifikasi Amazon](#)
- [Memecahkan masalah identitas dan akses Izin Terverifikasi Amazon](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat [Memecahkan masalah identitas dan akses Izin Terverifikasi Amazon](#))
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat [Cara kerja Izin Terverifikasi Amazon IAM](#))
- IAM administrator - menulis kebijakan untuk mengelola akses (lihat [Contoh kebijakan berbasis identitas untuk Izin Terverifikasi Amazon](#))

Mengautentikasi dengan identitas

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

Anda dapat masuk sebagai identitas federasi menggunakan kredensial dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensial. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat [Cara masuk ke Akun AWS Anda](#) dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat [Versi AWS Tanda Tangan 4 untuk permintaan API](#) di Panduan IAM Pengguna.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root di Panduan Pengguna IAM](#)

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensial dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat [Apa itu Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat [Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara](#) di IAM Panduan Pengguna.

[IAM Grup menentukan kumpulan](#) pengguna IAM dan membuat izin lebih mudah dikelola untuk kumpulan pengguna yang besar. Untuk informasi selengkapnya, lihat [Kasus penggunaan untuk IAM pengguna](#) di Panduan IAM Pengguna.

IAM peran

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

IAM peran dengan kredensi 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 mengautentikasi, 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 \(federasi\)](#) di Panduan IAM Pengguna. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, IAM Identity Center mengkorelasikan izin yang disetel ke peran. IAM Untuk informasi tentang set izin, lihat [Set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .

- Izin pengguna IAM sementara — Pengguna atau peran IAM 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 Layanan AWS, 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 Perbedaan IAM peran dari kebijakan berbasis sumber daya di Panduan Pengguna](#).IAM
- Aplikasi berjalan pada Amazon EC2 — Anda dapat menggunakan IAM peran untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat AWS CLI atau AWS permintaan API. Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan IAM peran untuk memberikan izin ke aplikasi yang berjalan pada Amazon EC2 instance](#) di IAM Panduan Pengguna.

Untuk mempelajari apakah akan menggunakan IAM peran atau pengguna IAM, 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 menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat [Ringkasan kebijakan JSON](#) di IAM Panduan Pengguna.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. IAM Administrator membuat IAM kebijakan dan menambahkannya ke peran, yang kemudian dapat diasumsikan oleh pengguna. IAM kebijakan menentukan izin terlepas dari metode yang digunakan untuk melakukan operasi.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Menentukan IAM izin khusus dengan kebijakan yang dikelola pelanggan di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan sebaris, lihat [Memilih antara kebijakan terkelola dan kebijakan sebaris](#) di IAM Panduan Pengguna.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan IAM peran dan kebijakan Amazon S3 bucket. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya.

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 [Ringkasan daftar kontrol akses \(ACL\)](#) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batas izin — Tetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas. IAM Untuk informasi selengkapnya, lihat [Batas izin untuk IAM entitas](#) di Panduan IAM Pengguna.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Kebijakan kontrol sumber daya \(RCPs\)](#) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. 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 Izin Terverifikasi Amazon IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Izin Terverifikasi, pelajari IAM fitur apa saja yang tersedia untuk digunakan dengan Izin Terverifikasi.

IAM fitur yang dapat Anda gunakan dengan Izin Terverifikasi Amazon

IAM fitur	Dukungan Izin Terverifikasi
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Tidak

IAM fitur	Dukungan Izin Terverifikasi
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Tidak
Peran terkait layanan	Tidak

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

Kebijakan berbasis identitas untuk Izin Terverifikasi

Mendukung kebijakan berbasis identitas	Ya
--	----

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. 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 [Menentukan IAM izin khusus dengan kebijakan yang dikelola pelanggan 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. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat [referensi elemen kebijakan IAM JSON](#) di IAM Panduan Pengguna.

Contoh kebijakan berbasis identitas untuk Izin Terverifikasi

Untuk melihat contoh kebijakan berbasis identitas Izin Terverifikasi, lihat. [Contoh kebijakan berbasis identitas untuk Izin Terverifikasi Amazon](#)

Kebijakan berbasis sumber daya dalam Izin Terverifikasi

Mendukung kebijakan berbasis sumber daya Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON 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 principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau IAM entitas di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun IAM di](#) Panduan IAM Pengguna.

Tindakan kebijakan untuk Izin Terverifikasi

Mendukung tindakan kebijakan Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan Izin Terverifikasi, lihat [Tindakan yang ditentukan oleh Izin Terverifikasi Amazon di Referensi](#) Otorisasi Layanan.

Tindakan kebijakan dalam Izin Terverifikasi menggunakan awalan berikut sebelum tindakan:

```
verifiedpermissions
```

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

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

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata Get, sertakan tindakan berikut:

```
"Action": "verifiedpermissions:Get*"
```

Untuk melihat contoh kebijakan berbasis identitas Izin Terverifikasi, lihat [Contoh kebijakan berbasis identitas untuk Izin Terverifikasi Amazon](#)

Sumber daya kebijakan untuk Izin Terverifikasi

Mendukung sumber daya kebijakan

Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya Izin Terverifikasi dan jenis sumber daya ARNs, lihat [Jenis sumber daya yang ditentukan oleh Izin Terverifikasi Amazon](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh Izin Terverifikasi Amazon](#).

Kunci kondisi kebijakan untuk Izin Terverifikasi

Mendukung kunci kondisi kebijakan khusus layanan	Tidak
--	-------

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. 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. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan IAM Pengguna.

ACLs di Izin Terverifikasi

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 Izin Terverifikasi

Mendukung ABAC (tanda dalam kebijakan)	Ya
--	----

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut tanda. Anda dapat melampirkan tag ke IAM entitas dan AWS sumber daya, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda 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 selengkapnya tentang ABAC, lihat [Mendefinisikan izin dengan otorisasi ABAC](#) di Panduan Pengguna.IAM Untuk melihat tutorial dengan langkah-langkah untuk menyiapkan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) di Panduan Pengguna.IAM

Menggunakan kredensi sementara dengan Izin Terverifikasi

Mendukung penggunaan kredensial sementara Ya

Kredensi sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. 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](#) dan [Layanan AWS yang berfungsi IAM di Panduan](#) Pengguna.IAM

Izin utama lintas layanan untuk Izin Terverifikasi

Mendukung izin pengguna utama Ya

Sesi akses terusan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses terusan](#).

Peran layanan untuk Izin Terverifikasi

Mendukung peran layanan Tidak

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 Layanan AWS](#) dalam IAM Panduan Pengguna.

Peran terkait layanan untuk Izin Terverifikasi

Mendukung peran terkait layanan

Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS 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.

IAM kebijakan untuk Izin Terverifikasi

Izin Terverifikasi mengelola izin pengguna dalam aplikasi Anda. Agar aplikasi Anda memanggil Izin Terverifikasi APIs atau agar Konsol Manajemen AWS pengguna diizinkan mengelola kebijakan Cedar di toko kebijakan Izin Terverifikasi, Anda harus menambahkan izin yang diperlukan. IAM

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON 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 (tercantum di bawah). 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 kebijakan JSON, lihat [referensi elemen kebijakan IAM JSON](#) di IAM Panduan Pengguna.

Tindakan	Deskripsi
CreateIdentitySource	Tindakan untuk membuat sumber identitas baru.

Tindakan	Deskripsi
CreatePolicy	Tindakan untuk membuat kebijakan Cedar di toko kebijakan. Anda dapat membuat kebijakan statis atau kebijakan yang ditautkan ke templat kebijakan.
CreatePolicyStore	Tindakan untuk membuat toko kebijakan baru.
CreatePolicyTemplate	Tindakan untuk membuat templat kebijakan baru.
DeleteIdentitySource	Tindakan untuk menghapus sumber identitas.
DeletePolicy	Tindakan untuk menghapus kebijakan dari toko kebijakan.
DeletePolicyStore	Tindakan untuk menghapus toko kebijakan.
DeletePolicyTemplate	Tindakan untuk menghapus templat kebijakan.
GetIdentitySource	Tindakan untuk mendapatkan sumber identitas.
GetPolicy	Tindakan untuk mengambil informasi tentang kebijakan tertentu.
GetPolicyStore	Tindakan untuk mengambil informasi tentang penyimpanan kebijakan tertentu.
GetPolicyTemplate	Tindakan untuk mendapatkan templat kebijakan.
GetSchema	Tindakan untuk mendapatkan skema.
IsAuthorized	Tindakan untuk mendapatkan respons otorisasi berdasarkan parameter yang dijelaskan dalam permintaan otorisasi .

Tindakan	Deskripsi
IsAuthorizedWithToken	Tindakan untuk mendapatkan respons otorisasi berdasarkan parameter yang dijelaskan dalam permintaan otorisasi di mana prinsipal berasal dari token identitas.
ListIdentitySources	Tindakan untuk membuat daftar semua sumber identitas di Akun AWS.
ListPolicies	Tindakan untuk mencantumkan semua kebijakan di toko kebijakan.
ListPolicyStores	Tindakan untuk mencantumkan semua toko kebijakan di Akun AWS.
ListPolicyTemplates	Tindakan untuk mencantumkan semua templat kebijakan di Akun AWS.
ListTagsForResource	Tindakan untuk mencantumkan semua tag untuk sumber daya.
PutSchema	Tindakan untuk menambahkan skema ke toko kebijakan.
TagResource	Tindakan untuk menambahkan tag ke sumber daya.
UpdateIdentitySource	Tindakan untuk memperbarui sumber identitas.
UpdatePolicy	Tindakan untuk memperbarui kebijakan di toko kebijakan.
UpdatePolicyStore	Tindakan untuk memperbarui toko kebijakan.
UpdatePolicyTemplate	Tindakan untuk memperbarui templat kebijakan .
UntagResource	Tindakan untuk menghapus tag dari sumber daya.

Contoh IAM kebijakan untuk izin CreatePolicy tindakan:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:CreatePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

Contoh kebijakan berbasis identitas untuk Izin Terverifikasi Amazon

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau mengubah sumber daya Izin Terverifikasi. Mereka juga tidak dapat melakukan tugas dengan menggunakan Konsol Manajemen AWS, AWS Command Line Interface (AWS CLI), atau AWS API. IAM Administrator harus membuat IAM kebijakan yang memberikan izin kepada pengguna dan peran untuk melakukan tindakan pada sumber daya yang mereka butuhkan. Administrator kemudian harus melampirkan kebijakan tersebut untuk pengguna yang membutuhkannya.

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

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Izin Terverifikasi, termasuk format ARNs untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk Izin Terverifikasi Amazon](#) di Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol Izin Terverifikasi](#)
- [Mengizinkan pengguna melihat izin mereka sendiri](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Izin Terverifikasi 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 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. Sebagai contoh, 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 Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat [elemen kebijakan IAM JSON: 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. IAM Access 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 [Memvalidasi kebijakan dengan IAM Access Analyzer](#) di Panduan Pengguna.IAM
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan.

Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Akses API aman dengan MFA](#) di IAM Panduan Pengguna.

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

Menggunakan konsol Izin Terverifikasi

Untuk mengakses konsol Izin Terverifikasi Amazon, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Izin Terverifikasi 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 sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol Izin Terverifikasi, lampirkan juga Izin Terverifikasi *ConsoleAccess* atau kebijakan *ReadOnly* AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan izin ke pengguna](#) di Panduan IAM Pengguna.

Mengizinkan pengguna melihat izin mereka sendiri

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
```

```
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

AWS kebijakan terkelola untuk Izin Terverifikasi Amazon

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Butuh waktu dan keahlian untuk [membuat kebijakan terkelola IAM pelanggan](#) yang hanya memberi tim Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di Akun AWS Anda. Untuk informasi selengkapnya tentang kebijakan AWS [AWS terkelola](#), [lihat kebijakan terkelola](#) di Panduan IAM Pengguna.

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan yang dikelola AWS untuk mendukung fitur-fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan yang dikelola AWS saat ada fitur baru yang diluncurkan atau saat ada operasi baru yang tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan `ReadOnlyAccess` AWS terkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS menambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk daftar dan deskripsi kebijakan fungsi pekerjaan, lihat [kebijakan AWS terkelola untuk fungsi pekerjaan](#) di Panduan IAM Pengguna.

AWS kebijakan terkelola: `AmazonVerifiedPermissionsFullAccess`

Kebijakan `AmazonVerifiedPermissionsFullAccess` terkelola memberikan akses penuh ke Izin Terverifikasi. Untuk bekerja dengan sumber identitas Amazon Cognito berbasis, Anda harus melampirkan kebijakan terpisah, seperti [AmazonCognitoReadOnly](#) kebijakan.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccountLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:CreatePolicyStore",
        "verifiedpermissions:ListPolicyStores"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PolicyStoreLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:*"
      ],
      "Resource": [
        "arn:aws:verifiedpermissions::*:policy-store/*"
      ]
    }
  ]
}
```

AWS kebijakan terkelola: AmazonVerifiedPermissionsReadOnlyAccess

Kebijakan AmazonVerifiedPermissionsReadOnlyAccess terkelola memberikan akses hanya-baca ke Izin Terverifikasi.

Kebijakan ini memberikan akses ke semua operasi baca Izin Terverifikasi Amazon, termasuk kueri otorisasi dan. APIs IsAuthorized IsAuthorizedWithToken

Note

Akses ke BatchIsAuthorized dan BatchIsAuthorizedWithToken diberikan secara otomatis ketika akses diberikan kepada IsAuthorized danIsAuthorizedWithToken, masing-masing.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccountLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:ListPolicyStores"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PolicyStoreLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:GetIdentitySource",
        "verifiedpermissions:GetPolicy",
        "verifiedpermissions:GetPolicyStore",
        "verifiedpermissions:GetPolicyTemplate",
        "verifiedpermissions:GetSchema",
        "verifiedpermissions:IsAuthorized",
        "verifiedpermissions:IsAuthorizedWithToken",
        "verifiedpermissions:ListIdentitySources",
        "verifiedpermissions:ListPolicies",

```

```

    "verifiedpermissions:ListPolicyTemplates"
  ],
  "Resource": [
    "arn:aws:verifiedpermissions::*:policy-store/*"
  ]
}
]
}

```

Pembaruan Izin Terverifikasi ke kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Izin Terverifikasi sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen Izin Terverifikasi.

Ubah	Deskripsi	Date
AmazonVerifiedPermissionsFullAccess – Kebijakan baru	Izin Terverifikasi menambahkan kebijakan baru untuk mengizinkan akses penuh ke Izin Terverifikasi.	Oktober 11, 2024
AmazonVerifiedPermissionsReadOnlyAccess – Kebijakan baru	Izin Terverifikasi menambahkan kebijakan baru untuk mengizinkan akses ke semua operasi baca Izin Terverifikasi Amazon, termasuk kueri otorisasi dan. APIs <code>IsAuthorized</code> <code>IsAuthorizedWithToken</code>	Oktober 11, 2024
Izin Terverifikasi mulai melacak perubahan	Izin Terverifikasi mulai melacak perubahan untuk kebijakan yang AWS dikelola.	Oktober 11, 2024

Memecahkan masalah identitas dan akses Izin Terverifikasi Amazon

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Izin Terverifikasi dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di Izin Terverifikasi](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Izin Terverifikasi saya](#)

Saya tidak berwenang untuk melakukan tindakan di Izin Terverifikasi

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 pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `verifiedpermissions:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
verifiedpermissions: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 `verifiedpermissions: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 Izin Terverifikasi.

Beberapa Layanan AWS 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 pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Izin Terverifikasi. 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 Izin Terverifikasi 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 Izin Terverifikasi mendukung fitur ini, lihat [Cara kerja Izin Terverifikasi Amazon IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di 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 Izin Terverifikasi Amazon

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) 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 Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat [Dokumentasi AWS Keamanan](#).

Ketahanan dalam Izin Terverifikasi Amazon

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.

Saat Anda membuat penyimpanan kebijakan Izin Terverifikasi, penyimpanan tersebut dibuat dalam satu individu AWS Region, dan secara otomatis direplikasi di seluruh pusat data yang membentuk Availability Zone Wilayah tersebut. Saat ini, Izin Terverifikasi tidak mendukung replikasi lintas wilayah apa pun.

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

Memantau panggilan API Izin Terverifikasi Amazon

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Izin Terverifikasi Amazon dan AWS solusi Anda lainnya. AWS menyediakan alat berikut untuk memantau Izin Terverifikasi, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- AWS CloudTrail menangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda dan mengirimkan file log ke Amazon S3 bucket 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](#).

Untuk informasi selengkapnya tentang memantau Izin Terverifikasi dengan CloudTrail, lihat [Mencatat panggilan API Izin Terverifikasi Amazon menggunakan AWS CloudTrail](#).

Mencatat panggilan API Izin Terverifikasi Amazon menggunakan AWS CloudTrail

Izin Terverifikasi Amazon terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan dalam Izin Terverifikasi. CloudTrail menangkap semua panggilan API untuk Izin Terverifikasi sebagai peristiwa. Panggilan yang diambil mencakup panggilan dari konsol Izin Terverifikasi dan panggilan kode ke operasi API Izin Terverifikasi. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke Amazon S3 bucket, termasuk peristiwa untuk Izin Terverifikasi. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa tindakan manajemen terbaru di CloudTrail konsol dalam riwayat Acara, tetapi bukan peristiwa untuk panggilan API seperti `isAuthorized`. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Izin Terverifikasi, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

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

Informasi Izin Terverifikasi di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di Izin Terverifikasi, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa

AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di Akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat peristiwa dengan Riwayat CloudTrail acara](#).

Untuk catatan peristiwa yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk Izin Terverifikasi, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke Amazon S3 ember. 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 Amazon S3 bucket 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](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua tindakan Izin Terverifikasi dicatat oleh CloudTrail dan didokumentasikan dalam Panduan [Referensi API Izin Terverifikasi Amazon](#). Misalnya, panggilan `keCreateIdentitySource`, `DeletePolicy`, dan `ListPolicyStores` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal 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 gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat [Elemen userIdentity CloudTrail](#).

Peristiwa data seperti [IsAuthorized](#) dan [IsAuthorizedWithToken](#) tidak dicatat secara default saat Anda membuat penyimpanan data jejak atau peristiwa. Untuk merekam peristiwa CloudTrail data, Anda

harus secara eksplisit menambahkan sumber daya atau jenis sumber daya yang didukung yang ingin Anda kumpulkan aktivitasnya. Untuk informasi selengkapnya, lihat [Peristiwa data](#) di Panduan AWS CloudTrail Pengguna.

Memahami entri file log Izin Terverifikasi

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke Amazon S3 bucket 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 panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Untuk panggilan API otorisasi, elemen respons, seperti keputusan, disertakan di bawah `additionalEventData` bukan `responseElements`.

Topik

- [IsAuthorized](#)
- [BatchIsAuthorized](#)
- [CreatePolicyStore](#)
- [ListPolicyStores](#)
- [DeletePolicyStore](#)
- [PutSchema](#)
- [GetSchema](#)
- [CreatePolicyTemplate](#)
- [DeletePolicyTemplate](#)
- [CreatePolicy](#)
- [GetPolicy](#)
- [CreateIdentitySource](#)
- [GetIdentitySource](#)
- [ListIdentitySources](#)
- [DeleteIdentitySource](#)

Note

Beberapa bidang telah disensor dari contoh untuk privasi data.

IsAuthorized

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-11-20T22:55:03Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "IsAuthorized",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
  "requestParameters": {
    "principal": {
      "entityType": "PhotoFlash::User",
      "entityId": "alice"
    },
    "action": {
      "actionType": "PhotoFlash::Action",
      "actionId": "ViewPhoto"
    },
    "resource": {
      "entityType": "PhotoFlash::Photo",
      "entityId": "VacationPhoto94.jpg"
    },
    "policyStoreId": "PSEXAMPLEabcdefg1111111"
  },
  "responseElements": null,
  "additionalEventData": {
    "decision": "ALLOW"
  },
  "requestID": "346c4b6a-d12f-46b6-bc06-6c857bd3b28e",
```

```

    "eventID": "8a4fed32-9605-45dd-a09a-5ebbf0715bbc",
    "readOnly": true,
    "resources": [
      {
        "accountId": "123456789012",
        "type": "AWS::VerifiedPermissions::PolicyStore",
        "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
  }

```

BatchIsAuthorized

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-11-20T23:02:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "BatchIsAuthorized",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
  "requestParameters": {
    "requests": [
      {
        "principal": {
          "entityType": "PhotoFlash::User",
          "entityId": "alice"
        },
        "action": {
          "actionType": "PhotoFlash::Action",

```

```
        "actionId": "ViewPhoto"
      },
      "resource": {
        "entityType": "PhotoFlash::Photo",
        "entityId": "VacationPhoto94.jpg"
      }
    },
    {
      "principal": {
        "entityType": "PhotoFlash::User",
        "entityId": "annalisa"
      },
      "action": {
        "actionType": "PhotoFlash::Action",
        "actionId": "DeletePhoto"
      },
      "resource": {
        "entityType": "PhotoFlash::Photo",
        "entityId": "VacationPhoto94.jpg"
      }
    }
  ],
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
},
"responseElements": null,
"additionalEventData": {
  "results": [
    {
      "request": {
        "principal": {
          "entityType": "PhotoFlash::User",
          "entityId": "alice"
        },
        "action": {
          "actionType": "PhotoFlash::Action",
          "actionId": "ViewPhoto"
        },
        "resource": {
          "entityType": "PhotoFlash::Photo",
          "entityId": "VacationPhoto94.jpg"
        }
      },
      "decision": "ALLOW"
    }
  ]
},
```

```

    {
      "request": {
        "principal": {
          "entityType": "PhotoFlash::User",
          "entityId": "annalisa"
        },
        "action": {
          "actionType": "PhotoFlash::Action",
          "actionId": "DeletePhoto"
        },
        "resource": {
          "entityType": "PhotoFlash::Photo",
          "entityId": "VacationPhoto94.jpg"
        }
      },
      "decision": "DENY"
    }
  ],
  "requestID": "a8a5caf3-78bd-4139-924c-7101a8339c3b",
  "eventID": "7d81232f-f3d1-4102-b9c9-15157c70487b",
  "readOnly": true,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "123456789012",
  "eventCategory": "Data"
}

```

CreatePolicyStore

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",

```

```

    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "validationSettings": {
      "mode": "OFF"
    }
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/PSEXAMPLEabcdefghijklmnop111111",
    "createdDate": "2023-05-22T07:43:33.962794Z",
    "lastUpdatedDate": "2023-05-22T07:43:33.962794Z"
  },
  "requestID": "1dd9360e-e2dc-4554-ab65-b46d2cf45c29",
  "eventID": "b6edae-3584-4b4e-a48e-311de46d7532",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}

```

ListPolicyStores

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },

```

```
"eventTime": "2023-05-22T07:43:33Z",
"eventSource": "verifiedpermissions.amazonaws.com",
"eventName": "ListPolicyStores",
"awsRegion": "us-west-2",
"sourceIPAddress": "203.0.113.0",
"userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
"requestParameters": {
  "maxResults": 10
},
"responseElements": null,
"requestID": "5ef238db-9f87-4f37-ab7b-6cf0ba5df891",
"eventID": "b0430fb0-12c3-4cca-8d05-84c37f99c51f",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}
```

DeletePolicyStore

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeletePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "1368e8f9-130d-45a5-b96d-99097ca3077f",
  "eventID": "ac482022-b2f6-4069-879a-dd509123d8d7",
  "readOnly": false,
}
```

```

"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

PutSchema

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-16T12:58:57Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "PutSchema",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "lastUpdatedDate": "2023-05-16T12:58:57.513442Z",
    "namespaces": "[some_namespace]",
    "createdDate": "2023-05-16T12:58:57.513442Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
  },
  "requestID": "631fbfa1-a959-4988-b9f8-f1a43ff5df0d",
  "eventID": "7cd0c677-733f-4602-bc03-248bae581fe5",
  "readOnly": false,

```

```
"resources": [  
  {  
    "accountId": "123456789012",  
    "type": "AWS::VerifiedPermissions::PolicyStore",  
    "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/  
PSEXAMPLEabcdefg111111"  
  }  
],  
"eventType": "AwsApiCall",  
"managementEvent": true,  
"recipientAccountId": "123456789012",  
"eventCategory": "Management"  
}
```

GetSchema

```
{  
  "eventVersion": "1.08",  
  "userIdentity": {  
    "type": "AssumedRole",  
    "principalId": "EXAMPLE_PRINCIPAL_ID",  
    "arn": "arn:aws:iam::222222222222:role/ExampleRole",  
    "accountId": "222222222222",  
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"  
  },  
  "eventTime": "2023-05-25T01:12:07Z",  
  "eventSource": "verifiedpermissions.amazonaws.com",  
  "eventName": "GetSchema",  
  "awsRegion": "us-west-2",  
  "sourceIPAddress": "203.0.113.0",  
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",  
  "requestParameters": {  
    "policyStoreId": "PSEXAMPLEabcdefg111111"  
  },  
  "responseElements": null,  
  "requestID": "a1f4d4cd-6156-480a-a9b8-e85a71dcc7c2",  
  "eventID": "0b3b8e3d-155c-46f3-a303-7e9e8b5f606b",  
  "readOnly": true,  
  "resources": [  
    {  
      "accountId": "222222222222",  
      "type": "AWS::VerifiedPermissions::PolicyStore",  

```

```
    "ARN": "arn:aws:verifiedpermissions::222222222222:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "222222222222",
"eventCategory": "Management"
}
```

CreatePolicyTemplate

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-16T13:00:24Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyTemplate",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "lastUpdatedDate": "2023-05-16T13:00:23.444404Z",
    "createdDate": "2023-05-16T13:00:23.444404Z",
    "policyTemplateId": "PTEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
  },
  "requestID": "73953bda-af5e-4854-afe2-7660b492a6d0",
  "eventID": "7425de77-ed84-4f91-a4b9-b669181cc57b",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
    }
  ]
}
```

```

    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

DeletePolicyTemplate

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::222222222222:role/ExampleRole",
    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-25T01:11:48Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeletePolicyTemplate",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "5ff0f22e-6bbd-4b85-a400-4fb74aa05dc6",
  "eventID": "c0e0c689-369e-4e95-a9cd-8de113d47ffa",
  "readOnly": false,
  "resources": [
    {
      "accountId": "222222222222",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::222222222222:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
}

```

```

"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "222222222222",
"eventCategory": "Management"
}

```

CreatePolicy

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:42:30Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
      "entityType": "PhotoApp::Role",
      "entityId": "PhotoJudge"
    },
    "resource": {
      "entityType": "PhotoApp::Application",
      "entityId": "PhotoApp"
    },
    "lastUpdatedDate": "2023-05-22T07:42:30.70852Z",
    "createdDate": "2023-05-22T07:42:30.70852Z"
  },
  "requestID": "93ffa151-3841-4960-9af6-30a7f817ef93",

```

```

"eventID": "30ab405f-3dff-43ff-8af9-f513829e8bde",
"readOnly": false,
"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

GetPolicy

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:29Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetPolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "23022a9e-2f5c-4dac-b653-59e6987f2fac",
  "eventID": "9b4d5037-bafa-4d57-b197-f46af83fc684",
  "readOnly": true,
  "resources": [
    {

```

```

    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

CreateIdentitySource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-19T01:27:44Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreateIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111",
    "configuration": {
      "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:000011112222:us-east-1:userpool/us-
east-1_aaaaaaaaaa"
      }
    },
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "principalEntityType": "User"
  },
  "responseElements": {
    "createdDate": "2023-07-14T15:05:01.599534Z",
    "identitySourceId": "ISEXAMPLEabcdefg111111",

```

```

    "lastUpdatedDate": "2023-07-14T15:05:01.599534Z",
    "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111"
  },
  "requestID": "afcc1e67-d5a4-4a9b-a74c-cdc2f719391c",
  "eventID": "f13a41dc-4496-4517-aeb8-a389eb379860",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefghijklmnop111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}

```

GetIdentitySource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:31Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefghijklmnop111111",
    "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111"
  },
  "responseElements": null,
  "requestID": "7a6ecf79-c489-4516-bb57-9ded970279c9",
}

```

```

"eventID": "fa158e6c-f705-4a15-a731-2cdb4bd9a427",
"readOnly": true,
"resources": [
  {
    "accountId": "333333333333",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "333333333333",
"eventCategory": "Management"
}

```

ListIdentitySources

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T20:05:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListIdentitySources",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "95d2a7bc-7e9a-4efe-918e-97e558aacaf7",
  "eventID": "d3dc53f6-1432-40c8-9d1d-b9eeb75c6193",
  "readOnly": true,
  "resources": [
    {
      "accountId": "333333333333",

```

```

    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "333333333333",
"eventCategory": "Management"
}

```

DeleteIdentitySource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeleteIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEEabcdefg111111",
    "policyStoreId": "PSEXAMPLEEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "d554d964-0957-4834-a421-c417bd293086",
  "eventID": "fe4d867c-88ee-4e5d-8d30-2fbc208c9260",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEEabcdefg111111"
    }
  ]
}

```

```
],  
  "eventType": "AwsApiCall",  
  "managementEvent": true,  
  "recipientAccountId": "333333333333",  
  "eventCategory": "Management"  
}
```

Membuat sumber daya Izin Terverifikasi Amazon dengan AWS CloudFormation

Izin Terverifikasi Amazon terintegrasi dengan AWS CloudFormation, layanan yang membantu Anda memodelkan dan menyiapkan AWS sumber daya sehingga Anda dapat menghabiskan lebih sedikit waktu untuk membuat dan mengelola sumber daya dan infrastruktur Anda. Anda membuat templat yang menjelaskan semua AWS sumber daya yang Anda inginkan (seperti penyimpanan kebijakan), dan CloudFormation ketentuan serta mengonfigurasi sumber daya tersebut untuk Anda.

Saat menggunakannya CloudFormation, Anda dapat menggunakan kembali template untuk menyiapkan sumber daya Izin Terverifikasi secara konsisten dan berulang kali. Jelaskan sumber daya Anda sekali, lalu sediakan sumber daya yang sama berulang-ulang di beberapa Akun AWS dan Wilayah.

Important

Identitas Amazon Cognito tidak tersedia sama dengan Izin Terverifikasi Wilayah AWS Amazon. Jika Anda menerima kesalahan CloudFormation terkait Identitas Amazon Cognito, seperti `Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient`, kami sarankan Anda membuat kumpulan Amazon Cognito pengguna dan klien di tempat terdekat secara geografis di mana Identitas Amazon AWS Region Cognito tersedia. Gunakan kumpulan pengguna yang baru dibuat ini saat membuat sumber identitas Izin Terverifikasi.

Izin dan CloudFormation templat terverifikasi

Untuk menyediakan dan mengonfigurasi sumber daya untuk Izin Terverifikasi dan layanan terkait, Anda harus memahami [CloudFormation templat](#). Templat adalah file teks dengan format JSON atau YAML. Template ini menjelaskan sumber daya yang ingin Anda sediakan di CloudFormation tumpukan Anda. Jika Anda tidak terbiasa dengan JSON atau YAMB, Anda dapat menggunakan CloudFormation Designer untuk membantu Anda memulai dengan template. CloudFormation Untuk informasi lebih lanjut, lihat [Apa itu CloudFormation Desainer?](#) dalam AWS CloudFormation User Guide.

Izin Terverifikasi mendukung pembuatan sumber identitas, kebijakan, penyimpanan kebijakan, templat kebijakan, dan alias penyimpanan kebijakan di. CloudFormation Untuk informasi

selengkapnya, termasuk contoh templat JSON dan YAMAL untuk sumber daya Izin Terverifikasi, lihat [referensi jenis sumber daya Izin Terverifikasi Amazon](#) di Panduan Pengguna.AWS CloudFormation

AWS Konstruksi CDK

AWS Cloud Development Kit (AWS CDK) Ini adalah kerangka pengembangan perangkat lunak open-source untuk mendefinisikan infrastruktur cloud dalam kode dan menyediakannya. CloudFormation Konstruksi, atau komponen cloud yang dapat digunakan kembali, dapat digunakan untuk membuat CloudFormation templat. Template ini kemudian dapat digunakan untuk menyebarkan infrastruktur cloud Anda.

Untuk mempelajari lebih lanjut dan mengunduh AWS CDK, lihat [AWS Cloud Development Kit](#).

Berikut ini adalah tautan ke dokumentasi untuk AWS CDK sumber daya Izin Terverifikasi, seperti konstruksi.

- [Izin Terverifikasi Amazon Konstruksi L2 CDK](#)

Pelajari lebih lanjut tentang CloudFormation

Untuk mempelajari selengkapnya CloudFormation, lihat sumber daya berikut:

- [AWS CloudFormation](#)
- [AWS CloudFormation Panduan Pengguna](#)
- [CloudFormation Referensi API](#)
- [AWS CloudFormation Panduan Pengguna Antarmuka Baris Perintah](#)

Akses Izin Terverifikasi Amazon menggunakan AWS PrivateLink

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

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, 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 Izin Terverifikasi.

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

Pertimbangan untuk Izin Terverifikasi

Sebelum Anda menyiapkan titik akhir antarmuka untuk Izin Terverifikasi, tinjau [Pertimbangan dalam Panduan](#).AWS PrivateLink

Izin Terverifikasi mendukung panggilan ke semua tindakan API-nya melalui titik akhir antarmuka.

Kebijakan titik akhir VPC tidak didukung untuk Izin Terverifikasi. Secara default, akses penuh ke Izin Terverifikasi diizinkan melalui titik akhir antarmuka. Atau, Anda dapat mengaitkan grup keamanan dengan antarmuka jaringan titik akhir untuk mengontrol lalu lintas ke Izin Terverifikasi melalui titik akhir antarmuka.

Buat titik akhir antarmuka untuk Izin Terverifikasi

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

Buat titik akhir antarmuka untuk Izin Terverifikasi menggunakan nama layanan berikut:

```
com.amazonaws.region.verifiedpermissions
```

Jika Anda mengaktifkan DNS pribadi untuk titik akhir antarmuka, Anda dapat membuat permintaan API ke Izin Terverifikasi menggunakan nama DNS Regional default. Misalnya, `verifiedpermissions.us-east-1.amazonaws.com`.

Buat kebijakan titik akhir untuk titik akhir antarmuka Anda

Kebijakan endpoint adalah sumber daya IAM yang dapat Anda lampirkan ke titik akhir antarmuka. Kebijakan endpoint default memungkinkan akses penuh ke Izin Terverifikasi melalui titik akhir antarmuka. Untuk mengontrol akses yang diizinkan ke Izin Terverifikasi dari VPC Anda, lampirkan kebijakan titik akhir kustom ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (Akun AWS, pengguna IAM, dan peran IAM).
- 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 titik akhir VPC untuk tindakan Izin Terverifikasi

Berikut ini adalah contoh kebijakan endpoint kustom. Saat Anda melampirkan kebijakan ini ke titik akhir antarmuka Anda, kebijakan ini akan memberikan akses ke tindakan Izin Terverifikasi yang tercantum untuk semua prinsipal di semua sumber daya.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:IsAuthorized",
        "verifiedpermissions:IsAuthorizedWithToken",
        "verifiedpermissions:GetPolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

}

Kuota untuk Izin Terverifikasi Amazon

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 Izin Terverifikasi, buka konsol [Service Quotas](#). Di panel navigasi, pilih AWS layanan dan pilih Izin Terverifikasi.

Untuk meminta peningkatan kuota, lihat [Meminta Peningkatan Kuota](#) dalam Panduan Pengguna Service Quotas. Jika kuota belum tersedia dalam Service Quotas, gunakan [formulir penambahan batas](#).

Anda Akun AWS memiliki kuota berikut yang terkait dengan Izin Terverifikasi.

Topik

- [Kuota untuk sumber daya](#)
- [Kuota untuk hierarki](#)
- [Kuota untuk operasi per detik](#)

Kuota untuk sumber daya

Nama	Default	Dapat disesuaikan	Deskripsi
Toko kebijakan per Wilayah per akun	Setiap Wilayah yang didukung: 30.000	Ya	Jumlah maksimum toko polis.
Templat kebijakan per toko kebijakan	Setiap Wilayah yang didukung: 40	Ya	Jumlah maksimum templat kebijakan di toko kebijakan.
Sumber identitas per toko kebijakan	1	Tidak	Jumlah maksimum sumber identitas yang

Nama	Default	Dapat disesuai	Deskripsi
			dapat Anda tentukan untuk toko kebijakan.
Alias toko kebijakan per toko kebijakan	10	Ya	Jumlah maksimum alias toko kebijakan yang dapat Anda kaitkan dengan satu toko kebijakan.
Ukuran permintaan otorisasi ¹	1 MB	Tidak	Ukuran maksimum permintaan otorisasi.
Ukuran kebijakan	10.000 byte	Ya	Ukuran maksimum kebijakan individu.
Ukuran skema	100.000 bita	Ya	Ukuran maksimum skema toko kebijakan.
Ukuran kebijakan per sumber daya	200.000 bytes ²	Ya	Ukuran maksimum semua kebijakan yang mereferensikan sumber daya tertentu.

¹ Kuota untuk permintaan otorisasi adalah sama untuk keduanya dan [IsAuthorized](#).
[IsAuthorizedWithToken](#)

² Batas default untuk ukuran total semua kebijakan yang dicakup untuk satu sumber daya adalah 200.000 byte. Demikian pula, ukuran total semua kebijakan, di mana ruang lingkup membiarkan sumber daya tidak terdefinisi, sehingga berlaku untuk semua sumber daya, dibatasi secara default hingga 200.000 byte. Perhatikan bahwa untuk kebijakan yang ditautkan templat, ukuran templat kebijakan hanya dihitung satu kali, ditambah ukuran setiap set parameter yang digunakan untuk membuat instance setiap kebijakan yang ditautkan templat. Batas ini dapat dinaikkan, asalkan desain kebijakan Anda memenuhi batasan tertentu. Jika Anda perlu menjelajahi opsi ini, [hubungi Dukungan](#).

Contoh ukuran kebijakan terkait templat

Anda dapat menentukan bagaimana kebijakan terkait templat berkontribusi pada ukuran Kebijakan per kuota sumber daya dengan mengambil jumlah panjang pokok dan sumber daya. Jika pokok atau sumber daya tidak ditentukan, panjang potongan itu adalah 0. Jika sumber daya tidak ditentukan, ukurannya dihitung terhadap kuota "unspecified" sumber daya. Ukuran badan template itu sendiri tidak berdampak pada ukuran kebijakan.

Mari kita lihat template berikut:

```
@id("template1")
permit (
  principal in ?principal,
  action in [Action::"view", Action::"comment"],
  resource in ?resource
)
unless {
  resource.tag == "private"
};
```

Mari buat kebijakan berikut dari template itu:

```
TemplateLinkedPolicy {
  policyId: "policy1",
  templateId: "template1",
  principal: User::"alice",
  resource: Photo::"car.jpg"
}

TemplateLinkedPolicy {
  policyId: "policy2",
  templateId: "template1",
  principal: User::"bob",
  resource: Photo::"boat.jpg"
}

TemplateLinkedPolicy {
  policyId: "policy3",
  templateId: "template1",
  principal: User::"jane",
  resource: Photo::"car.jpg"
```

```
TemplateLinkedPolicy {
  policyId: "policy4",
  templateId: "template1",
  principal: User::"jane",
  resource
}
```

Sekarang, mari kita hitung ukuran kebijakan tersebut dengan menghitung karakter dalam `principal` dan `resource` untuk masing-masing kebijakan tersebut. Setiap karakter dihitung sebagai 1 byte.

Ukuran `policy1` akan menjadi panjang pokok `User::"alice"` (13) ditambah panjang sumber daya `Photo::"car.jpg"` (16). Menambahkannya kita memiliki $13 + 16 = 29$ byte.

Ukuran `policy2` akan menjadi panjang pokok `User::"bob"` (11) ditambah panjang sumber daya `Photo::"boat.jpg"` (17). Menambahkannya kita memiliki $11 + 17 = 28$ byte.

Ukuran `policy3` akan menjadi panjang pokok `User::"jane"` (12) ditambah panjang sumber daya `Photo::"car.jpg"` (16). Menambahkannya kita memiliki $12 + 16 = 28$ byte.

Ukuran `policy4` akan menjadi panjang pokok `User::"jane"` (12) ditambah panjang sumber daya (0). Menambahkannya kita memiliki $12 + 0 = 12$ byte.

Karena `policy2` merupakan satu-satunya kebijakan yang mereferensikan sumber daya `Photo::"boat.jpg"`, ukuran total sumber daya adalah 28 byte.

Karena `policy1` dan `policy3` keduanya mereferensikan sumber daya `Photo::"car.jpg"`, ukuran sumber daya total adalah $29 + 28 = 57$ byte.

Karena `policy4` merupakan satu-satunya kebijakan yang mereferensikan "unspecified" sumber daya, ukuran total sumber daya adalah 12 byte.

Kuota untuk hierarki

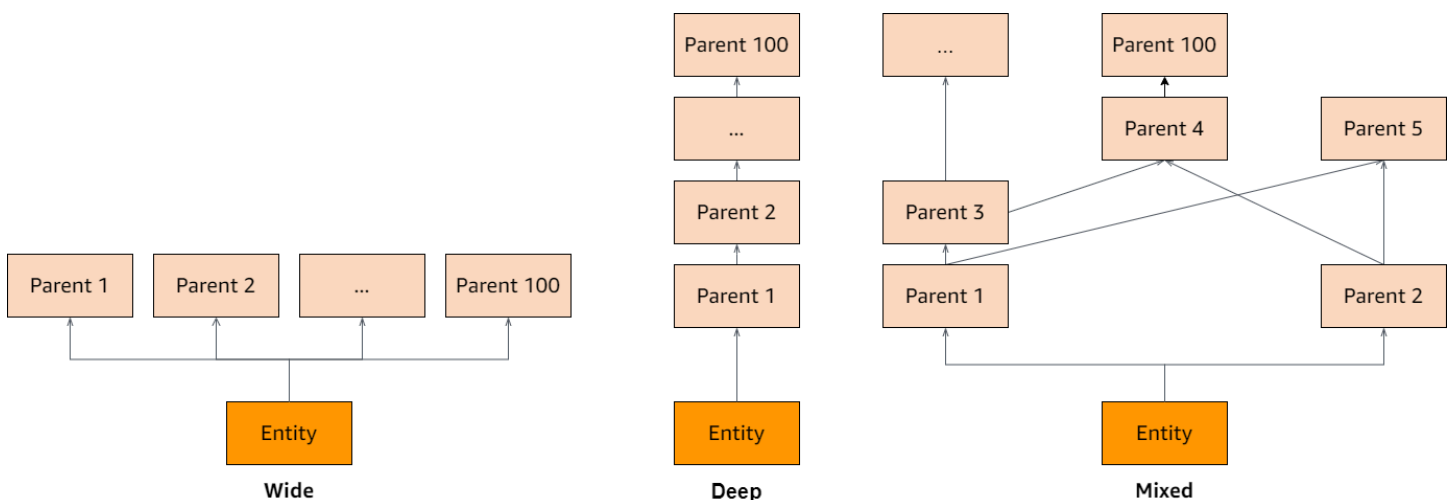
Note

Kuota berikut dikumpulkan, artinya ditambahkan bersama. Jumlah maksimum orang tua transitif untuk kelompok adalah apa yang terdaftar. Misalnya, jika batas orang tua Transitif per kepala sekolah adalah 100, itu berarti mungkin ada 100 orang tua dari kepala sekolah dan 0

orang tua untuk tindakan dan sumber daya, atau kombinasi orang tua yang menambahkan hingga 100 orang tua total.

Nama	Default	Dapat diseskan	Deskripsi
Orang tua transitif per kepala sekolah	100	Tidak	Jumlah maksimum orang tua transitif untuk setiap kepala sekolah.
Orang tua transitif per tindakan	100	Tidak	Jumlah maksimum orang tua transitif untuk setiap tindakan.
Orang tua transitif per sumber daya	100	Tidak	Jumlah maksimum orang tua transitif untuk setiap sumber daya.

Diagram di bawah ini menggambarkan bagaimana orang tua transitif dapat didefinisikan untuk suatu entitas (prinsipal, tindakan, atau sumber daya).



Kuota untuk operasi per detik

Izin Terverifikasi membatasi permintaan ke titik akhir layanan AWS Region ketika permintaan aplikasi melebihi kuota untuk operasi API. Izin Terverifikasi mungkin menampilkan pengecualian jika Anda melebihi kuota dalam permintaan per detik, atau Anda mencoba operasi penulisan simultan. Anda dapat melihat kuota RPS Anda saat ini di Service [Quotas](#). Untuk mencegah aplikasi melebihi kuota untuk suatu operasi, Anda harus mengoptimalkannya untuk percobaan ulang dan backoff eksponensial. Untuk informasi selengkapnya, lihat [Coba lagi dengan pola backoff serta Mengelola dan memantau pembatasan API](#) di beban kerja Anda.

Nama	Default	Dapat disesuai	Deskripsi
BatchGetPolicy permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum BatchGetPolicy permintaan per detik per toko kebijakan.
BatchIsAuthorized permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 30	Ya	Jumlah maksimum BatchIsAuthorized permintaan per detik per toko kebijakan.
BatchIsAuthorizedWithToken permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 30	Ya	Jumlah maksimum BatchIsAuthorizedWithToken permintaan per detik per toko kebijakan.
CreateIdentitySource permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum CreateIdentitySource permintaan per detik per toko kebijakan.
CreatePolicy permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum CreatePolicy permintaan

Nama	Default	Dapat disetujui	Deskripsi
			n per detik per toko kebijakan.
CreatePolicyStore permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum CreatePolicyStore permintaan per detik.
CreatePolicyTemplate permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum CreatePolicyTemplate permintaan per detik per toko kebijakan.
DeleteIdentitySource permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum DeleteIdentitySource permintaan per detik per toko kebijakan.
DeletePolicy permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum DeletePolicy permintaan per detik per toko kebijakan.
DeletePolicyStore permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum DeletePolicyStore permintaan per detik.
DeletePolicyTemplate permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum DeletePolicyTemplate permintaan per detik per toko kebijakan.
GetIdentitySource permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum GetIdentitySource permintaan per detik per toko kebijakan.

Nama	Default	Dapat disetujui	Deskripsi
GetPolicy permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum GetPolicy permintaan per detik per toko kebijakan.
GetPolicyStore permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum GetPolicyStore permintaan per detik.
GetPolicyTemplate permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum GetPolicyTemplate permintaan per detik per toko kebijakan.
GetSchema permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum GetSchema permintaan per detik per toko kebijakan.
IsAuthorized permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 200	Ya	Jumlah maksimum IsAuthorized permintaan per detik per toko kebijakan.
IsAuthorizedWithToken permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 200	Ya	Jumlah maksimum IsAuthorizedWithToken permintaan per detik per toko kebijakan.
ListIdentitySources permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum ListIdentitySources permintaan per detik per toko kebijakan.

Nama	Default	Dapat disetujui	Deskripsi
ListPolicies permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum ListPolicies permintaan per detik per toko kebijakan.
ListPolicyStores permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum ListPolicyStores permintaan per detik.
ListPolicyTemplates permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum ListPolicyTemplates permintaan per detik per toko kebijakan.
PutSchema permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum PutSchema permintaan per detik per toko kebijakan.
UpdateIdentitySource permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum UpdateIdentitySource permintaan per detik per toko kebijakan.
UpdatePolicy permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum UpdatePolicy permintaan per detik per toko kebijakan.
UpdatePolicyStore permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Tidak	Jumlah maksimum UpdatePolicyStore permintaan per detik.

Nama	Default	Dapat disesu an	Deskripsi
UpdatePolicyTemplate permintaan per detik per Wilayah per toko kebijakan	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum UpdatePolicyTemplate permintaan per detik per toko kebijakan.

Izin Terverifikasi Amazon dan istilah dan konsep bahasa kebijakan Cedar

Anda harus memahami konsep berikut untuk menggunakan Izin Terverifikasi Amazon.

Konsep Izin Terverifikasi

- [Model otorisasi](#)
- [Permintaan otorisasi](#)
- [Respon otorisasi](#)
- [Kebijakan yang dipertimbangkan](#)
- [Data konteks](#)
- [Menentukan kebijakan](#)
- [Data entitas](#)
- [Izin, otorisasi, dan prinsip](#)
- [Penegakan kebijakan](#)
- [Toko kebijakan](#)
- [Alias toko kebijakan](#)
- [Nama kebijakan](#)
- [Nama templat kebijakan](#)
- [Kebijakan yang memuaskan](#)
- [Perbedaan antara Izin Terverifikasi Amazon dan bahasa kebijakan Cedar](#)

Konsep bahasa kebijakan cedar

- [Otorisasi](#)
- [Entitas](#)
- [Grup dan hierarki](#)
- [Ruang nama](#)
- [Kebijakan](#)
- [Templat kebijakan](#)

- [Skema](#)

Model otorisasi

Model otorisasi menjelaskan ruang lingkup [permintaan otorisasi](#) yang dibuat oleh aplikasi dan dasar untuk mengevaluasi permintaan tersebut. Ini didefinisikan dalam hal berbagai jenis sumber daya, tindakan yang diambil pada sumber daya tersebut, dan jenis prinsip yang mengambil tindakan tersebut. Ini juga mempertimbangkan konteks di mana tindakan tersebut diambil.

Kontrol Akses Berbasis Peran (RBAC) adalah dasar evaluasi di mana peran didefinisikan dan dikaitkan dengan serangkaian izin. Peran ini kemudian dapat ditugaskan ke satu atau lebih identitas. Identitas yang ditetapkan memperoleh izin yang terkait dengan peran tersebut. Jika izin yang terkait dengan peran diubah, maka modifikasi secara otomatis memengaruhi identitas apa pun yang telah ditetapkan peran tersebut. Cedar dapat mendukung keputusan RBAC melalui penggunaan kelompok utama.

Attribute-based Access Control (ABAC) adalah dasar evaluasi di mana izin yang terkait dengan identitas ditentukan oleh atribut identitas tersebut. Cedar dapat mendukung keputusan ABAC melalui penggunaan kondisi kebijakan yang merujuk atribut prinsipal.

Bahasa kebijakan Cedar memungkinkan kombinasi RBAC dan ABAC dalam satu kebijakan dengan mengizinkan izin ditentukan untuk sekelompok pengguna, yang memiliki kondisi berbasis atribut.

Permintaan otorisasi

Permintaan otorisasi adalah permintaan yang dibuat dari Izin Terverifikasi oleh aplikasi untuk mengevaluasi serangkaian kebijakan untuk menentukan apakah prinsipal dapat melakukan tindakan pada sumber daya untuk konteks tertentu.

Respon otorisasi

Respons otorisasi adalah respons terhadap permintaan [otorisasi](#). Ini termasuk mengizinkan atau menolak keputusan, ditambah informasi tambahan, seperti kebijakan IDs yang menentukan.

Kebijakan yang dipertimbangkan

Kebijakan yang dipertimbangkan adalah serangkaian kebijakan lengkap yang dipilih oleh Izin Terverifikasi untuk dimasukkan saat mengevaluasi permintaan [otorisasi](#).

Data konteks

Data konteks adalah nilai atribut yang memberikan informasi tambahan untuk dievaluasi.

Menentukan kebijakan

Menentukan kebijakan adalah kebijakan yang menentukan [respons otorisasi](#). Misalnya, jika ada dua [kebijakan yang puas](#), di mana satu adalah penolakan dan yang lainnya adalah izin, maka kebijakan penolakan akan menjadi kebijakan penentu. Jika ada beberapa kebijakan izin yang dipenuhi dan tidak ada kebijakan larangan yang memuaskan, maka ada beberapa kebijakan penentu. Jika tidak ada kebijakan yang cocok dan tanggapannya ditolak, tidak ada kebijakan yang menentukan.

Data entitas

Data entitas adalah data tentang prinsipal, tindakan, dan sumber daya. Data entitas yang relevan untuk evaluasi kebijakan adalah keanggotaan grup sepanjang hierarki entitas dan nilai atribut prinsipal dan sumber daya.

Izin, otorisasi, dan prinsip

Izin Terverifikasi mengelola izin dan otorisasi berbutir halus dalam aplikasi kustom yang Anda buat.

Prinsipal adalah pengguna aplikasi, baik manusia atau mesin, yang memiliki identitas terikat pada pengenal seperti nama pengguna atau ID mesin. Proses otentikasi menentukan apakah prinsipal benar-benar identitas yang mereka klaim.

Terkait dengan identitas itu adalah seperangkat izin aplikasi yang menentukan apa yang diizinkan oleh prinsipal tersebut untuk dilakukan dalam aplikasi itu. Otorisasi adalah proses menilai izin tersebut untuk menentukan apakah prinsipal diizinkan untuk melakukan tindakan tertentu dalam aplikasi. Izin ini dapat dinyatakan sebagai [kebijakan](#).

Penegakan kebijakan

Penegakan kebijakan adalah proses menegakkan keputusan evaluasi dalam aplikasi di luar Izin Terverifikasi. Jika evaluasi Izin Terverifikasi mengembalikan penolakan, maka penegakan hukum akan memastikan bahwa prinsipal dicegah mengakses sumber daya.

Toko kebijakan

Toko kebijakan adalah wadah untuk kebijakan dan templat. Setiap toko berisi skema yang digunakan untuk memvalidasi kebijakan yang ditambahkan ke toko. Secara default, setiap aplikasi memiliki toko kebijakan sendiri, tetapi beberapa aplikasi dapat berbagi satu toko kebijakan. Ketika aplikasi membuat permintaan otorisasi, itu mengidentifikasi toko kebijakan yang digunakan untuk mengevaluasi permintaan itu. Toko kebijakan menyediakan cara untuk mengisolasi serangkaian kebijakan, dan oleh karena itu dapat digunakan dalam aplikasi multi-penyewa untuk memuat skema dan kebijakan untuk setiap penyewa. Satu aplikasi dapat memiliki toko kebijakan terpisah untuk setiap penyewa.

Saat mengevaluasi [permintaan otorisasi](#), Izin Terverifikasi hanya mempertimbangkan subset kebijakan di penyimpanan kebijakan yang relevan dengan permintaan tersebut. Relevansi ditentukan berdasarkan ruang lingkup kebijakan. Ruang lingkup mengidentifikasi pokok dan sumber daya spesifik yang diterapkan kebijakan, dan tindakan yang dapat dilakukan oleh prinsipal pada sumber daya. Mendefinisikan ruang lingkup membantu meningkatkan kinerja dengan mempersempit serangkaian kebijakan yang dipertimbangkan.

Alias toko kebijakan

Alias toko kebijakan adalah nama yang ramah untuk toko kebijakan. Anda dapat menggunakan alias penyimpanan kebijakan untuk mengidentifikasi penyimpanan kebijakan dalam operasi Izin Terverifikasi yang menerima parameter. `policyStoreId` Alias toko kebijakan adalah AWS sumber daya independen dengan milik mereka sendiri ARNs. Setiap alias dikaitkan dengan satu toko kebijakan pada satu waktu, dan beberapa alias dapat dikaitkan dengan toko kebijakan yang sama. Untuk informasi selengkapnya, lihat [Alias penyimpanan kebijakan Izin Terverifikasi Amazon](#).

Nama kebijakan

Nama kebijakan adalah nama ramah opsional untuk kebijakan. Nama kebijakan harus unik untuk semua kebijakan dalam toko kebijakan dan diawali dengan `name/`. Anda dapat menggunakan nama kebijakan sebagai pengganti ID kebijakan dalam operasi bidang kontrol yang menerima `policyId` parameter. Nama dapat disetel saat membuat atau memperbarui kebijakan. Hanya `GetPolicy` dan `ListPolicies` kembalikan nama di output.

Nama templat kebijakan

Nama templat kebijakan adalah nama ramah opsional untuk templat kebijakan. Nama templat kebijakan harus unik untuk semua templat kebijakan dalam penyimpanan kebijakan dan diawali dengan `name/`. Anda dapat menggunakan nama templat kebijakan sebagai pengganti ID templat kebijakan dalam operasi bidang kontrol yang menerima `policyTemplateId` parameter. Nama dapat disetel saat membuat atau memperbarui templat kebijakan. Hanya `GetPolicyTemplate` dan `ListPolicyTemplates` kembalikan nama di output.

Kebijakan yang memuaskan

Kebijakan puas adalah kebijakan yang sesuai dengan parameter [permintaan otorisasi](#).

Perbedaan antara Izin Terverifikasi Amazon dan bahasa kebijakan Cedar

Izin Terverifikasi Amazon menggunakan mesin bahasa kebijakan Cedar untuk melakukan tugas otorisasi. Namun, ada beberapa perbedaan antara implementasi Cedar asli dan implementasi Cedar yang ditemukan di Izin Terverifikasi. Topik ini mengidentifikasi perbedaan-perbedaan tersebut.

Definisi namespace

Implementasi Izin Terverifikasi dari Cedar memiliki perbedaan berikut dari implementasi Cedar asli:

- Izin Terverifikasi hanya mendukung satu [namespace dalam skema yang](#) ditentukan di penyimpanan kebijakan.
- Izin Terverifikasi tidak memungkinkan Anda membuat [namespace](#) yang berupa string kosong atau menyertakan nilai berikut: `aws`, `amazon` atau `cedar`

Dukungan template kebijakan

Baik Izin Terverifikasi dan Cedar mengizinkan placeholder dalam lingkup hanya untuk dan `principal resource`. Namun, Izin Terverifikasi juga tidak mengharuskan `principal` dan tidak `resource` dibatasi.

Kebijakan berikut berlaku di Cedar tetapi ditolak oleh Izin Terverifikasi karena tidak `principal` dibatasi.

```
permit(principal, action == Action::"view", resource == ?resource);
```

Kedua contoh berikut ini valid di Cedar dan Izin Terverifikasi karena keduanya `principal` dan `resource` memiliki kendala.

```
permit(principal == User::"alice", action == Action::"view", resource == ?resource);
```

```
permit(principal == ?principal, action == Action::"a", resource in ?resource);
```

Dukungan skema

Izin Terverifikasi mengharuskan semua nama kunci skema JSON menjadi string yang tidak kosong. Cedar memungkinkan string kosong dalam beberapa kasus, seperti untuk properti atau ruang nama.

Definisi kelompok aksi

Metode otorisasi Cedar memerlukan daftar entitas untuk dipertimbangkan ketika mengevaluasi permintaan otorisasi terhadap kebijakan.

Anda dapat menentukan tindakan dan kelompok tindakan yang digunakan oleh aplikasi Anda dalam skema. Namun, Cedar tidak menyertakan skema sebagai bagian dari permintaan evaluasi. Sebagai gantinya, Cedar menggunakan skema hanya untuk memvalidasi kebijakan dan templat kebijakan yang Anda kirimkan. Karena Cedar tidak mereferensikan skema selama permintaan evaluasi, bahkan jika Anda mendefinisikan grup tindakan dalam skema, Anda juga harus menyertakan daftar grup tindakan apa pun sebagai bagian dari daftar entitas yang harus diteruskan ke operasi API otorisasi.

Izin Terverifikasi melakukan ini untuk Anda. Setiap grup tindakan yang Anda tentukan dalam skema Anda secara otomatis ditambahkan ke daftar entitas yang Anda berikan sebagai parameter untuk operasi `IsAuthorized` atau `IsAuthorizedWithToken`.

Pemformatan entitas

Pemformatan JSON entitas dalam Izin Terverifikasi menggunakan `entityList` parameter berbeda dari Cedar dengan cara berikut:

- Dalam Izin Terverifikasi, objek JSON harus memiliki semua pasangan kunci-nilai yang dibungkus dalam objek JSON dengan nama. `Record`
- Daftar JSON di Izin Terverifikasi harus dibungkus dalam pasangan nilai kunci JSON di mana nama kuncinya `Set` dan nilainya adalah daftar JSON asli dari Cedar.

- Untuk `String`, `Long`, dan `Boolean` jenis nama, setiap pasangan kunci-nilai dari Cedar digantikan oleh objek JSON di Izin Terverifikasi. Nama objek adalah nama kunci asli. Di dalam objek JSON, ada satu pasangan kunci-nilai di mana nama kunci adalah nama tipe dari nilai skalar (`String`, `Long`, atau `Boolean`) dan nilainya adalah nilai dari entitas Cedar.
- Pemformatan sintaks entitas Cedar dan entitas Izin Terverifikasi berbeda dengan cara berikut:

Format cedar	Format Izin Terverifikasi
<code>uid</code>	<code>Identifier</code>
<code>type</code>	<code>EntityType</code>
<code>id</code>	<code>EntityId</code>
<code>attrs</code>	<code>Attributes</code>
<code>parents</code>	<code>Parents</code>

Example- Daftar

Contoh berikut menunjukkan bagaimana daftar entitas dinyatakan dalam Cedar dan Izin Terverifikasi, masing-masing.

Cedar

```
[
  {
    "number": 1
  },
  {
    "sentence": "Here is an example sentence"
  },
  {
    "Question": false
  }
]
```

Verified Permissions

```
{
```

```
"Set": [
  {
    "Record": {
      "number": {
        "Long": 1
      }
    }
  },
  {
    "Record": {
      "sentence": {
        "String": "Here is an example sentence"
      }
    }
  },
  {
    "Record": {
      "question": {
        "Boolean": false
      }
    }
  }
]
}
```

Example- Evaluasi kebijakan

Contoh berikut menunjukkan bagaimana entitas diformat untuk mengevaluasi kebijakan dalam permintaan otorisasi di Cedar dan Izin Terverifikasi, masing-masing.

Cedar

```
[
  {
    "uid": {
      "type": "PhotoApp::User",
      "id": "alice"
    },
    "attrs": {
      "age": 25,
      "name": "alice",
      "userId": "123456789012"
    }
  }
]
```

```
    },
    "parents": [
      {
        "type": "PhotoApp::UserGroup",
        "id": "alice_friends"
      },
      {
        "type": "PhotoApp::UserGroup",
        "id": "AVTeam"
      }
    ]
  },
  {
    "uid": {
      "type": "PhotoApp::Photo",
      "id": "vacationPhoto.jpg"
    },
    "attrs": {
      "private": false,
      "account": {
        "__entity": {
          "type": "PhotoApp::Account",
          "id": "ahmad"
        }
      }
    },
    "parents": []
  },
  {
    "uid": {
      "type": "PhotoApp::UserGroup",
      "id": "alice_friends"
    },
    "attrs": {},
    "parents": []
  },
  {
    "uid": {
      "type": "PhotoApp::UserGroup",
      "id": "AVTeam"
    },
    "attrs": {},
    "parents": []
  }
}
```

```
]
```

Verified Permissions

```
[
  {
    "Identifier": {
      "EntityType": "PhotoApp::User",
      "EntityId": "alice"
    },
    "Attributes": {
      "age": {
        "Long": 25
      },
      "name": {
        "String": "alice"
      },
      "userId": {
        "String": "123456789012"
      }
    },
    "Parents": [
      {
        "EntityType": "PhotoApp::UserGroup",
        "EntityId": "alice_friends"
      },
      {
        "EntityType": "PhotoApp::UserGroup",
        "EntityId": "AVTeam"
      }
    ]
  },
  {
    "Identifier": {
      "EntityType": "PhotoApp::Photo",
      "EntityId": "vacationPhoto.jpg"
    },
    "Attributes": {
      "private": {
        "Boolean": false
      },
      "account": {
        "EntityIdentifier": {
```

```

        "EntityType": "PhotoApp::Account",
        "EntityId": "ahmad"
      }
    },
    "Parents": []
  },
  {
    "Identifier": {
      "EntityType": "PhotoApp::UserGroup",
      "EntityId": "alice_friends"
    },
    "Parents": []
  },
  {
    "Identifier": {
      "EntityType": "PhotoApp::UserGroup",
      "EntityId": "AVTeam"
    },
    "Parents": []
  }
]

```

Batas panjang dan ukuran

Izin Terverifikasi mendukung penyimpanan dalam bentuk penyimpanan kebijakan untuk menyimpan skema, kebijakan, dan templat kebijakan Anda. Penyimpanan tersebut menyebabkan Izin Terverifikasi memberlakukan beberapa batas panjang dan ukuran yang tidak relevan dengan Cedar.

Objek	Batas Izin Terverifikasi (dalam byte)	Batas cedar
Ukuran kebijakan ¹	10.000	Tidak ada
Deskripsi kebijakan sebaris	150	Tidak berlaku untuk Cedar
Ukuran templat kebijakan	10.000	Tidak ada
Ukuran skema	100.000	Tidak ada
Jenis entitas	200	Tidak ada

Objek	Batas Izin Terverifikasi (dalam byte)	Batas cedar
ID Kebijakan	64	Tidak ada
ID Templat Kebijakan	64	Tidak ada
ID Entitas	200	Tidak ada
ID toko kebijakan	64	Tidak berlaku untuk Cedar

¹ Ada batasan untuk kebijakan per penyimpanan kebijakan di Izin Terverifikasi berdasarkan ukuran gabungan prinsip, tindakan, dan sumber daya kebijakan yang dibuat di toko kebijakan. Ukuran total semua kebijakan yang berkaitan dengan satu sumber daya tidak dapat melebihi 200.000 byte. Untuk kebijakan yang ditautkan templat, ukuran templat kebijakan dihitung hanya sekali, ditambah ukuran setiap set parameter yang digunakan untuk membuat instance setiap kebijakan yang ditautkan templat.

Izin Terverifikasi Amazon ditingkatkan ke Cedar 4 FAQ

Izin Terverifikasi Amazon memutakhirkan versi Cedar yang digunakannya dari versi 2 ke versi 4. Cedar adalah bahasa sumber terbuka yang Anda gunakan untuk menulis kebijakan, templat kebijakan, dan skema di toko kebijakan Anda. Dengan dukungan Cedar 4 di Izin Terverifikasi, Anda dapat menggunakan fitur baru seperti `is` operator dan tag entitas untuk menulis kebijakan yang lebih ekspresif.

Izin Terverifikasi Amazon secara otomatis memutakhirkan toko kebijakan ke Cedar 4. Namun, beberapa kebijakan, skema, dan permintaan otorisasi yang ditulis untuk Cedar 2 tidak sesuai dengan Cedar 4. Jika ini terjadi pada toko kebijakan Anda, maka kami tidak akan memutakhirkannya secara otomatis. Anda mungkin perlu membuat perubahan pada kebijakan, templat kebijakan, skema, atau kode aplikasi Anda sebelum dapat meningkatkan ke Cedar 4.

Topik

- [Mengapa beberapa kebijakan, templat kebijakan, dan skema tidak kompatibel dengan Cedar 4?](#)
- [Bagaimana cara mengetahui apakah toko kebijakan saya menggunakan Cedar 2 atau Cedar 4?](#)
- [Bagaimana cara meningkatkan ke Cedar 4?](#)
- [Dapatkah saya menurunkan versi toko polis saya dari Cedar 4 ke Cedar 2?](#)
- [Mengapa saya menerima pesan kesalahan yang mengatakan toko kebijakan saya dikonfigurasi untuk Cedar 2?](#)
- [Bagaimana cara membuat skema saya kompatibel dengan Cedar 4?](#)
- [Bagaimana cara membuat kebijakan dan template saya kompatibel dengan Cedar 4?](#)

Mengapa beberapa kebijakan, templat kebijakan, dan skema tidak kompatibel dengan Cedar 4?

Tim Cedar telah membuat beberapa perubahan yang tidak kompatibel ke belakang sejak Cedar 2, untuk memperbaiki bug dan menyederhanakan bahasa. Perubahan ini meliputi:

- perubahan sintaks untuk kebijakan, templat kebijakan, dan skema
- validator kebijakan yang lebih tepat, yang mendeteksi lebih banyak kesalahan
- perubahan perilaku fungsi bawaan seperti `isInRange`

[Untuk daftar lengkap perubahan yang tidak kompatibel ke belakang, cari item yang ditandai dengan \(*\) di changelog Cedar.](#)

Bagaimana cara mengetahui apakah toko kebijakan saya menggunakan Cedar 2 atau Cedar 4?

Anda dapat memeriksa versi Cedar yang digunakan toko kebijakan Anda menggunakan konsol Izin Terverifikasi Amazon, atau menggunakan operasi. GetPolicyStore

Note

Semua toko kebijakan di wilayah yang sama Akun AWS dan menggunakan versi Cedar yang sama.

Console

Untuk memeriksa versi Cedar dari toko kebijakan (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka konsol Izin Terverifikasi Amazon di <https://console.aws.amazon.com/verifiedpermissions/>.
2. Dari panel navigasi, pilih Toko kebijakan, lalu pilih toko kebijakan yang ingin Anda periksa.
3. Pilih Pengaturan di panel navigasi.
4. Di kotak Detail, cari bidang versi Cedar.

Bidang berbunyi CEDAR_2 jika toko kebijakan Anda menggunakan Cedar 2, dan CEDAR_4 jika menggunakan Cedar 4.

CLI

Untuk memeriksa versi Cedar dari toko kebijakan ()AWS CLI

1. Instal dan konfigurasi AWS Command Line Interface (AWS CLI), jika Anda belum melakukannya. Untuk selengkapnya, lihat [Menginstal atau memperbarui versi terbaru AWS CLI](#).
2. Gunakan perintah `get-policy-store`. Dalam contoh berikut, ganti *policy-store-id* dengan pengenalan toko kebijakan Anda:

```
aws verifiedpermissions get-policy-store \  
--policy-store-id policy-store-id
```

cedarVersionBidang dalam output menunjukkan versi Cedar mana yang digunakan oleh toko kebijakan. Contoh:

```
{  
  "policyStoreId": "ABCDEFGH12345678abcdefgh",  
  "arn": "arn:aws:verifiedpermissions::111122223333:policy-store/  
ABCDEFGH12345678abcdefgh",  
  "validationSettings": {  
    "mode": "STRICT"  
  },  
  "createdDate": "2025-06-03T13:09:47.752255+00:00",  
  "lastUpdatedDate": "2025-06-03T13:09:47.752255+00:00",  
  "deletionProtection": "ENABLED",  
  "cedarVersion": "CEDAR_2"  
}
```

Bidang berbunyi CEDAR_2 jika toko kebijakan Anda menggunakan Cedar 2, dan CEDAR_4 jika menggunakan Cedar 4.

Bagaimana cara meningkatkan ke Cedar 4?

Izin Terverifikasi Amazon telah meningkatkan sebagian besar pelanggan ke Cedar 4. Jika Anda belum pernah membuat toko kebijakan, maka setiap toko kebijakan baru yang Anda buat akan menggunakan Cedar 4. Jika Anda adalah pelanggan lama, maka kami kemungkinan telah meningkatkan Anda ke Cedar 4. Lihat [Bagaimana cara mengetahui apakah toko kebijakan saya menggunakan Cedar 2 atau Cedar 4?](#) untuk memeriksa versi Cedar mana yang digunakan toko kebijakan Anda.

Jika Anda belum ditingkatkan, Izin Terverifikasi mendeteksi kebijakan, templat kebijakan, skema, atau permintaan otorisasi di salah satu toko kebijakan Anda yang tidak kompatibel dengan Cedar 4. Kami akan mengirimkan pemberitahuan email yang menjelaskan sumber daya mana yang tidak kompatibel nanti pada tahun 2025. Untuk meningkatkan lebih cepat, buka kasing dengan Dukungan.

⚠ Important

Semua toko kebijakan dalam satu sama Akun AWS menggunakan versi Cedar yang sama. Jika satu toko kebijakan di akun Anda tidak kompatibel dengan Cedar 4, maka Anda tidak dapat menggunakan Cedar 4 di toko kebijakan mana pun di akun itu.

Dapatkah saya menurunkan versi toko polis saya dari Cedar 4 ke Cedar 2?

Tidak. Jika Anda mengalami masalah setelah toko polis Anda ditingkatkan ke Cedar 4, buka kasing dengan Dukungan.

Mengapa saya menerima pesan kesalahan yang mengatakan toko kebijakan saya dikonfigurasi untuk Cedar 2?

Beberapa fitur Izin Terverifikasi Amazon mengandalkan fitur baru di Cedar 4. Jika toko kebijakan Anda tidak menggunakan Cedar 4, Anda tidak dapat menggunakan bidang API berikut:

- Dalam `IsAuthorized`, `BatchIsAuthorized`, `IsAuthorizedWithToken` dan `BatchIsAuthorizedWithToken` operasi:
 - `datetime`, `decimal` atau `duration` nilai di `attributes` atau `context` bidang

Anda tidak dapat menggunakan sintaks atau tipe data dalam kebijakan, templat kebijakan, atau skema yang diperkenalkan setelah Cedar 2 hingga penyimpanan kebijakan Anda ditingkatkan.

Bagaimana cara membuat skema saya kompatibel dengan Cedar 4?

Konsol Izin Terverifikasi dapat secara otomatis memperbaiki beberapa masalah kompatibilitas dalam skema Anda. Jika skema Anda tidak dapat diperbaiki secara otomatis, konsol akan menampilkan daftar kesalahan untuk Anda perbaiki secara manual.

⚠ Important

Editor kode di konsol Izin Terverifikasi Amazon selalu menampilkan kesalahan dan peringatan dari Cedar 4, meskipun toko kebijakan Anda menggunakan Cedar 2. Anda dapat terus membuat pembaruan skema yang tidak kompatibel dengan Cedar 4 menggunakan tombol Simpan perubahan, atau API Izin Terverifikasi.

Untuk memperbaiki skema menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Izin Terverifikasi Amazon di izin [terverifikasi](#).
2. Dari panel navigasi, pilih Toko kebijakan, lalu pilih toko kebijakan yang ingin Anda periksa.
3. Pilih Skema di panel navigasi.
4. Jika skema Anda dapat diperbaiki secara otomatis, Anda akan melihat spanduk bertuliskan "Klik 'Perbaiki' untuk melihat pratinjau versi yang kompatibel". Pilih Perbaiki.
5. Tinjau perubahan yang dibuat pada skema Anda, dan klik Pratinjau skema yang diperbarui.
6. Tinjau skema yang diperbarui, dan klik Simpan perubahan.

Jika skema Anda tidak dapat diperbaiki secara otomatis, Anda dapat melihat daftar kesalahan untuk diperbaiki sendiri di konsol.

1. Buka halaman Edit skema seperti dijelaskan di atas.
2. Pilih mode JSON.
3. Arahkan kursor ke ikon kesalahan merah di selokan di sisi kiri editor kode. Pesan kesalahan ditampilkan di tooltip.

Berikut adalah beberapa kesalahan umum yang mungkin Anda temui dan cara mengatasinya:

gagal mengurai skema dari JSON: `` ***field-name***

Dengan Cedar 2, Anda dapat menyertakan bidang arbitrer di bagian skema seperti definisi tipe, bahkan jika mereka tidak memiliki arti apa pun sebagai bagian dari skema Cedar. Di Cedar 4, ini tidak lagi diizinkan. Untuk mengatasi kesalahan ini, hapus bidang yang dipanggil ***field-name*** dari skema JSON Anda. Untuk daftar bidang skema yang valid, lihat dokumentasi [Cedar](#).

jenis ekstensi tidak diketahui ``extension-name``

Di Cedar 2, saat Anda mendeklarasikan atribut yang type adaExtension, Anda dapat menentukan nilai apa pun untuk name bidang tersebut, apakah nilainya adalah nama jenis ekstensi yang valid atau tidak. Ini sekarang kesalahan dengan Cedar 4. Untuk mengatasinya, ganti `extension-name` dengan nama jenis ekstensi yang valid. Anda dapat menemukan daftar nama jenis ekstensi yang valid di [dokumentasi Cedar](#).

Jika Anda masih tidak yakin bagaimana mengatasi kesalahan dalam skema Anda, hubungi Dukungan

Bagaimana cara membuat kebijakan dan template saya kompatibel dengan Cedar 4?

Konsol Izin Terverifikasi menunjukkan kesalahan apa pun dalam kebijakan atau templat yang membuatnya tidak kompatibel dengan Cedar 4.

Untuk melihat kesalahan kebijakan atau templat di konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Izin Terverifikasi Amazon di izin [terverifikasi](#).
2. Dari panel navigasi, pilih Toko kebijakan, lalu pilih toko kebijakan yang ingin Anda periksa.
3. Pilih templat Kebijakan atau Kebijakan di panel navigasi, yang sesuai.
4. Pilih kebijakan atau templat yang tidak kompatibel.
5. Pilih Edit
6. Arahkan kursor ke ikon kesalahan merah di selokan di sisi kiri editor kode. Pesan kesalahan ditampilkan di tooltip.

Berikut adalah beberapa kesalahan umum yang mungkin Anda temui dan cara mengatasinya:

literal set kosong dilarang dalam kebijakan

Di Cedar 2, Anda dapat menggunakan sintaks `mySet == []` untuk memeriksa apakah satu set kosong. Dengan Cedar 4, kebijakan yang menggunakan sintaks ini tidak lagi memvalidasi skema. Ganti `mySet == []` dalam kebijakan Anda dengan `mySet.isEmpty()`.

Riwayat dokumen untuk Panduan Pengguna Izin Terverifikasi Amazon

Tabel berikut menjelaskan rilis dokumentasi untuk Izin Terverifikasi.

Perubahan	Deskripsi	Tanggal
Nama kebijakan dan nama templat kebijakan	Sekarang Anda dapat menetapkan nama ke kebijakan dan templat kebijakan untuk mereferensikannya dengan nama yang ramah.	Maret 4, 2025
Alias toko kebijakan	Anda sekarang dapat membuat alias toko kebijakan untuk merujuk ke toko kebijakan Anda dengan nama yang ramah.	Februari 26, 2025
Kebijakan AWS terkelola baru	Anda sekarang dapat menggunakan kebijakan <code>AmazonVerifiedPermissionsFullAccess</code> dan <code>AmazonVerifiedPermissionsReadOnlyAccess</code> IAM terkelola dengan Izin Terverifikasi.	Oktober 11, 2024
Sumber identitas OIDC	Anda sekarang dapat mengotorisasi pengguna dari penyedia identitas OpenID Connect (OIDC).	Juni 8, 2024
Otorisasi Batch dengan token sumber identitas	Anda sekarang dapat mengotorisasi pengguna dari kumpulan Amazon	April 5, 2024

	Cognito pengguna dalam satu permintaan BatchIsAuthorizedWithToken API.	
Membuat toko kebijakan dengan API Gateway	Sekarang Anda dapat membuat toko kebijakan dari API dan kumpulan Amazon Cognito pengguna yang ada.	April 1, 2024
Konsep dan contoh konteks	Menambahkan informasi tentang konteks dalam permintaan otorisasi dengan Izin Terverifikasi.	Februari 1, 2024
Konsep dan contoh otorisasi	Menambahkan informasi tentang permintaan otorisasi dengan Izin Terverifikasi.	Februari 1, 2024
AWS CloudFormation integrasi	Izin Terverifikasi mendukung pembuatan sumber identitas , kebijakan, penyimpanan kebijakan, dan templat kebijakan di CloudFormation.	Juni 30, 2023
Rilis awal	Rilis awal Panduan Pengguna Izin Terverifikasi Amazon	13 Juni 2023

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