aws

Panduan Pengguna

Izin Terverifikasi Amazon

Izin Terverifikasi Amazon: Panduan Pengguna

Table of Contents

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 bagi pengguna aplikasi.

Topik

Otorisasi dalam Izin Terverifikasi

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

Izin Terverifikasi adalah layanan yang memungkinkan pelanggan membuat, memelihara, dan menguji kebijakan di AWS Management Console. Izin dinyatakan menggunakan bahasa kebijakan Cedar. Aplikasi klien memanggil API otorisasi 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 pengguna yang berwenang yang dapat mengakses aplikasi, dan hanya dapat melakukan apa yang diizinkan oleh setiap pengguna. 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 2.4.

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.

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 dan kredenal OAuth 2.0. AWS Saat membuat toko kebijakan, Anda memiliki opsi untuk membuat kepala sekolah 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 API dalam skala apa pun. Saat membuat penyimpanan kebijakan, Anda memiliki opsi untuk membuat tindakan dan sumber daya dari API di API Gateway. Untuk informasi selengkapnya tentang API Gateway, lihat Panduan Pengembang API Gateway.
- 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.

AWS Management Console

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 Windows PowerShell Pengguna.

- izin terverifikasi di Referensi Perintah AWS CLI
- Izin Terverifikasi Amazon di AWS Tools for Windows PowerShell

AWS SDK

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

Untuk mempelajari lebih lanjut dan mengunduh AWS SDK, lihat Alat untuk Amazon Web Services.

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

- 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 Konstruksi CDK

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

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 kredensi 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 Billing and Cost Management. 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 AWS Support.

Syarat dan konsep Izin Terverifikasi Amazon

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 prinsipal
- Penegakan kebijakan
- Toko kebijakan
- Kebijakan yang memuaskan
- Perbedaan antara Izin Terverifikasi dan 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 ID kebijakan 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 prinsipal

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.

Kebijakan yang memuaskan

Kebijakan yang memuaskan adalah kebijakan yang sesuai dengan parameter permintaan otorisasi.

Perbedaan antara Izin Terverifikasi dan 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 satunamespace dalam skemadidefinisikan di toko kebijakan.
- Izin Terverifikasi tidak memungkinkan Anda untuk membuatnamespacedengan nilai berikut:`aws`,`amazon`, atau`cedar`.

Dukungan template kebijakan

Baik Izin Terverifikasi dan Cedar mengizinkan placeholder dalam cakupan hanya untuk`principal`dan`resource`. Namun, Izin Terverifikasi juga tidak mengharuskan`principal`dan`resource`tidak dibatasi.

Kebijakan berikut berlaku di Cedar tetapi ditolak oleh Izin Terverifikasi karena`principal`tidak 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.

Dukungan jenis ekstensi

Izin Terverifikasi mendukung Cedarjenis ekstensidalam kebijakan, tetapi saat ini tidak mendukung memasukkannya dalam definisi skema atau sebagai bagian dari`entities`parameter dari`IsAuthorized`dan`IsAuthorizedWithToken`operasi.

Jenis ekstensi termasuk titik tetap (`decimal`) dan alamat IP (`ipaddr`) tipe data.

Format Cedar JSON untuk entitas

Pada saat ini, Izin Terverifikasi mengharuskan Anda untuk meneruskan daftar entitas yang akan dipertimbangkan dalam pertanyaan ulang otorisasi menggunakan struktur yang ditentukan untukEntitiesDefinition, yang merupakan array dariEntityItemelemen. Izin Terverifikasi saat ini tidak mendukung meneruskan daftar entitas yang akan dipertimbangkan dalam permintaan otorisasi diFormat Cedar JSON. Untuk persyaratan khusus memformat entitas Anda untuk digunakan dalam Izin Terverifikasi, lihatPemformatan entitas di Izin Terverifikasi Amazon.

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 ke`IsAuthorized`atau`IsAuthorizedWithToken`operasi.

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	10.000	Tidak ada
Deskripsi kebijakan sebaris	150	Tidak berlaku untuk Cedar
Ukuran template kebijakan	10.000	Tidak ada
Ukuran skema	10.000	Tidak ada
Jenis entitas	200	Tidak ada
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.

Memulai dengan Izin Terverifikasi

Gunakan tutorial ini untuk memulai dengan Izin Terverifikasi Amazon.

Topik

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/signup.
2. Ikuti petunjuk online.

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

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. 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 mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan mengunjungi https://aws.amazon.com/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

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

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai 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.

IAM kebijakan untuk Izin Terverifikasi

Izin Terverifikasi mengelola izin pengguna dalam aplikasi Anda. Agar aplikasi Anda memanggil API Izin Terverifikasi atau agar AWS Management Console 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.

Aksi	Deskripsi
CreatePolicyStore	Tindakan untuk membuat toko kebijakan baru.
DeletePolicyStore	Tindakan untuk menghapus toko kebijakan.
ListPolicyStores	Tindakan untuk mencantumkan semua toko kebijakan di Akun AWS.

Aksi	Deskripsi
CreatePolicy	Tindakan untuk membuat kebijakan Cedar di toko kebijakan. Anda dapat membuat kebijakan statis atau kebijakan yang ditautkan ke templat kebijakan.
DeletePolicy	Tindakan untuk menghapus kebijakan dari toko kebijakan.
GetPolicy	Tindakan untuk mengambil informasi tentang kebijakan tertentu.
ListPolicies	Tindakan untuk mencantumkan semua kebijakan di toko kebijakan.
IsAuthorized	Tindakan untuk mendapatkan respons otorisasi berdasarkan parameter yang dijelaskan dalam permintaan otorisasi.

Contoh IAM kebijakan untuk izin CreatePolicy tindakan:

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "verifiedpermissions:CreatePolicy"
            ],
            "Resource": "*"
        }
    ]
 }
```

Buat toko kebijakan Izin Terverifikasi pertama Anda

Saat masuk ke konsol Izin Terverifikasi untuk pertama kalinya, Anda dapat memilih cara membuat toko kebijakan pertama dan kebijakan Cedar. Ikuti prosedur masuk yang sesuai dengan jenis

pengguna Anda seperti yang dijelaskan dalam topik Cara masuk AWS di Panduan Pengguna AWS Masuk. Di halaman Beranda Konsol, pilih layanan Izin Terverifikasi Amazon. Pilih Mulai.

Membuat toko kebijakan sampel

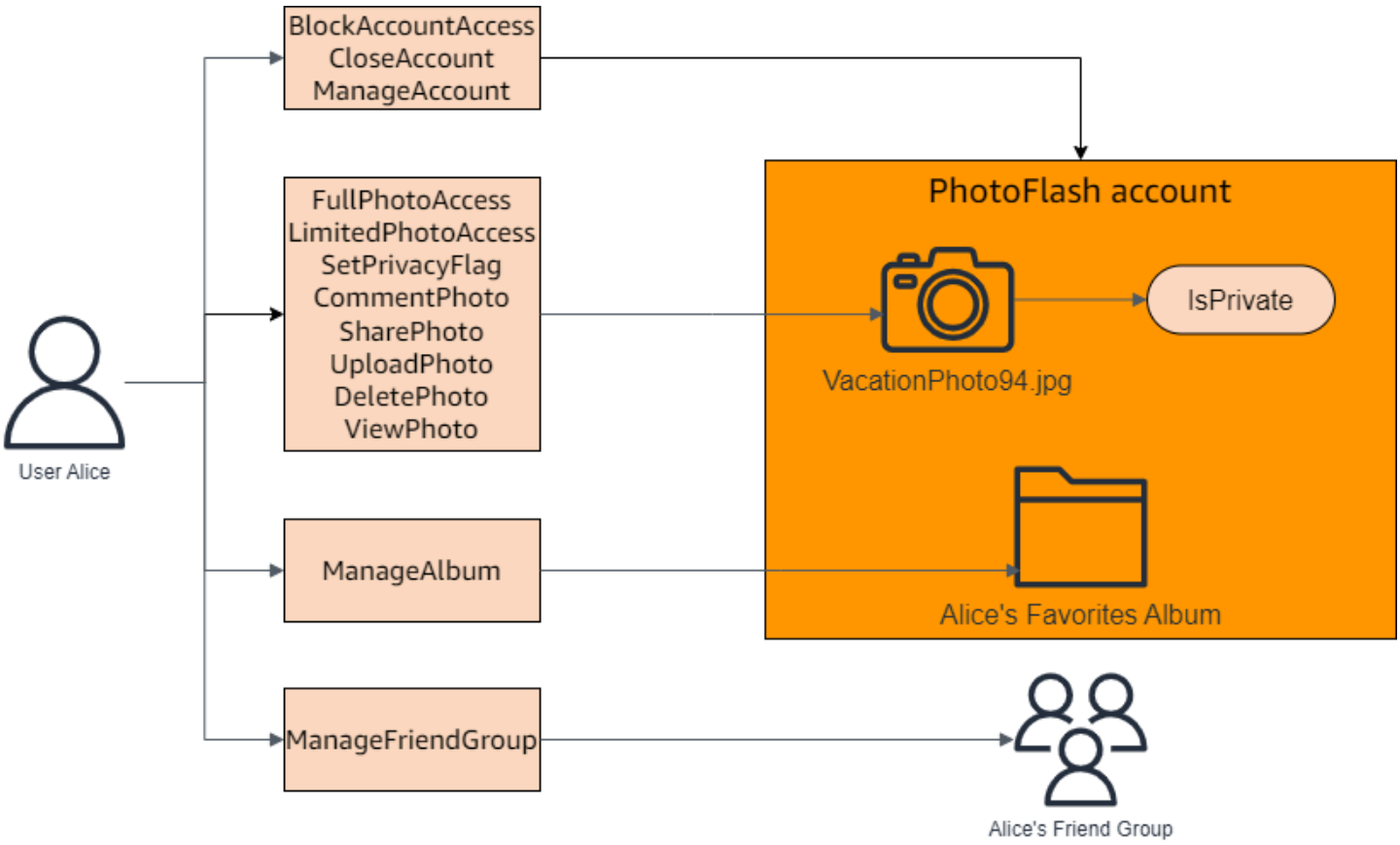
Jika ini adalah pertama kalinya Anda menggunakan Izin Terverifikasi, sebaiknya gunakan salah satu contoh toko kebijakan untuk membiasakan diri dengan cara kerja Izin Terverifikasi. Penyimpanan kebijakan sampel menyediakan kebijakan dan skema yang telah ditentukan sebelumnya.

Untuk membuat penyimpanan kebijakan menggunakan metode konfigurasi penyimpanan kebijakan Sample

1. Di konsol Izin Terverifikasi, pilih Buat toko kebijakan baru.
2. Di bagian Opsi awal, pilih Contoh penyimpanan kebijakan.
3. Di bagian proyek Contoh, pilih jenis contoh aplikasi Izin Terverifikasi yang akan digunakan. Untuk tutorial ini, pilih toko PhotoFlashkebijakan.
4. Namespace untuk skema penyimpanan kebijakan sampel Anda dibuat secara otomatis berdasarkan proyek sampel yang Anda pilih.
5. Pilih Buat toko kebijakan.

   Toko kebijakan Anda dibuat dengan kebijakan, templat kebijakan, dan skema untuk penyimpanan kebijakan sampel.

Diagram di bawah ini menggambarkan hubungan antara tindakan penyimpanan kebijakan PhotoFlash sampel dan jenis sumber daya yang diterapkan.



Membuat kebijakan yang ditautkan templat untuk penyimpanan kebijakan sampel

Penyimpanan kebijakan PhotoFlash sampel mencakup kebijakan, templat kebijakan, dan skema. Anda dapat membuat kebijakan terkait templat berdasarkan templat kebijakan yang disertakan dengan penyimpanan kebijakan sampel.

Untuk membuat kebijakan yang ditautkan templat untuk penyimpanan kebijakan sampel

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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 dengan deskripsi Berikan akses penuh ke foto bersama non-pribadi, lalu pilih Berikutnya.
5. Untuk Kepala Sekolah, masukkan`PhotoFlash::User::"Alice"`. Untuk Sumber Daya, masukkan`PhotoFlash::Album::"Bob-Vacation-Album"`.

6. Pilih Buat kebijakan terkait templat.

   Kebijakan terkait templat baru ditampilkan di bawah Kebijakan.
7. Buat kebijakan terkait template lain untuk penyimpanan kebijakan PhotoFlash sampel. Pilih Buat kebijakan, lalu pilih Buat kebijakan terkait templat.
8. Pilih tombol radio di sebelah templat kebijakan dengan deskripsi Berikan akses terbatas ke foto bersama non-pribadi, lalu pilih Berikutnya.
9. Untuk Kepala Sekolah, masukkan`PhotoFlash::FriendGroup::"MySchoolFriends"`. Untuk Sumber Daya, masukkan`PhotoFlash::Album::"Alice's favorite album"`.
10. Pilih Buat kebijakan terkait templat.

    Kebijakan terkait templat baru ditampilkan di bawah Kebijakan.

Kami akan menguji kebijakan terkait template baru di bagian tutorial selanjutnya. Untuk lebih banyak contoh nilai yang dapat Anda gunakan untuk membuat kebijakan terkait templat, lihat. PhotoFlash PhotoFlashcontoh kebijakan terkait templat

Menguji toko kebijakan sampel

Setelah membuat penyimpanan kebijakan sampel dan kebijakan terkait templat, Anda dapat menguji contoh kebijakan statis Izin Terverifikasi dan kebijakan terkait templat baru Anda dengan menjalankan permintaan otorisasi simulasi menggunakan bangku uji Izin Terverifikasi.

Bergantung pada saat Anda membuat toko kebijakan sampel, templat kebijakan Anda mungkin berbeda dari referensi dalam prosedur ini. Sebelum Anda memulai bagian tutorial ini, periksa apakah Anda memiliki setiap templat kebijakan yang mengikuti di toko kebijakan PhotoFlash contoh Anda. Jika kebijakan Anda tidak selaras dengan kebijakan ini, edit kebijakan yang ada atau buat penyimpanan kebijakan baru dari opsi PhotoFlashproyek Contoh.

Berikan akses penuh ke foto bersama non-pribadi

```
permit (
    principal in ?principal,
    action in PhotoFlash::Action::"FullPhotoAccess",
    resource in ?resource
)
when { resource.IsPrivate == false };
```

Berikan akses terbatas ke foto bersama non-pribadi

```
permit (
    principal in ?principal,
    action in PhotoFlash::Action::"LimitedPhotoAccess",
    resource in ?resource
)
when { resource.IsPrivate == false };
```

Untuk menguji kebijakan toko kebijakan sampel

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Test bench.
3. Pilih mode Visual.
4. Di bagian Principal, pilih PhotoFlash: :User dari tipe utama dalam skema Anda. Ketik pengenal untuk pengguna di kotak teks. Misalnya, `Alice`.
5. Jangan memilih Tambahkan induk untuk kepala sekolah.
6. Untuk atribut Account: Entity, pastikan entitas PhotoFlash: :Account dipilih. Ketik pengenal untuk akun. Misalnya, `Alice-account`.
7. Di bagian Sumber Daya, pilih jenis sumber daya PhotoFlash: :Photo. Ketik pengenal untuk foto di kotak teks. Misalnya, `photo.jpeg`.
8. Pilih Tambahkan induk dan pilih PhotoFlash: :Account untuk jenis entitas. Ketik pengenal yang sama untuk akun induk untuk foto yang Anda tentukan di bidang Akun: Entitas untuk pengguna. Misalnya, `Alice-account`.
9. Di bagian Tindakan, pilih PhotoFlash: :Tindakan::” ViewPhoto "dari daftar tindakan yang valid.
10. Di bagian Entitas tambahan, pilih Tambahkan entitas ini untuk menambahkan entitas akun yang disarankan.
11. Pilih Jalankan permintaan otorisasi di bagian atas halaman untuk mensimulasikan permintaan otorisasi untuk kebijakan Cedar di toko kebijakan sampel. Bangku tes harus menampilkan keputusan untuk mengizinkan permintaan.

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 terkait templat yang Anda buat di bagian sebelumnya.

Nilai pokok	Akun Utama: Nilai entitas	Nilai sumber daya	Nilai induk sumber daya	Aksi	Keputusan otorisasi
PhotoFlash: :Pengguna | Alice	PhotoFlash: :Akun | Akun Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Akun | Akun BOB	PhotoFlash: :Tindakan ::"" ViewPhoto	Menyangkal
PhotoFlash: :Pengguna | Alice	PhotoFlash: :Akun | Akun Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Akun | Akun Alice	PhotoFlash: :Tindakan ::"" ViewPhoto	Izinkan
PhotoFlash: :Pengguna | Alice	PhotoFlash: :Akun | Akun Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash: :Album | Bob-Liburan Album	PhotoFlash: :Tindakan ::"" ViewPhoto	Izinkan
PhotoFlash: :Pengguna | Alice	PhotoFlash: :Akun | Akun Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash: :Album | Bob-Liburan Album	PhotoFlash: :Tindakan ::"" DeletePhoto	Menyangkal
PhotoFlash: :Pengguna | Alice	PhotoFlash: :Akun | Akun Alice	PhotoFlash: :Foto | Bob-photo.jpeg, IsPrivate: Boolean | benar	PhotoFlash: :Album | Bob-Liburan Album	PhotoFlash: :Tindakan ::"" ViewPhoto	Menyangkal
PhotoFlash: :Pengguna | Jane, PhotoFlash:: | FriendGroup	PhotoFlash: :Akun | Akun Jane	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | Album favorit Alice	PhotoFlash: :Tindakan ::"" ViewPhoto	Izinkan

Nilai pokok	Akun Utama: Nilai entitas	Nilai sumber daya	Nilai induk sumber daya	Aksi	Keputusan otorisasi
MySchoolFriends					
PhotoFlash: :Pengguna | Jane, PhotoFlash:: | FriendGroup MySchoolFriends	PhotoFlash: :Akun | Akun Jane	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | Album favorit Alice	PhotoFlash: :Tindakan ::"" DeletePhoto	Menyangkal

Membuat toko kebijakan dengan API dan penyedia identitas yang terhubung

Kasus penggunaan umum untuk Izin Terverifikasi Amazon adalah untuk mengotorisasi permintaan dari klien aplikasi ke API back-end. AWS memiliki layanan untuk otentikasi pengguna aplikasi: Amazon Cognito. AWS juga memiliki layanan untuk API yang dihosting aman: Amazon API Gateway. Saat menggabungkan penyimpanan kebijakan Izin Terverifikasi dengan keduanya Layanan AWS, Anda dapat menghubungkan autentikasi kumpulan pengguna dan otorisasi API di aplikasi Anda dengan serangkaian kebijakan yang konsisten dan terpusat. Toko kebijakan Izin Terverifikasi memiliki dukungan bawaan untuk sumber identitas kumpulan pengguna Amazon Cognito dan API Gateway API.

Untuk membuat penyimpanan kebijakan yang ditautkan ke kumpulan pengguna dan API yang ada, pilih Siapkan dengan Cognito dan API Gateway saat Anda membuat penyimpanan kebijakan baru.

Toko kebijakan terkait API secara otomatis menyediakan model otorisasi dan sumber daya Anda untuk permintaan otorisasi. Proses pembuatan Penyiapan dengan Cognito dan API Gateway menghasilkan penyimpanan kebijakan dengan sumber identitas kumpulan pengguna, dan otorisasi Lambda yang menghubungkan API Gateway ke Izin Terverifikasi. Awalnya, 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. Misalnya, Izin Terverifikasi hanya dapat memberikan akses kepada pengguna yang memiliki `email` atribut di domain`mycompany.co.uk`.

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

Untuk mempelajari informasi lebih lanjut, lihat Toko kebijakan terkait API.

Toko kebijakan Izin Terverifikasi Amazon

Toko kebijakan adalah wadah untuk kebijakan dan templat kebijakan. Setiap toko kebijakan berisi skema yang digunakan untuk memvalidasi kebijakan yang ditambahkan ke penyimpanan kebijakan. 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.

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. Izin Terverifikasi mendukung satu namespace per toko kebijakan. Untuk informasi selengkapnya, lihat Ruang nama dalam Panduan Referensi bahasa kebijakan Cedar.

Topik

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 serta sumber daya Anda dari API Amazon API Gateway. Kami merekomendasikan opsi ini jika Anda ingin aplikasi Anda mengotorisasi permintaan API dengan keanggotaan grup pengguna.
- 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

Wizard penyiapan terpandu menuntun 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 ke fungsi toko kebijakan saat ini, misalnya Pembaruan cuaca.
4. Di bagian Detail, ketik Namespace untuk skema Anda.
5. Pilih Selanjutnya.
6. Pada jendela Jenis sumber daya, ketikkan nama untuk jenis sumber daya Anda.
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 Wajib. Izin Terverifikasi menggunakan nilai atribut yang ditentukan saat memverifikasi kebijakan terhadap skema. 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. 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.
10. Pilih Selanjutnya.
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.

Ketik nama Atribut dan pilih tipe Atribut untuk setiap atribut prinicpal. Izin Terverifikasi menggunakan nilai atribut yang ditentukan saat memverifikasi kebijakan terhadap skema. Untuk menghapus atribut yang telah ditambahkan untuk tipe prinicpal, 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 Wilayah AWSdan ketik ID kumpulan pengguna dari kumpulan pengguna Amazon Cognito untuk disambungkan. Pilih Hubungkan. Untuk informasi selengkapnya, lihat Otorisasi dengan Izin Terverifikasi Amazon di Panduan Pengembang Amazon Cognito.

12. Pilih Selanjutnya.
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.
    - 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.
    - 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 Mengatur dengan API Gateway dan metode konfigurasi sumber identitas

Opsi API Gateway mengamankan API 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 Mengatur dengan API Gateway dan sumber identitas, lalu 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 dan pilih Impor API.

    d. Ketika Anda puas dengan pilihan Anda, pilih Berikutnya.
4. Di Sumber identitas, pilih jenis penyedia Identitas. Anda dapat memilih kumpulan pengguna Amazon Cognito atau tipe iDP OpenID Connect (OIDC).
5. Jika Anda memilih Amazon Cognito:

    a. Pilih kumpulan pengguna yang sama Wilayah AWS 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 aplikasi Amazon Cognito di 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 ID klien aplikasi yang diharapkan. Untuk menerima pengguna yang melakukan autentikasi dengan kumpulan pengguna, pilih Jangan memvalidasi ID klien aplikasi.

d. Pilih Selanjutnya.

6. Jika Anda memilih penyedia OIDC:

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 Bekerja dengan sumber identitas dalam skema dan kebijakan.

c. Dalam klaim Token, pilih cara Anda ingin menyiapkan atribut pengguna di toko kebijakan Anda. Atribut ini menentukan klaim yang dapat dirujuk oleh kebijakan Anda.

   i. Pilih sumber Klaim.

      A. Untuk memberikan contoh token, pilih Ekstrak dari payload JWT dan tempel muatan JWT dari jenis Token pilihan Anda. JWT berisi header, payload, dan tanda tangan. Sampel JWT Anda harus didekodekan dan hanya muatan. Untuk mengurai payload, pilih Extract.

      B. Untuk memasukkan set atribut Anda sendiri, pilih Masukkan klaim secara manual.

   ii. Masukkan atau konfirmasikan setiap nama klaim Token dan jenis nilai Klaim yang ingin Anda tambahkan ke atribut prinsipal pengguna atau konteks tindakan dalam skema Anda.

d. Dalam klaim Pengguna dan grup, pilih klaim Pengguna untuk sumber identitas. Ini 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 jenis pengguna di toko kebijakan Anda.

e. Dalam klaim Pengguna dan grup, pilih klaim Grup 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.

f. Dalam validasi Audiens atau ID Klien, masukkan ID klien atau URL audiens yang ingin Anda terima oleh toko kebijakan dalam permintaan otorisasi, jika ada. Untuk token akses, masukkan nilai klaim audiens sepertihttps://myapp.example.com. Untuk token ID, masukkan ID klien seperti1example23456789.

g. Pilih Selanjutnya.

7. Jika Anda memilih Amazon Cognito, Izin Terverifikasi akan 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 Selanjutnya.

8. Di Deploy integrasi aplikasi, tinjau langkah-langkah yang akan diambil Izin Terverifikasi untuk membuat toko kebijakan dan otorisasi Lambda Anda.

9. Saat Anda siap membuat sumber daya baru, pilih Buat dan terapkan.

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, konfigurasikan 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 Lambda API Gateway 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. Menambahkan dan menyempurnakan 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.

   - PhotoFlashadalah 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.
   - DigitalPetToko 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**
>
> DigitalPetToko kebijakan contoh Store tidak menyertakan templat kebijakan. Toko kebijakan PhotoFlashdan TinyTodocontoh menyertakan templat kebijakan.

- TinyTodoadalah 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. Contoh kebijakan terkait templat untuk penyimpanan kebijakan contoh Izin Terverifikasi

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, lihatSkema toko kebijakan Izin Terverifikasi Amazon.

Untuk informasi selengkapnya tentang membuat kebijakan untuk toko kebijakan Anda, lihat Membuat kebijakan statis Izin Terverifikasi Amazon danMembuat kebijakan yang ditautkan templat.

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, lihatSkema toko kebijakan Izin Terverifikasi Amazon.
> - Untuk menambahkan kebijakan, lihatMembuat kebijakan statis Izin Terverifikasi Amazon.
> - Untuk menambahkan templat kebijakan, lihatMembuat template kebijakan.

```
$ aws verifiedpermissions create-policy-store \
    --validation-settings "mode=STRICT"
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T17:41:29.103459+00:00",
    "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

AWS SDKs

Anda dapat membuat toko kebijakan menggunakan `CreatePolicyStore` API. Untuk informasi selengkapnya, lihat CreatePolicyMenyimpan di Panduan Referensi API Izin Terverifikasi Amazon.

Toko kebijakan terkait API

Saat membuat penyimpanan kebijakan baru di konsol Izin Terverifikasi Amazon, Anda dapat memilih opsi Mengatur dengan API Gateway dan sumber identitas. Dengan opsi ini, Anda membuat toko kebijakan terkait API, model otorisasi untuk aplikasi yang mengautentikasi dengan kumpulan pengguna Amazon Cognito atau penyedia identitas OIDC (iDP) dan mendapatkan data dari Amazon API Gateway API. Untuk memulai, lihat Membuat toko kebijakan dengan API dan penyedia identitas yang terhubung.

Topik

- Bagaimana Izin Terverifikasi mengotorisasi permintaan API
- Menambahkan kontrol akses berbasis atribut (ABAC)

- Pertimbangan untuk toko kebijakan terkait API
- Pemecahan masalah toko kebijakan terkait API

> **Important**
>
> Penyimpanan kebijakan yang Anda buat dengan Pengaturan dengan API Gateway dan opsi 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 Menerapkan Izin Terverifikasi untuk produksi secara terprogram dengan AWS Cloud Development Kit (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 IsAuthorizedWithTokenAPI. 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.

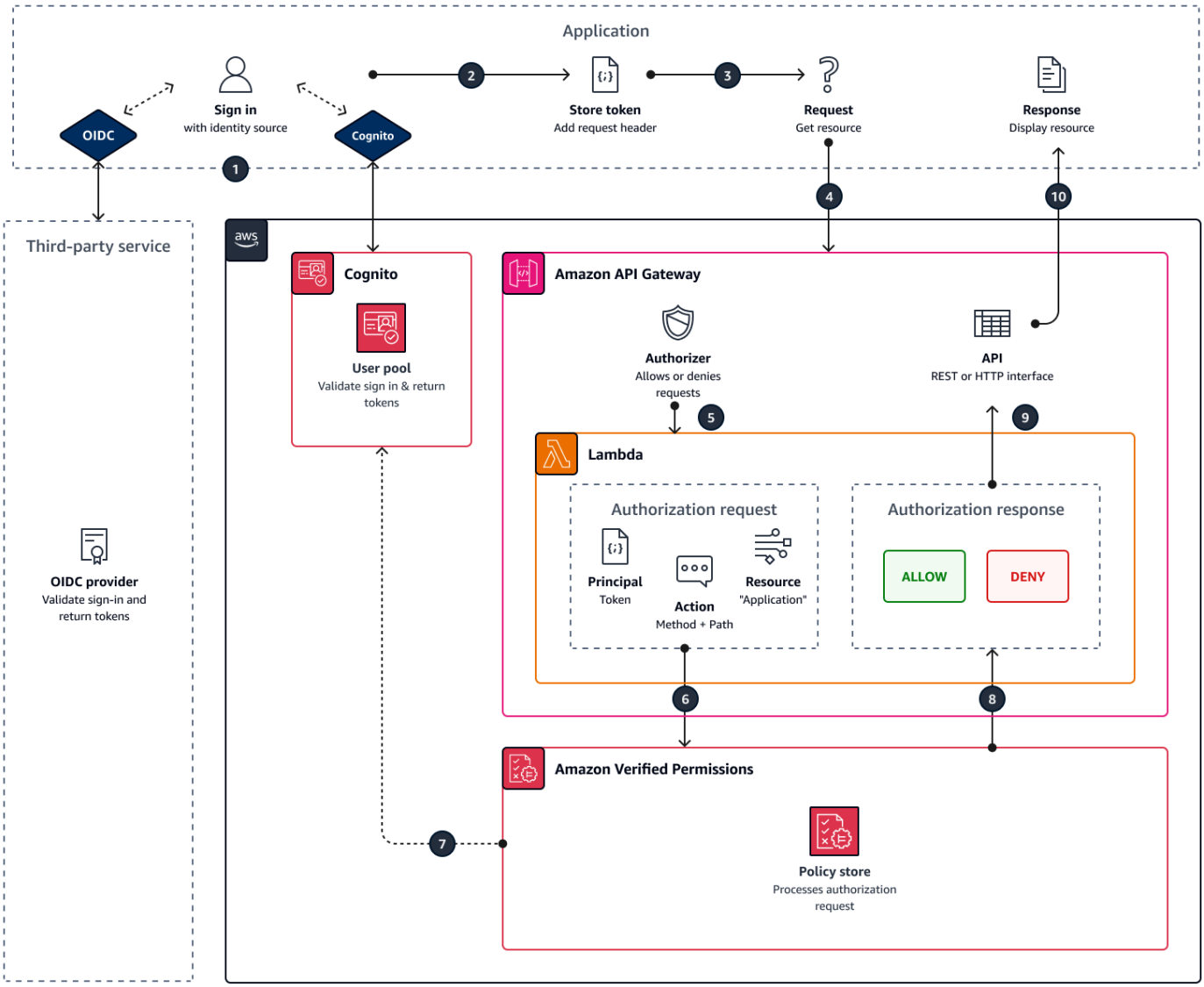
Komponen sumber identitas dan otorisasi API Gateway dengan Izin Terverifikasi

- Kumpulan pengguna Amazon Cognito 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 REST API Gateway. 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 Pengaturan dengan Cognito dan API Gateway secara otomatis membuat otorisasi Lambda ini untuk Anda.

- Toko kebijakan Izin Terverifikasi. Sumber identitas toko kebijakan adalah kumpulan pengguna 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 Cognito dan API Gateway, Izin Terverifikasi akan membuat skema dan kebijakan penyimpanan kebijakan. Skema dan kebijakan mencerminkan tindakan API dan grup kumpulan pengguna yang ingin Anda otorisasi untuk mengambil tindakan. Izin Terverifikasi juga menciptakan fungsi dan otorisasi Lambda. Anda harus mengonfigurasi otorisasi baru pada metode di API Anda.



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 JWT. Untuk informasi selengkapnya, lihat Menggunakan token dengan kumpulan pengguna di Panduan Pengembang Amazon Cognito..
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 token Amazon Cognito 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. Authorizer Lambda mengembalikan `Deny` respons `Allow` atau ke API Gateway.
10. API mengembalikan data atau `ACCESS_DENIED` respons terhadap aplikasi Anda. Aplikasi Anda memproses dan menampilkan hasil permintaan API.

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 di Amazon Cognito, lihat Menggunakan token dengan kumpulan pengguna di Panduan Pengembang Amazon Cognito.

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 tingkat kontrol akses berbutir halus tertinggi. 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 untuk 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 Bekerja dengan sumber identitas dalam skema dan kebijakan misalnya kebijakan yang memproses klaim dalam ID dan token akses.

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 cache tanggapan

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.

Grup Amazon Cognito 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 pengenal 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: Izin Terverifikasi hanya akan menanyakan API saat Anda membuat toko 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 tidak terhubung ke VPC. Secara default. API yang memiliki akses jaringan terbatas pada VPC pribadi tidak 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 AWS 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 kredensyal ini dengan administrator lain di organisasi Anda.

Lihat Pindah ke produksi dengan AWS CloudFormation ikhtisar sumber daya yang dibuat oleh Izin Terverifikasi.

Pindah ke produksi dengan AWS CloudFormation

Penyimpanan kebijakan terkait API adalah cara untuk membuat model otorisasi untuk API Gateway API dengan cepat. 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 mengekspor 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::IAM: :Role, AWS::IAM: :Policy, AWS::ApiGateway: :Authorizer, AWS::Lambda::Permission: Salin templat dari tab Template tumpukan yang digunakan Izin Terverifikasi saat Anda membuat toko kebijakan.

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 Wilayah AWS 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 konsol API Gateway, sumber daya tambahan untuk penyimpanan kebijakan Anda mungkin gagal atau belum diterapkan. Toko kebijakan terkait API menyebarkan sumber daya ini dalam tumpukan. AWS 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, lihatBekerja dengan sumber identitas dalam skema dan kebijakan.

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 default`Application`.

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.

Mengganti toko kebijakan Izin Terverifikasi

AWS Management Console

Untuk mengganti toko kebijakan atau membuat toko kebijakan tambahan

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih beralih di samping Penyimpanan kebijakan saat ini.
3. Anda dapat beralih di antara toko kebijakan yang ada atau membuat toko kebijakan tambahan.
   - Untuk beralih penyimpanan kebijakan, pilih ID penyimpanan kebijakan dari toko kebijakan untuk beralih ke.
   - Untuk membuat toko kebijakan baru, pilih Buat toko kebijakan baru. Ikuti petunjuk dalam Membuat toko kebijakan Izin Terverifikasi.

AWS CLI

Untuk mengganti toko kebijakan atau membuat toko kebijakan tambahan

AWS CLIItu tidak mempertahankan penyimpanan kebijakan “default”. Sebagai gantinya, sebagian besar AWS CLI perintah menggunakan `--policy-store-id` untuk menentukan penyimpanan kebijakan mana yang akan digunakan untuk setiap perintah.

Untuk membuat toko kebijakan baru, gunakan create-policy-storeperintah.

Menghapus toko kebijakan Izin Terverifikasi

AWS Management Console

Untuk menghapus toko kebijakan

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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.

AWS CLI

Untuk menghapus toko kebijakan

Anda dapat menghapus toko kebijakan dengan menggunakan `delete-policy-store` operasi.

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

Perintah ini tidak menghasilkan output jika berhasil.

Skema toko kebijakan Izin Terverifikasi Amazon

Skema adalah deklarasi struktur tipe entitas yang didukung oleh aplikasi Anda, dan tindakan yang mungkin diberikan aplikasi Anda dalam permintaan otorisasi.

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.

AWS Management Console

Untuk membuat skema

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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": "PSEXAMPLEabcdefg111111",
    "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 PutSchemadi Panduan Referensi API Izin Terverifikasi Amazon.

Mengedit skema dalam mode Visual

Saat Anda memilih Skema di konsol Izin Terverifikasi, mode Visual menampilkan jenis dan Tindakan Entitas yang membentuk skema Anda. Pada tampilan tingkat atas ini atau dari dalam detail entitas apa pun, Anda dapat memilih Edit skema untuk mulai membuat pembaruan pada skema Anda. Mode visual tidak tersedia dengan beberapa format skema seperti catatan bersarang.

Editor skema visual dimulai dengan serangkaian diagram yang menggambarkan hubungan antara entitas dalam skema Anda.Choose Expand untuk memaksimalkan tampilan Anda tentang hubungan entitas skema Anda.

Diagram tindakan

Tampilan diagram Tindakan mencantumkan jenis Prinsipal yang telah Anda konfigurasikan di toko kebijakan, Tindakan yang memenuhi syarat untuk dilakukan, dan Sumber Daya yang 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 di https://console.aws.amazon.com/verifiedpermissions/. 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 Wajib. 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 Wajib. 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.

Mengedit skema dalam mode JSON

Untuk mengedit skema Izin Terverifikasi dalam mode JSON

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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.

Menghapus skema

AWS Management Console

Untuk menghapus skema Izin Terverifikasi

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. Pilih toko polis Anda.
2. Di panel navigasi di sebelah kiri, pilih Skema.
3. Pilih Hapus skema.

AWS CLI

Untuk menghapus skema Izin Terverifikasi

Tidak ada perintah hapus skema. Anda dapat menghapus skema di penyimpanan kebijakan dengan menggunakan `put-schema` perintah dengan skema kosong di `cedarJson` bidang. Skema kosong diwakili oleh sepasang kurawal kurawal '{}'.

```
$ aws verifiedpermissions put-schema \
    --policy-store-id PSEXAMPLEabcdefg111111 \
    --definition cedarJson='{}'{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "namespaces": [],
    "createdDate": "2023-06-14T21:55:27.347581Z",
    "lastUpdatedDate": "2023-06-19T17:55:04.95944Z"
}
```

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 permintaan jika validasi gagal.

AWS Management Console

Untuk menyetel mode validasi kebijakan untuk penyimpanan kebijakan

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. Pilih toko polis Anda.
2. Pilih Pengaturan.
3. Di bagian Mode validasi kebijakan, pilih Ubah.
4. Lakukan salah satu dari cara berikut:
   - Untuk mengaktifkan validasi kebijakan dan menegakkan 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 UpdatePolicyStoreoperasi dan menentukan nilai parameter yang berbeda. ValidationSettings

```
$ aws verifiedpermissions update-policy-store \
```

```
    --validation-settings "mode=OFF",
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-17T18:36:10.134448+00:00",
    "lastUpdatedDate": "2023-05-17T18:36:10.134448+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "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 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.

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 nilai seperti pengidentifikasi unik universal (UUID) 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 pengenal 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: prinsipal == Pengguna: :"a1b2c3d4-e5f6-a1b2-c3d4-example11111",//alice
- 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.

Pemformatan entitas di Izin Terverifikasi Amazon

Izin Terverifikasi Amazon menggunakan bahasa kebijakan Cedar untuk membuat kebijakan. Sintaks kebijakan dan tipe data yang didukung cocok dengan sintaks dan tipe data yang diuraikan dalam konstruksi kebijakan dasar di tipe Cedar dan Data yang didukung oleh topik Cedar dalam Panduan Referensi bahasa kebijakan Cedar. Namun, ada perbedaan antara Izin Terverifikasi dan Cedar dalam pemformatan entitas saat membuat permintaan otorisasi.

Pemformatan JSON entitas dalam Izin Terverifikasi 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
`uid`	`Identifier`
`type`	`EntityType`
`id`	`EntityId`
`attrs`	`Attributes`
`parents`	`Parents`

Contoh berikut menunjukkan bagaimana entitas dalam daftar diformat menggunakan Cedar.

```
[
  {
    "number": 1
  },
```

```
  {
    "sentence": "Here is an example sentence"
  },
  {
    "Question": false
  }
]
```

Contoh berikut menunjukkan bagaimana entitas yang sama dari contoh daftar Cedar sebelumnya diformat dalam Izin Terverifikasi.

```
{
  "Set": [
    {
      "Record": {
        "number": {
          "Long": 1
        }
      }
    },
    {
      "Record": {
        "sentence": {
          "String": "Here is an example sentence"
        }
      }
    },
    {
      "Record": {
        "question": {
          "Boolean": false
        }
      }
    }
  ]
}
```

Contoh berikut menunjukkan bagaimana entitas Cedar diformat untuk mengevaluasi kebijakan dalam permintaan otorisasi.

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

Contoh berikut menunjukkan bagaimana entitas yang sama dari contoh Cedar sebelumnya diformat dalam Izin Terverifikasi.

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

Membuat kebijakan statis Izin Terverifikasi Amazon

Anda dapat membuat kebijakan statis Cedar untuk mengizinkan atau menolak prinsipal melakukan tindakan tertentu pada sumber daya tertentu untuk aplikasi Anda.

AWS Management Console

Untuk membuat kebijakan statis

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. Pilih toko polis Anda.

2. Pada panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih Buat kebijakan dan kemudian pilih Buat kebijakan statis.
4. Di bagian Efek kebijakan, pilih apakah kebijakan akan mengizinkan atau melarang ketika permintaan cocok dengan kebijakan.
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 untuk 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 Selanjutnya.
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 tersebut.
11. Pilih Buat kebijakan.

AWS CLI

Untuk membuat kebijakan statis

Anda dapat membuat kebijakan statis dengan menggunakan CreatePolicyoperasi. 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"
}
```

Mengedit kebijakan statis Izin Terverifikasi Amazon

Anda dapat mengedit kebijakan statis Cedar yang ada di toko kebijakan Anda. Anda dapat langsung memperbarui hanya kebijakan statis. Anda hanya dapat mengubah elemen tertentu dari kebijakan statis:

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

Anda tidak dapat mengubah elemen kebijakan statis ini:

- Mengubah kebijakan dari kebijakan statis menjadi kebijakan yang ditautkan templat.
- Mengubah efek kebijakan statis dari `permit` ataufor`bid`.

- Yang `principal` direferensikan oleh kebijakan statis.
- Yang `resource` direferensikan oleh kebijakan statis.

Untuk mengubah kebijakan yang ditautkan template, Anda harus memperbarui template sebagai gantinya. Untuk informasi selengkapnya, lihat Mengedit templat kebijakan.

AWS Management Console

Untuk mengedit kebijakan statis

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. Pilih toko polis Anda.
2. Pada panel navigasi di sebelah kiri, pilih Kebijakan.
3. Pilih tombol radio di sebelah kebijakan statis untuk 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 UpdatePolicyoperasi. 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 mereferensikan 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"
    }
}
```

Melihat kebijakan

AWS Management Console

Untuk melihat kebijakan Izin Terverifikasi

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. Pilih toko polis Anda.
2. Pada panel navigasi di sebelah kiri, pilih Kebijakan. Semua kebijakan yang Anda buat akan ditampilkan.
3. Pilih kotak teks Pencarian untuk memfilter kebijakan berdasarkan Prinsipal atau Sumber Daya.

4. Pilih tombol radio di samping kebijakan untuk menampilkan detail tentang kebijakan, seperti saat kebijakan dibuat, diperbarui, dan konten kebijakan.
5. Anda dapat menghapus kebijakan dengan memilih tombol radio di samping kebijakan, lalu memilih Hapus. Pilih Hapus kebijakan untuk mengonfirmasi penghapusan kebijakan.

AWS CLI

Untuk mencantumkan semua kebijakan yang tersedia di toko kebijakan

Anda dapat melihat daftar kebijakan dengan menggunakan GetPolicyoperasi. Contoh berikut mengambil daftar yang menyertakan kebijakan statis dan kebijakan terkait template.

```
$ aws verifiedpermissions list-policies \
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "Policies": [
        {
            "createdDate": "2023-05-17T18:38:31.359864+00:00",
            "definition": {
                "static": {
                    "Description": "Grant everyone of janeFriends UserGroup access
 to the vacationFolder Album"
                }
            },
            "lastUpdatedDate": "2023-05-18T16:15:04.366237+00:00",
            "policyId": "SPEXAMPLEabcdefg111111",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "STATIC",
            "resource": {
                "entityId": "publicFolder",
                "entityType": "Album"
            }
        },
        {
            "createdDate": "2023-05-22T18:57:53.298278+00:00",
            "definition": {
                "templateLinked": {
                    "policyTemplateId": "PTEXAMPLEabcdefg111111"
                }
            },
            "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",
            "policyId": "TPEXAMPLEabcdefg111111",
```

```
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "TEMPLATELINKED",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
            },
            "resource": {
                "entityId": "VacationPhoto94.jpg",
                "entityType": "Photo"
            }
        }
    ]
}
```

Untuk melihat detail untuk kebijakan individu

Anda dapat mengambil detail untuk kebijakan dengan menggunakan GetPolicyoperasi. Contoh berikut mengambil detail untuk kebijakan yang ditautkan templat.

```
$ aws verifiedpermissions get-policy \
    --policy-id TPEXAMPLEabcdefg111111
    --policy-store-id PSEXAMPLEabcdefg111111

{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEabcdefg111111/
TPEXAMPLEabcdefg111111",
    "createdDate": "2023-03-15T16:03:07.620867Z",
    "lastUpdatedDate": "2023-03-15T16:03:07.620867Z",
    "policyDefinition": {
        "templatedPolicy": {
            "policyTemplateId": "PTEXAMPLEabcdefg111111",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
            },
            "resource": {
                "entityId": "Vacation94.jpg",
                "entityType": "Photo"
            }
        }
    },
    "policyId": "TPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
```

```
    "policyType": "TEMPLATELINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
    },
    "resource": {
        "entityId": "Vacation94.jpg",
        "entityType": "Photo"
    }
}
```

Kebijakan contoh Izin Terverifikasi Amazon

Contoh kebijakan Izin Terverifikasi berikut didasarkan pada skema yang ditentukan untuk aplikasi hipotetis yang disebut PhotoFlash dijelaskan di bagian skema Contoh dari Panduan Referensi bahasa kebijakan Cedar. 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

Memungkinkan akses ke entitas individu

Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan pengguna `alice` untuk melihat foto`VacationPhoto94.jpg`.

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource == Photo::"VacationPhoto94.jpg"
);
```

Memungkinkan akses ke grup entitas

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan siapa pun dalam grup `alice_friends` untuk melihat foto`VacationPhoto94.jpg`.

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

Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan pengguna `alice` untuk melihat foto apa pun di album`alice_vacation`.

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource in Album::"alice_vacation"
);
```

Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan pengguna `alice` untuk melihat, mengedit, atau menghapus foto apa pun di album`alice_vacation`.

```
permit(
  principal == User::"alice",
  action in [Action::"view", Action::"edit", Action::"delete"],
  resource in Album::"alice_vacation"
);
```

Contoh ini menunjukkan cara Anda membuat kebijakan yang mengizinkan izin bagi pengguna `alice` di album`alice_vacation`, di mana `admin` grup ditentukan dalam hierarki skema yang berisi izin untuk melihat, mengedit, dan menghapus foto.

```
permit(
  principal == User::"alice",
  action in PhotoflashRole::"admin",
  resource in Album::"alice_vacation"
);
```

Contoh ini menunjukkan cara Anda membuat kebijakan yang mengizinkan izin bagi pengguna di album`alice_vacation`, `alice` di mana `viewer` grup ditentukan dalam hierarki skema yang berisi izin untuk melihat dan mengomentari foto. Pengguna juga `alice` diberikan `edit` izin oleh tindakan kedua yang tercantum dalam kebijakan.

```
permit(
  principal == User::"alice",
  action in [PhotoflashRole::"viewer", Action::"edit"],
  resource in Album::"alice_vacation"
)
```

Memungkinkan akses untuk entitas apa pun

Contoh ini menunjukkan bagaimana Anda dapat membuat kebijakan yang memungkinkan prinsipal yang diautentikasi untuk melihat album`alice_vacation`.

```
permit(
  principal,
  action == Action::"view",
  resource in Album::"alice_vacation"
);
```

Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan pengguna `alice` mencantumkan semua album di `jane` akun, mencantumkan foto di setiap album, dan melihat foto di akun.

```
permit(
  principal == User::"alice",
  action in [Action::"listAlbums", Action::"listPhotos", Action::"view"],
  resource in Account::"jane"
);
```

Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan pengguna `alice` melakukan tindakan apa pun pada sumber daya di album`jane_vaction`.

```
permit(
  principal == User::"alice",
  action,
  resource in Album::"jane_vacation"
);
```

Memungkinkan akses untuk atribut entitas (ABAC)

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Izin Terverifikasi memungkinkan atribut dilampirkan ke prinsip, tindakan, dan sumber daya. Atribut-atribut ini kemudian dapat direferensikan dalam `when` dan `unless` klausul kebijakan yang mengevaluasi atribut prinsip, tindakan, dan sumber daya yang membentuk konteks permintaan.

Contoh berikut menggunakan atribut yang didefinisikan dalam aplikasi hipotetis yang disebut PhotoFlash dijelaskan di bagian skema Contoh dari Panduan Referensi bahasa kebijakan Cedar.

Contoh ini menunjukkan bagaimana Anda dapat membuat kebijakan yang memungkinkan kepala sekolah mana pun di `HardwareEngineering` departemen dengan tingkat pekerjaan lebih besar dari atau sama dengan 5 untuk melihat dan mencantumkan foto di album`device_prototypes`.

```
permit(
  principal,
  action in [Action::"listPhotos", Action::"view"],
  resource in Album::"device_prototypes"
)
when {
  principal.department == "HardwareEngineering" &&
  principal.jobLevel >= 5
};
```

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan pengguna `alice` untuk melihat sumber daya jenis file apa pun`JPEG`.

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource
)
when {
  resource.fileType == "JPEG"
};
```

Tindakan memiliki atribut konteks. Anda harus meneruskan atribut ini dalam `context` permintaan otorisasi. Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan pengguna `alice` melakukan `readOnly` tindakan apa pun. Anda juga dapat mengatur `appliesTo` properti untuk tindakan dalam skema Anda. Ini menentukan tindakan yang valid untuk sumber daya ketika

Anda ingin memastikan bahwa, misalnya, pengguna hanya dapat mencoba `ViewPhoto` untuk mengotorisasi sumber daya jenis. `PhotoFlash::Photo`

```
permit(
  principal == PhotoFlash::User::"alice",
  action,
  resource
) when {
    context has readOnly &&
    context.readOnly == true
};
```

Cara yang lebih baik untuk mengatur properti tindakan dalam skema Anda, bagaimanapun, adalah dengan mengaturnya ke dalam kelompok tindakan fungsional. Misalnya, Anda dapat membuat tindakan bernama `ReadOnlyPhotoAccess` dan disetel `PhotoFlash::Action::"ViewPhoto"` menjadi anggota `ReadOnlyPhotoAccess` sebagai grup tindakan. Contoh ini menunjukkan cara membuat kebijakan yang memberi Alice akses ke tindakan hanya-baca di grup tersebut.

```
permit(
  principal == PhotoFlash::User::"alice",
  action,
  resource
) when {
    action in PhotoFlash::Action::"ReadOnlyPhotoAccess"
};
```

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan semua kepala sekolah melakukan tindakan apa pun pada sumber daya yang memiliki atributnya. `owner`

```
permit(
  principal,
  action,
  resource
)
when {
  principal == resource.owner
};
```

Contoh ini menunjukkan bagaimana Anda dapat membuat kebijakan yang memungkinkan prinsipal mana pun untuk melihat sumber daya apa pun jika `department` atribut untuk prinsipal cocok dengan `department` atribut sumber daya.

> **Note**
>
> Jika entitas tidak memiliki atribut yang disebutkan dalam kondisi kebijakan, maka kebijakan tersebut akan diabaikan saat membuat keputusan otorisasi dan evaluasi kebijakan tersebut gagal untuk entitas tersebut. Misalnya, prinsipal apa pun yang tidak memiliki `department` atribut tidak dapat diberikan akses ke sumber daya apa pun oleh kebijakan ini.

```
permit(
  principal,
  action == Action::"view",
  resource
)
when {
  principal.department == resource.owner.department
};
```

Contoh ini menunjukkan bagaimana Anda dapat membuat kebijakan yang memungkinkan prinsipal untuk melakukan tindakan apa pun pada sumber daya jika prinsipal adalah sumber daya ATAU jika prinsipal adalah bagian dari `admins` grup untuk sumber daya tersebut. `owner`

```
permit(
  principal,
  action,
  resource,
)
when {
  principal == resource.owner |
  resource.admins.contains(principal)
};
```

Menolak akses

Jika kebijakan berisi `forbid` efek kebijakan, kebijakan akan membatasi izin, bukan memberikan izin.

> **Important**
>
> Selama otorisasi, jika baik a `permit` dan `forbid` kebijakan ditegakkan, diutamakan`forbid`.

Contoh berikut menggunakan atribut yang didefinisikan dalam aplikasi hipotetis yang disebut PhotoFlash dijelaskan di bagian skema Contoh dari Panduan Referensi bahasa kebijakan Cedar.

Contoh ini menunjukkan cara Anda membuat kebijakan yang menolak pengguna `alice` melakukan semua tindakan kecuali `readOnly` pada sumber daya apa pun.

```
forbid (
  principal == User::"alice",
  action,
  resource
)
unless {
  action.readOnly
};
```

Contoh ini menunjukkan bagaimana Anda dapat membuat kebijakan yang menolak akses ke semua sumber daya yang memiliki `private` atribut kecuali prinsipal memiliki `owner` atribut untuk sumber daya.

```
forbid (
  principal,
  action,
  resource
)
when {
  resource.private
}
unless {
  principal == resource.owner
};
```

Templat kebijakan Izin Terverifikasi Amazon

Anda dapat membuat templat kebijakan Cedar di Izin Terverifikasi untuk menentukan aturan kontrol akses untuk sistem Anda. Templat kebijakan adalah kebijakan Cedar dengan placeholder untuk`principal`,`resource`, atau keduanya. Templat kebijakan memungkinkan kebijakan didefinisikan sekali dan kemudian dilampirkan ke beberapa prinsip dan sumber daya. Pembaruan pada templat kebijakan tercermin di semua prinsip dan sumber daya yang menggunakan templat. Untuk informasi selengkapnya, lihatTemplat kebijakan cedardalam Panduan Referensi bahasa kebijakan Cedar.

Sebaiknya gunakan templat kebijakan untuk membuat kebijakan yang dapat dibagikan di seluruh aplikasi Anda. Misalnya, Anda dapat membuat templat kebijakan untuk editor yang menyediakan izin baca, edit, dan komentar untuk prinsipal dan sumber daya yang menggunakan templat kebijakan.

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

Ketika prinsipal ditetapkan sebagai editor untuk sumber daya, aplikasi Anda dapat membuat instance kebijakan menggunakan templat untuk memberikan izin bagi prinsipal untuk melakukan tindakan baca, edit, dan komentar pada sumber daya.

Membuat template kebijakan

AWS Management Console

Untuk membuat templat kebijakan

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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 CreatePolicyTemplateoperasi. 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 PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-18T21:17:47.284268+00:00",
    "lastUpdatedDate": "2023-05-18T21:17:47.284268+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

Membuat kebijakan yang ditautkan templat

Anda dapat membuat kebijakan terkait templat untuk ditautkan ke templat kebijakan. 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.

AWS Management Console

Untuk membuat kebijakan yang ditautkan templat dengan membuat instance template kebijakan

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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 == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

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

```
{
    "templateLinked": {
        "policyTemplateId": "pt-4651be67-c128-4d22-8e67-9b068980c631",
        "principal": {
            "entityType": "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 PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-22T18:57:53.298278+00:00",
    "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",
    "policyId": "TPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "TEMPLATELINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "Photo"
    }
}
```

Mengedit templat kebijakan

AWS Management Console

Untuk mengedit templat kebijakan Anda

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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 memperbarui templat kebijakan

Anda dapat membuat kebijakan statis dengan menggunakan UpdatePolicyoperasi. 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 PTEXAMPLEabcdefg111111 \
    --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": "PTEXAMPLEabcdefg111111"
}
```

Contoh kebijakan terkait templat untuk penyimpanan kebijakan contoh Izin Terverifikasi

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.

PhotoFlashcontoh kebijakan terkait templat

Contoh ini 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 ini 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 ini 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 ini 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 ini 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 ini 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

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 DigitalPetStorecontoh.

TinyToDo contoh kebijakan terkait templat

Contoh ini 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 ini menunjukkan cara membuat kebijakan terkait 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"
);
```

Menggunakan Izin Terverifikasi Amazon dengan penyedia identitas

Sumber identitas adalah representasi dari 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. Sumber identitas Izin Terverifikasi meningkatkan otorisasi dengan koneksi langsung ke toko identitas pusat dan layanan otentikasi Anda.

Anda dapat menggunakan penyedia identitas OpenID Connect (OIDC) () dengan Izin TerverifikasiIdPs. Aplikasi Anda dapat menghasilkan permintaan otorisasi dengan identitas (ID) OIDC atau mengakses token web JSON (JWT). Dengan token ID, Izin Terverifikasi membaca ID pengguna dan klaim atribut sebagai prinsip untuk kontrol akses berbasis atribut (ABAC). Dengan token akses, Izin Terverifikasi membaca ID pengguna sebagai kepala sekolah dan klaim lainnya sebagai konteks. Dengan kedua jenis token, Anda dapat memetakan klaim seperti `groups` ke grup utama, dan membuat kebijakan yang mengevaluasi kontrol akses berbasis peran (RBAC).

Anda dapat menambahkan kumpulan pengguna Amazon Cognito atau iDP OpenID Connect (OIDC) kustom sebagai sumber identitas Anda.

Topik

Bekerja dengan sumber identitas Amazon Cognito

Izin Terverifikasi bekerja sama dengan kumpulan pengguna Amazon Cognito. Amazon Cognito JWT 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:

- The Wilayah AWS.
- ID kolam pengguna.
- 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`.
- (Opsional) ID klien dari kumpulan pengguna 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`

Klaim token kumpulan pengguna dapat berisi atribut, cakupan, grup, ID klien, dan data khusus. Amazon Cognito JWT memiliki kemampuan untuk menyertakan berbagai informasi yang dapat berkontribusi pada keputusan otorisasi di Izin Terverifikasi. Ini termasuk:

1. Nama pengguna dan klaim grup dengan `cognito:` awalan
2. Atribut pengguna khusus dengan a `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, diBekerja dengan sumber identitas dalam skema dan kebijakan.

> **Important**
>
> Meskipun Anda dapat mencabut token Amazon Cognito sebelum kedaluwarsa, JWT 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.

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, tulislah 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 IsAuthorizedWithTokenakan gagal dengan. `ValidationException`

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

```
permit(
     principal == ExampleCo::User::"us-east-1_example|4fe90f4a-ref8d9-4033-
a750-4c8622d62fb6",
     action,
     resource == ExampleCo::Photo::"VacationPhoto94.jpg"
)
when {
     principal["cognito:username"]) == "alice" &&
     principal["custom:department"]) == "Finance"
};
```

Untuk informasi selengkapnya tentang pemetaan klaim, lihatMemetakan token ID ke skema. Untuk informasi selengkapnya tentang otorisasi untuk pengguna Amazon Cognito, lihat Otorisasi dengan Izin Terverifikasi Amazon di Panduan Pengembang Amazon Cognito.

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 menghasilkan JWT 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.
- 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 ID entitas 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 sumber identitas atau Penyiapan terpandu, Izin Terverifikasi menetapkan awalan nama penerbit dikurangi`https://`, misalnya. `MyCorp::User::"auth.example.com|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"`

Otorisasi dengan sumber identitas OIDC menggunakan operasi API yang sama dengan sumber identitas kumpulan pengguna: IsAuthorizedWithTokendan Token. BatchIs AuthorizedWith

Contoh 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.

```
permit(
```

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

Validasi klien dan audiens

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 aplikasi OAuth atau OIDC yang dikonfigurasi dengan penyedia, misalnya. `1example23456789` Audiens adalah jalur URL yang terkait dengan pihak yang mengandalkan, atau tujuan, dari aplikasi target, misalnya`https://myapplication.example.com`. `aud`Klaim tidak selalu dikaitkan dengan audiens.

Izin Terverifikasi melakukan pemirsa sumber identitas dan validasi klien sebagai berikut:

Amazon Cognito

Token ID Amazon Cognito 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 ID klien aplikasi ini dengan `aud` klaim token ID atau klaim token akses. `client_id` Izin Terverifikasi tidak memvalidasi URL audiens pihak terkait untuk sumber identitas Amazon Cognito.

OIDC

Token ID OIDC memiliki `aud` klaim yang berisi daftar ID klien. Token akses memiliki `aud` klaim yang berisi URL audiens untuk token tersebut. Token akses juga memiliki `client_id` klaim yang berisi ID klien yang dimaksud.

Anda dapat memasukkan satu atau beberapa nilai untuk validasi Audiens dengan penyedia OIDC. Saat Anda memilih token ID tipe Token, Izin Terverifikasi memvalidasi ID klien, memeriksa bahwa setidaknya satu anggota ID klien dalam `aud` klaim cocok dengan nilai validasi audiens.

Izin Terverifikasi memvalidasi audiens untuk token akses, memeriksa apakah `aud` klaim tersebut cocok dengan nilai validasi audiens. Nilai token akses ini terutama berasal dari `aud` klaim tetapi dapat berasal dari `cid` atau `client_id` klaim jika tidak ada `aud` klaim. Periksa dengan IDP Anda untuk klaim dan format audiens yang benar.

Contoh nilai validasi pemirsa token ID adalah`1example23456789`.

Contoh nilai validasi pemirsa token akses adalah`https://myapplication.example.com`.

Otorisasi sisi klien untuk JWT

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 cara Anda memanggil Izin Terverifikasi dari ID OIDC.[1]

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

    var 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
    };
    var 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
    };
    var 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
```

```
    };
    var entities = {
        entityList: [],
    };
    entities.entityList.push(...getUserEntitiesFromToken(payload));
    var 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 attibuteItem = [];
      value.forEach((item) => {
        attibuteItem.push({
          string: item,
        });
      });
      attributes[key] = {
        set: attibuteItem,
      };
    } 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 === '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 need to use the claim that
 represents the user-id
    }
  };
  return [entityItem];
}
```

[1] Contoh kode ini menggunakan pustaka aws-jwt-verify untuk memverifikasi JWT yang ditandatangani oleh OIDC-kompatibel. IdPs

Membuat sumber identitas Izin Terverifikasi Amazon

Prosedur berikut menambahkan sumber identitas ke toko kebijakan yang ada. Setelah Anda menambahkan sumber identitas Anda, Anda harus menambahkan atribut ke skema Anda.

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 Penyiapan 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 CreateIdentity Source di API 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 identitas Amazon Cognito dengan skema penyimpanan kebijakan, Anda harus menyetel jenis entitas utama ke entitas penyimpanan kebijakan yang didukung.

Topik

- Sumber identitas Amazon Cognito
- Sumber identitas OIDC

Sumber identitas Amazon Cognito

AWS Management Console

Untuk membuat sumber identitas kumpulan pengguna Amazon Cognito

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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 Wilayah AWS dan masukkan ID kumpulan Pengguna untuk sumber identitas Anda.
5. Dalam konfigurasi Principal, pilih tipe Principal untuk sumber identitas. 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 ID aplikasi klien.

- Untuk memvalidasi ID aplikasi klien, pilih Hanya terima token dengan ID aplikasi 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 ID aplikasi klien jika Anda tidak ingin memvalidasi ID aplikasi klien.

8. Pilih Buat sumber identitas.
9. 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 Amazon Cognito ke atribut utama Cedar, lihat. Bekerja dengan sumber identitas dalam skema dan kebijakan

   Saat Anda membuat penyimpanan kebijakan terkait API, Izin Terverifikasi akan menanyakan kumpulan pengguna Anda untuk atribut pengguna dan membuat skema tempat tipe utama Anda diisi dengan atribut kumpulan pengguna.

AWS CLI

Untuk membuat sumber identitas kumpulan pengguna Amazon Cognito

Anda dapat membuat sumber identitas dengan menggunakan operasi CreateIdentitySumber. Contoh berikut membuat sumber identitas yang dapat mengakses identitas yang diautentikasi dari kumpulan pengguna Amazon Cognito.

`config.txt`File berikut berisi rincian kumpulan pengguna Amazon Cognito untuk digunakan oleh parameter --configuration dalam perintah. `create-identity-source`

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

```
}
```

Perintah:

```
$ 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"
}
```

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 Amazon Cognito ke atribut utama Cedar, lihat. Bekerja dengan sumber identitas dalam skema dan kebijakan

Saat Anda membuat penyimpanan kebijakan terkait API, Izin Terverifikasi akan menanyakan kumpulan pengguna Anda untuk atribut pengguna dan membuat skema tempat tipe utama Anda diisi dengan atribut kumpulan pengguna.

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.

Sumber identitas OIDC

AWS Management Console

Untuk membuat sumber identitas OpenID Connect (OIDC)

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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 Bekerja dengan sumber identitas dalam skema dan kebijakan.
7. Dalam klaim Pengguna dan grup, 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. Dalam klaim Pengguna dan grup, 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 Audiens, masukkan ID klien atau URL audiens yang ingin diterima oleh toko kebijakan Anda dalam permintaan otorisasi, jika ada.
10. Pilih Buat sumber identitas.
11. Perbarui skema Anda untuk membuat Cedar sadar akan 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 Amazon Cognito ke atribut utama Cedar, lihat. Bekerja dengan sumber identitas dalam skema dan kebijakan

    Saat Anda membuat penyimpanan kebijakan terkait API, Izin Terverifikasi akan menanyakan kumpulan pengguna Anda untuk atribut pengguna dan membuat skema tempat tipe utama Anda diisi dengan atribut kumpulan pengguna.

AWS CLI

Untuk membuat sumber identitas OIDC

Anda dapat membuat sumber identitas dengan menggunakan operasi CreateIdentitySumber. Contoh berikut membuat sumber identitas yang dapat mengakses identitas yang diautentikasi dari kumpulan pengguna Amazon Cognito.

`config.txt`File berikut berisi rincian IdP OIDC untuk digunakan oleh `--configuration` parameter perintah. `create-identity-source` Contoh ini membuat sumber identitas OIDC untuk token ID.

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

`config.txt`File berikut berisi rincian IdP OIDC untuk digunakan oleh `--configuration` parameter perintah. `create-identity-source` Contoh ini menciptakan sumber identitas OIDC untuk token akses.

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

```
        }
    }
}
```

Perintah:

```
$ 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"
}
```

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 Amazon Cognito ke atribut utama Cedar, lihat. Bekerja dengan sumber identitas dalam skema dan kebijakan

Saat Anda membuat penyimpanan kebijakan terkait API, Izin Terverifikasi akan menanyakan kumpulan pengguna Anda untuk atribut pengguna dan membuat skema tempat tipe utama Anda diisi dengan atribut kumpulan pengguna.

Mengedit sumber identitas Izin Terverifikasi Amazon

Anda dapat mengedit beberapa parameter sumber identitas Anda setelah Anda membuatnya. 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.

Topik

- Sumber identitas kumpulan pengguna Amazon Cognito
- Sumber identitas OpenID Connect (OIDC)

Sumber identitas kumpulan pengguna Amazon Cognito

AWS Management Console

Untuk memperbarui sumber identitas kumpulan pengguna Amazon Cognito

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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 Wilayah AWS 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 ID aplikasi klien.
   - Untuk memvalidasi ID aplikasi klien, pilih Hanya terima token dengan ID aplikasi 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 ID aplikasi klien jika Anda tidak ingin memvalidasi ID aplikasi klien.
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 identitas kumpulan pengguna Amazon Cognito

Anda dapat memperbarui sumber identitas dengan menggunakan operasi UpdateIdentitySumber. Contoh berikut memperbarui sumber identitas yang ditentukan untuk menggunakan kumpulan pengguna Amazon Cognito yang berbeda.

`config.txt`File berikut berisi rincian kumpulan pengguna Amazon Cognito untuk digunakan oleh parameter --configuration dalam perintah. `create-identity-source`

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

Perintah:

```
$ 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"
}
```

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

Sumber identitas OpenID Connect (OIDC)

AWS Management Console

Untuk memperbarui sumber identitas OIDC

1. Buka konsol Izin Terverifikasi di https://console.aws.amazon.com/verifiedpermissions/. 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 operasi UpdateIdentitySumber. Contoh berikut memperbarui sumber identitas yang ditentukan untuk menggunakan penyedia OIDC yang berbeda.

`config.txt`File berikut berisi rincian kumpulan pengguna Amazon Cognito untuk digunakan oleh parameter --configuration dalam perintah. `create-identity-source`

```
{
    "openIdConnectConfiguration": {
        "issuer": "https://auth2.example.com",
        "tokenSelection": {
                "identityTokenOnly": {
```

```
                        "clientIds":["2example10111213"],
                        "principalIdClaim": "sub"
                },
        },
        "entityIdPrefix": "MyOIDCProvider",
        "groupConfiguration": {
              "groupClaim": "groups",
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

Perintah:

```
$ 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"
}
```

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

Bekerja dengan sumber identitas dalam skema dan kebijakan

Anda mungkin menemukan bahwa Anda ingin menambahkan sumber identitas ke toko kebijakan dan klaim penyedia peta ke skema toko kebijakan Anda. Anda dapat mengotomatiskan proses ini atau memperbarui skema Anda secara manual. Bagian panduan pengguna ini memiliki informasi berikut:

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

Penyimpanan kebijakan terkait API dan penyimpanan kebijakan dengan sumber identitas melalui Penyiapan terpandu tidak memerlukan pemetaan manual atribut token identitas (ID) ke skema. Anda

dapat memberikan Izin Terverifikasi dengan atribut di kumpulan pengguna atau token OIDC dan membuat skema yang diisi dengan atribut pengguna. Dalam otorisasi token ID, Izin Terverifikasi memetakan klaim ke atribut entitas utama. Anda mungkin perlu memetakan token Amazon Cognito 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 menggunakan Amazon Cognito atau penyedia identitas OIDC (iDP) sebagai sumber identitas di toko kebijakan Izin Terverifikasi, Anda harus memiliki atribut penyedia dalam skema Anda. 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, Anda harus menambahkan atribut penyedia ke skema. Skema Anda harus sesuai dengan entitas yang dibuat oleh token penyedia IsAuthorizedWithTokenatau permintaan BatchIsAuthorizedWithToken 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

- Hal-hal yang perlu diketahui tentang pemetaan skema
- Memetakan token ID ke skema
- Memetakan token akses
- Notasi alternatif untuk klaim Amazon Cognito colon-delimited

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 sumber 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 identitas Amazon Cognito, 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 ke penyimpanan kebijakan Anda, dan sintaks otorisasi Lambda untuk API Gateway API Anda. 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:groups` `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 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"
      ]
   }
}
```

Contoh kebijakan untuk atribut yang akan memvalidasi skema ini terlihat seperti berikut:

```
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.*"
};
```

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.

Token ID Amazon Cognito

Token ID Amazon Cognito berfungsi 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 Pengembang Amazon Cognito.

Klaim yang berguna dalam token ID Amazon Cognito

`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 atribut Amazon Cognito 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 Pengembang Amazon Cognito.

Contoh ID token berikut memiliki masing-masing dari empat jenis atribut. Ini termasuk klaim khusus Amazon Cognito`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-TOKEN1111111",
    "email": "alice@example.com"
}
```

Saat membuat sumber identitas dengan kumpulan pengguna Amazon Cognito, Anda menentukan jenis entitas utama yang dihasilkan oleh 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, lihatMengedit skema dalam mode JSON. 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
         }
      }
   }
}
```

Setelah memperbarui skema untuk mencerminkan atribut Amazon Cognito, Anda dapat membuat kebijakan yang mereferensikan atribut.

```
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"
};
```

Token ID OIDC

Bekerja dengan token ID dari penyedia OIDC hampir sama dengan bekerja dengan token ID Amazon Cognito. 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"
         }
      }
   }
}
```

Kebijakan berikut berlaku untuk anggota grup di penyedia OIDC Anda.

```
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*"
};
```

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.

Memetakan token akses Amazon Cognito

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

Klaim yang berguna dalam token akses Amazon Cognito

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 Pengembang Amazon Cognito.

Token akses Amazon Cognito 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-TOKEN1111111",
    "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-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, lihatMengedit skema dalam mode JSON.

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

Setelah memperbarui skema untuk mencerminkan atribut Amazon Cognito, Anda dapat membuat kebijakan yang mereferensikan atribut.

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

Memetakan token akses OIDC

Sebagian besar token akses dari penyedia OIDC eksternal sejajar erat dengan token akses Amazon Cognito. 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, lihatMengedit skema dalam mode JSON.

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

Setelah memperbarui skema untuk mencerminkan atribut IDP, Anda dapat membuat kebijakan yang mereferensikan atribut.

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

Notasi alternatif untuk klaim Amazon Cognito colon-delimited

Pada saat Izin Terverifikasi diluncurkan, skema yang direkomendasikan untuk token Amazon Cognito mengklaim `cognito:groups` menyukai `custom:store` dan mengonversi string yang dibatasi titik dua ini untuk menggunakan karakter sebagai pembatas hierarki. `.` Format ini disebut notasi titik. Misalnya, 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
         }
      }
   }
}
```

Dengan skema dalam format ini, Anda dapat membuat kebijakan dengan notasi titik seperti pada contoh berikut:

```
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"
};
```

Merancang model otorisasi untuk aplikasi Anda

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, mengumpulkan pemahaman yang jelas tentang apa yang harus dilihat pengguna akhir saat mengelola izin di UI aplikasi. 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? Apakah mereka memiliki hubungan satu sama lain? Misalnya, apakah file berada di dalam folder?
- 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?
- Haruskah izin mewarisi seluruh sumber daya, seperti file yang mewarisi izin dari folder induk?
- 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 dilihat 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.

Bagian ini memberikan panduan umum tentang cara mendekati latihan desain, hal-hal yang harus diperhatikan, dan kumpulan praktik terbaik untuk menggunakan Izin Terverifikasi dengan sukses.

Selain pedoman yang disajikan di sini, ingatlah untuk mempertimbangkan praktik terbaik dalam panduan referensi bahasa kebijakan Cedar.

Topik

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.

Fokus pada sumber daya Anda di luar operasi API

Di sebagian besar aplikasi yang dihadapi konsumen, izin dimodelkan di sekitar sumber daya yang didukung oleh aplikasi. 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, pendekatan yang berfokus pada API ini bisa kurang optimal, karena API hanyalah proxy untuk apa yang benar-benar ingin dilindungi oleh pelanggan Anda: data dan sumber daya yang mendasarinya. Jika beberapa API mengontrol akses ke sumber daya yang sama, mungkin sulit bagi administrator untuk mempertimbangkan 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 mengabaikan semantik API dan sebagai gantinya fokus pada data dan sumber daya yang mendasarinya serta operasi asosiasi mereka. Menerapkan pendekatan ini ke contoh keanggotaan grup akan menghasilkan izin abstrak, sepertiviewGroupMembership, yang masing-masing dari tiga operasi API harus berkonsultasi.

Nama API	Izin
`listMembersOfGroup`	Memerlukan `viewGroupMembership` izin pada grup
`listGroupMembershipsForUser`	memerlukan `viewGroupMembership` izin pada pengguna
`isUserInGroups`	memerlukan `viewGroupMembership` 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.

Otorisasi majemuk adalah normal

Otorisasi gabungan terjadi ketika satu aktivitas pengguna, seperti mengklik tombol di antarmuka aplikasi Anda, memerlukan beberapa kueri otorisasi individual untuk menentukan apakah aktivitas tersebut diizinkan. Misalnya, memindahkan file ke direktori baru dalam sistem file mungkin memerlukan tiga izin yang berbeda: kemampuan untuk menghapus file dari direktori sumber, kemampuan untuk menambahkan file ke direktori tujuan, dan mungkin kemampuan untuk menyentuh file itu sendiri (tergantung pada aplikasi).

Jika Anda baru merancang model otorisasi, Anda mungkin berpikir bahwa setiap keputusan otorisasi harus dapat diselesaikan dalam satu kueri otorisasi. Tetapi ini dapat menyebabkan model yang

terlalu kompleks dan pernyataan kebijakan yang berbelit-belit. Dalam praktiknya, menggunakan otorisasi majemuk dapat berguna dalam membantu Anda menghasilkan model otorisasi yang lebih sederhana. Salah satu ukuran dari model otorisasi yang dirancang dengan baik adalah bahwa ketika Anda memiliki tindakan individual yang cukup terurai, operasi gabungan Anda, seperti memindahkan file, dapat diwakili oleh agregasi intuitif primitif.

Situasi lain di mana otorisasi majemuk terjadi adalah ketika banyak pihak terlibat dalam proses pemberian izin. Pertimbangkan direktori organisasi tempat pengguna dapat menjadi anggota grup. Pendekatan sederhana adalah memberi izin kepada pemilik grup untuk menambahkan siapa pun. Namun, bagaimana jika Anda ingin pengguna Anda terlebih dahulu menyetujui untuk ditambahkan? Ini memperkenalkan perjanjian jabat tangan di mana pengguna dan grup harus menyetujui keanggotaan. Untuk mencapai hal ini, Anda dapat memperkenalkan izin lain yang terikat pada pengguna dan menentukan apakah pengguna dapat ditambahkan ke grup apa pun, atau ke grup tertentu. Ketika penelepon kemudian mencoba menambahkan anggota ke grup, aplikasi harus memberlakukan kedua sisi izin: bahwa pemanggil memiliki izin untuk menambahkan anggota ke grup yang ditentukan, dan bahwa pengguna individu yang ditambahkan memiliki izin untuk ditambahkan. KapanN-cara jabat tangan ada, itu umum untuk diamatiNkueri otorisasi gabungan untuk menegakkan setiap bagian dari perjanjian.

Jika Anda menemukan diri Anda dengan tantangan desain di mana banyak sumber daya terlibat dan tidak jelas bagaimana memodelkan izin, itu bisa menjadi tanda bahwa Anda memiliki skenario otorisasi gabungan. Dalam hal ini, solusi dapat ditemukan dengan menguraikan operasi menjadi beberapa pemeriksaan otorisasi individu.

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.

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 menciptakan volume permintaan otorisasi yang relatif lebih tinggi yang mungkin berdampak pada tagihan Anda. AWS 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

Templat kebijakan Anda 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, lihatKuota 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. Must include tenant identifiers in policies and authorization requests.	Rendah. Isolation is default behavior. Tenant-specific policies are inaccessible to other tenants.
Aliran otorisasi	Rendah. All queries target one policy store.	Sedang. Must maintain mappings between each tenant and their policy store ID.
Template dan manajemen skema	Rendah. Must make one schema work for all tenants.	Tinggi. Schemas and templates might be less complex individually, but changes require more coordination and complexity.
Manajemen kebijakan global	Rendah. All policies are global and can be centrally updated.	Tinggi. You must add global policies to each policy store in onboarding. Replicate global policy updates between many policy stores.
Penyewa off-boarding	Sedang. Must identify and delete only tenant-specific policies.	Rendah. Delete the policy store.
Kuota sumber daya layanan	Tinggi. Tenants share resource quotas that affect policy stores like schema size, policy size per resource, and	Rendah. Each tenant has dedicated resource quotas.

identity sources per policy store.

Bagaimana memilih

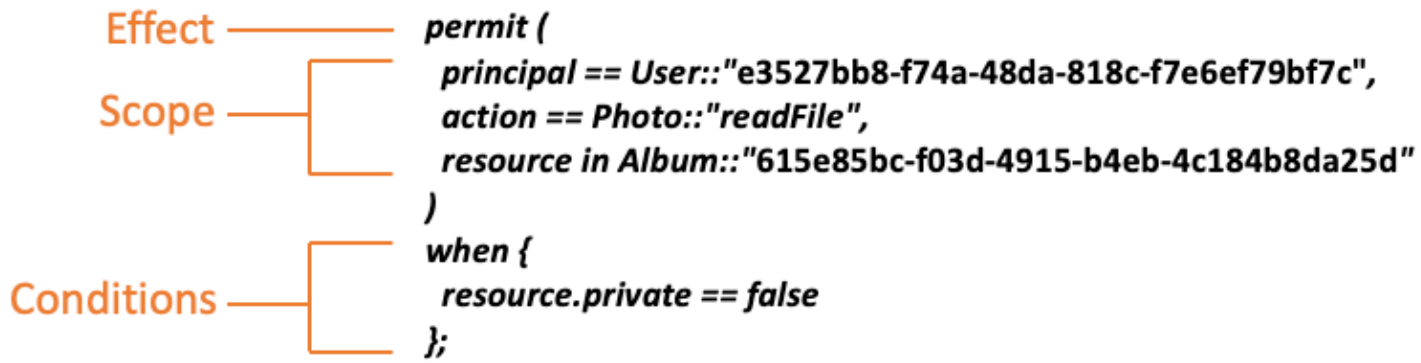
Setiap aplikasi multi-tenant berbeda. Hati-hati membandingkan dua pendekatan dan pertimbangan mereka sebelum membuat keputusan arsitektur.

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.

Jika memungkinkan, isi cakupan kebijakan

Ruang lingkup kebijakan adalah bagian dari pernyataan kebijakan Cedar setelah`permit`atau`forbid`kata kunci dan antara tanda kurung pembuka.



Kami menyarankan Anda mengisi nilai untuk`principal`dan`resource`bila memungkinkan. Hal ini memungkinkan Izin Terverifikasi mengindeks kebijakan untuk pengambilan yang lebih efisien

dan karenanya meningkatkan kinerja. Jika Anda perlu memberikan izin yang sama ke banyak prinsipal atau sumber daya yang berbeda, kami sarankan Anda menggunakan templat kebijakan dan melampirkannya ke setiap pasangan utama dan sumber daya.

Hindari membuat satu kebijakan besar yang berisi daftar prinsipal dan sumber daya dalam`when`klausa. Melakukannya kemungkinan akan menyebabkan Anda mengalami batas skalabilitas atau tantangan operasional. Misalnya, untuk menambah atau menghapus satu pengguna dari daftar besar dalam kebijakan, Anda perlu membaca seluruh kebijakan, mengedit daftar, menulis kebijakan baru secara lengkap, dan menangani kesalahan konkurensi jika satu administrator menimpa perubahan orang lain. Sebaliknya, dengan menggunakan banyak izin berbutir halus, menambahkan atau menghapus pengguna semudah menambahkan atau menghapus kebijakan tunggal yang berlaku untuk mereka.

Setiap sumber daya hidup dalam wadah

Saat Anda mendesain model otorisasi, setiap tindakan harus dikaitkan dengan sumber daya tertentu. Dengan tindakan seperti`viewFile`, sumber daya yang dapat Anda terapkan adalah intuitif: file individual, atau mungkin kumpulan file dalam folder. Namun, operasi seperti`createFile`kurang intuitif. Saat memodelkan kemampuan untuk membuat file, sumber daya apa yang diterapkan? Itu tidak bisa menjadi file itu sendiri, karena file tersebut belum ada.

Ini adalah contoh masalah umum penciptaan sumber daya. Pembuatan sumber daya adalah masalah bootstrap. Harus ada cara agar sesuatu memiliki izin untuk membuat sumber daya bahkan ketika belum ada sumber daya. Solusinya adalah mengenali bahwa setiap sumber daya harus ada dalam beberapa wadah, dan wadah itu sendiri yang bertindak sebagai titik jangkar untuk izin. Misalnya, jika folder sudah ada di sistem, kemampuan untuk membuat file dapat dimodelkan sebagai izin pada folder itu, karena itu adalah lokasi di mana izin diperlukan untuk membuat instance sumber daya baru.

```
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    action == Action::"createFile",
    resource == Folder::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

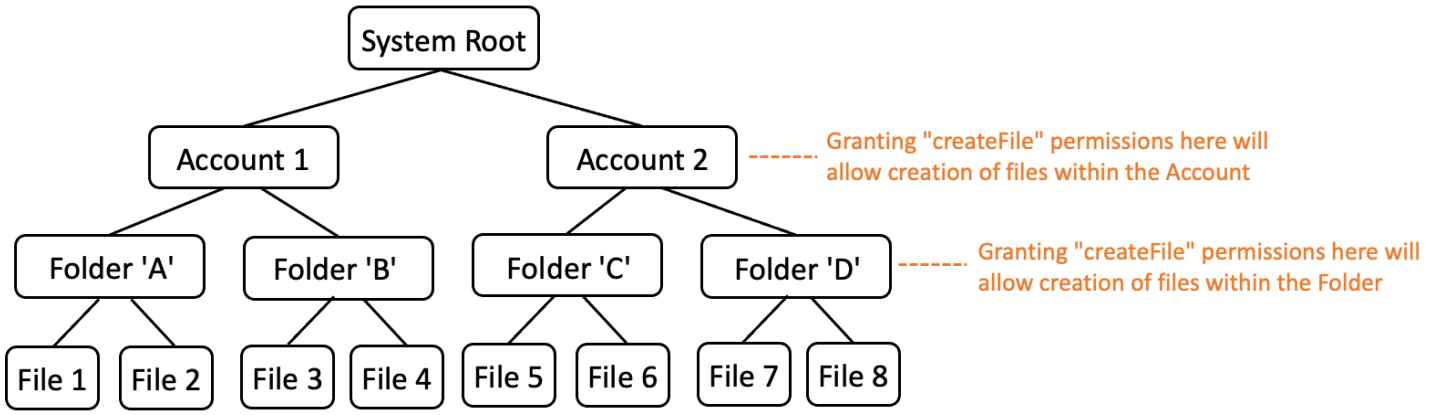
Tetapi bagaimana jika tidak ada folder? Mungkin ini adalah akun pelanggan baru dalam aplikasi di mana belum ada sumber daya. Dalam situasi ini, masih ada konteks yang dapat dipahami secara intuitif dengan bertanya: di mana pelanggan dapat membuat file baru? Anda tidak ingin mereka dapat

membuat file di dalam akun pelanggan acak apa pun. Sebaliknya, ada konteks tersirat: batas akun pelanggan sendiri. Oleh karena itu, akun itu sendiri mewakili wadah untuk pembuatan sumber daya, dan ini dapat secara eksplisit dimodelkan dalam kebijakan yang mirip dengan contoh berikut.

```
// Grants permission to create files within an account,
// or within any sub-folder inside the account.
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    action == Action::"createFile",
    resource in Account::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

Namun, bagaimana jika tidak ada akun juga? Anda dapat memilih untuk merancang alur kerja pendaftaran pelanggan sehingga membuat akun baru dalam sistem. Jika demikian, Anda memerlukan wadah untuk menahan batas terluar di mana proses dapat membuat akun. Wadah tingkat root ini mewakili sistem secara keseluruhan dan mungkin diberi nama sesuatu seperti “root sistem”. Namun, keputusan apakah ini diperlukan, dan apa nama itu terserah Anda, pemilik aplikasi.

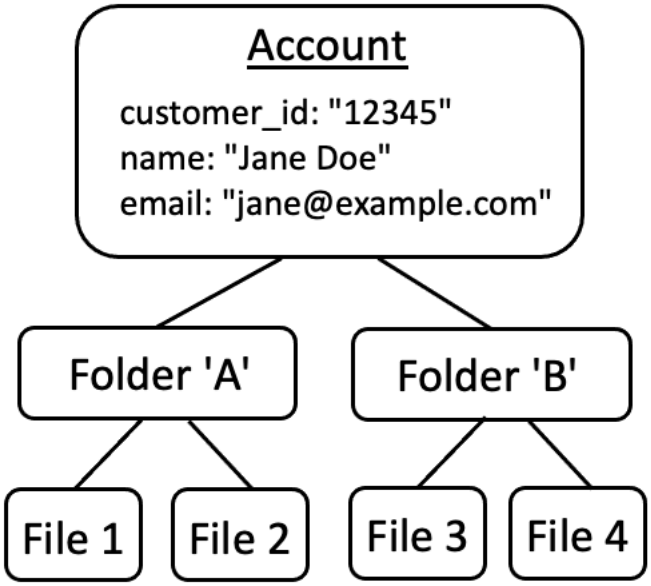
Untuk aplikasi contoh ini, hierarki kontainer yang dihasilkan akan muncul sebagai berikut:



Ini adalah salah satu contoh hierarki. Yang lain juga valid. Hal yang perlu diingat adalah bahwa pembuatan sumber daya selalu terjadi dalam konteks wadah sumber daya. Wadah ini bisa implisit, seperti batas akun, dan mudah untuk mengabaikannya. Saat merancang model otorisasi Anda, pastikan untuk mencatat asumsi implisit ini sehingga dapat didokumentasikan secara formal dan direpresentasikan dalam model otorisasi.

Pisahkan prinsipal dari wadah sumber daya

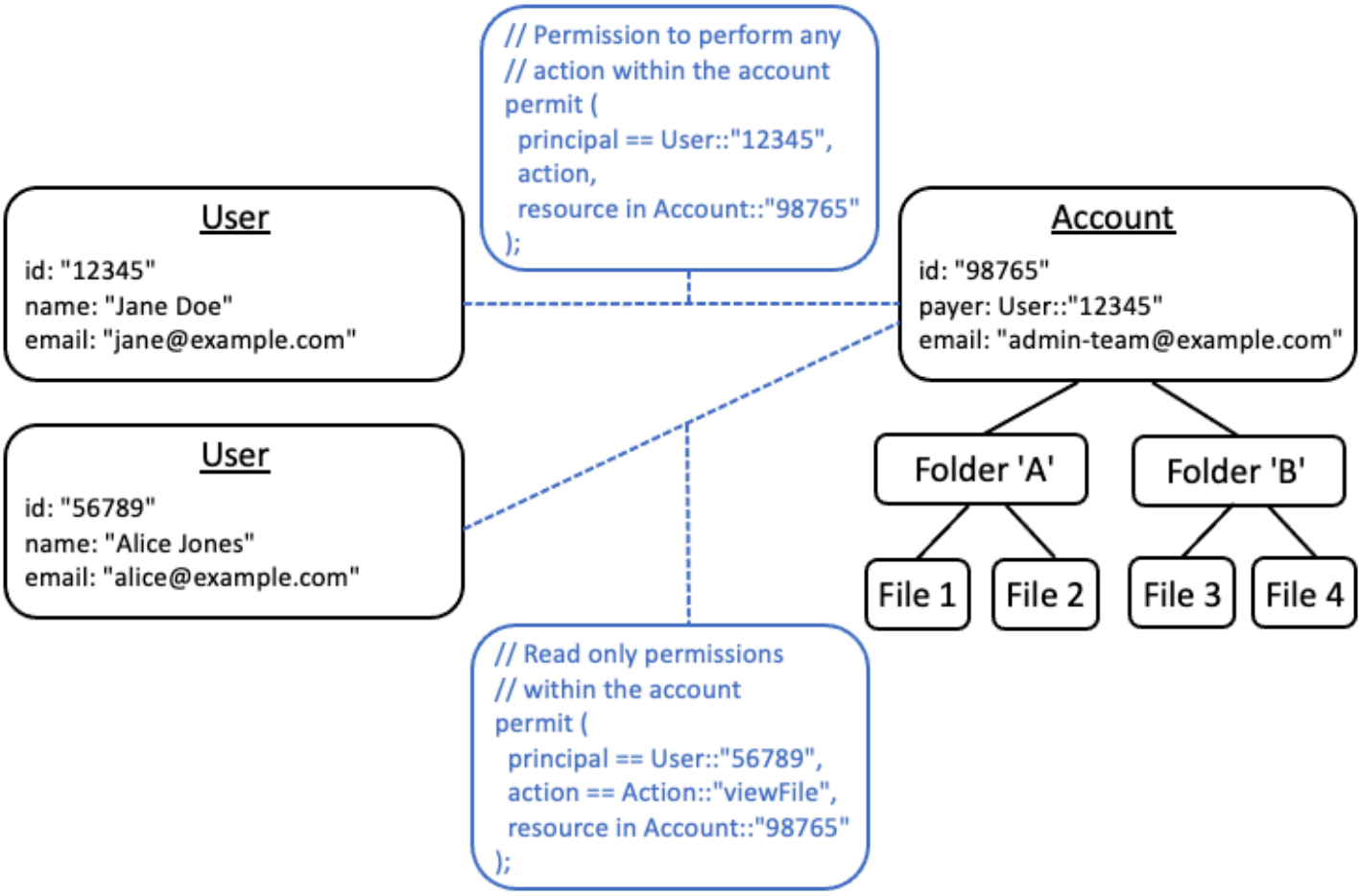
Ketika Anda merancang hierarki sumber daya, salah satu kecenderungan umum, terutama untuk aplikasi yang dihadapi konsumen, adalah menggunakan identitas pengguna pelanggan sebagai wadah untuk sumber daya dalam akun pelanggan.



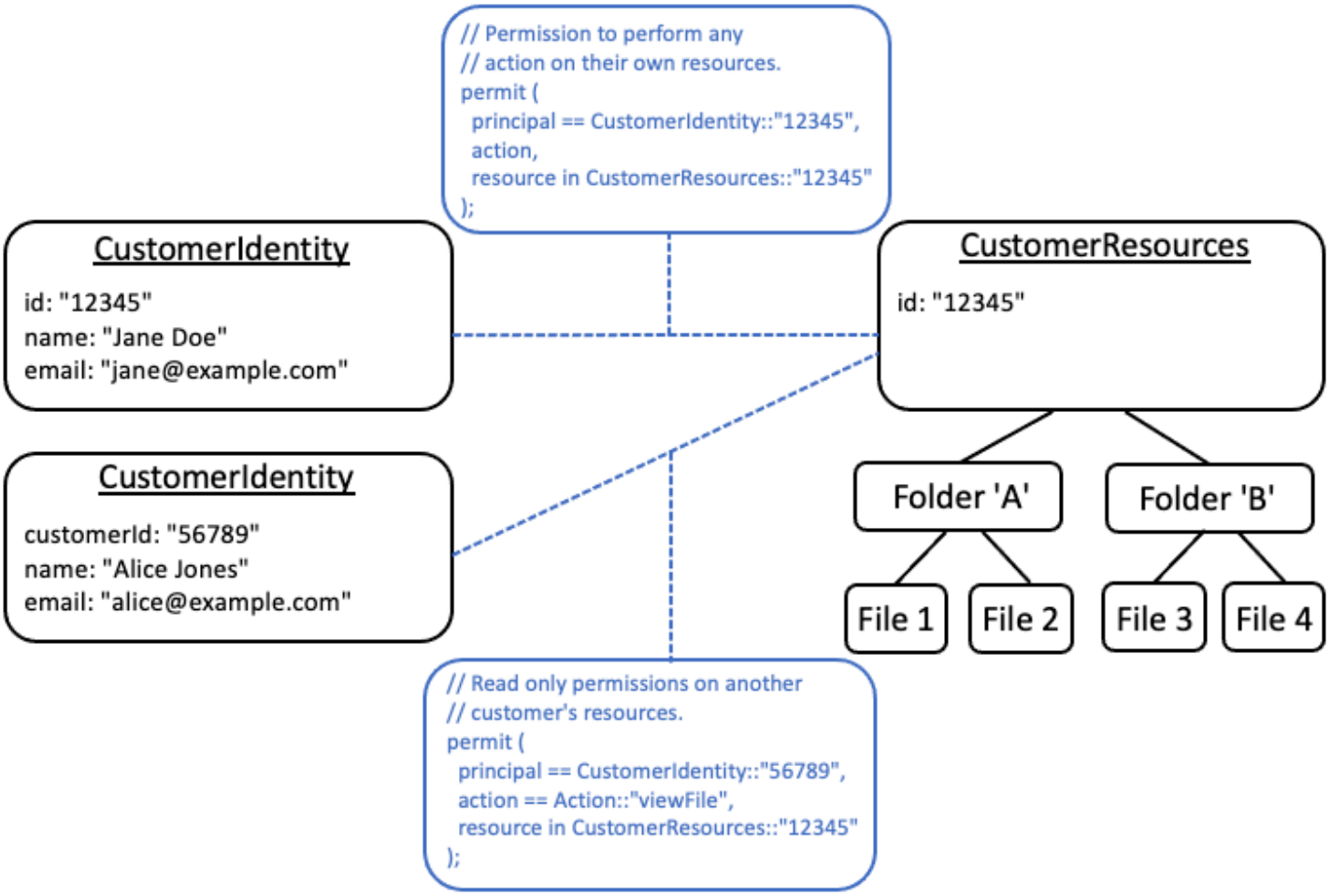
Kami merekomendasikan Anda memperlakukan strategi ini sebagai anti-pola. Ini karena ada kecenderungan alami dalam aplikasi yang lebih kaya untuk mendelegasikan akses ke pengguna tambahan. Misalnya, Anda dapat memilih untuk memperkenalkan akun “keluarga”, tempat pengguna lain dapat berbagi sumber daya akun. Demikian pula, pelanggan perusahaan terkadang ingin menunjuk beberapa anggota tenaga kerja sebagai operator untuk bagian akun. Anda mungkin juga perlu mentransfer kepemilikan akun ke pengguna lain, atau menggabungkan sumber daya dari beberapa akun secara bersamaan.

Ketika identitas pengguna digunakan sebagai wadah sumber daya untuk akun, skenario sebelumnya menjadi lebih sulit dicapai. Lebih mengkhawatirkan, jika orang lain diberikan akses ke wadah akun dalam pendekatan ini, mereka mungkin secara tidak sengaja diberikan akses untuk memodifikasi identitas pengguna itu sendiri, seperti mengubah email Jane atau kredensi login.

Oleh karena itu, bila memungkinkan untuk melakukannya, pendekatan yang lebih tangguh adalah memisahkan prinsipal dari wadah sumber daya, dan memodelkan hubungan di antara mereka dengan menggunakan konsep seperti “izin admin” atau “kepemilikan”.



Jika Anda memiliki aplikasi yang sudah ada yang tidak dapat mengejar model terpisah ini, kami sarankan Anda mempertimbangkan untuk menirunya sebanyak mungkin saat merancang model otorisasi. Misalnya, aplikasi yang hanya memiliki satu konsep bernama`Customer`yang merangkum identitas pengguna, kredensi login, dan sumber daya yang mereka miliki, dapat memetakan ini ke model otorisasi yang berisi satu entitas logis untuk`Customer Identity`(berisi nama, email, dll) dan entitas logis terpisah untuk`Customer Resources`atau`Customer Account`, bertindak sebagai node induk untuk semua sumber daya yang mereka miliki. Kedua entitas dapat berbagi hal yang sama`Id`, tetapi dengan yang berbeda`Type`.



Jangan menyematkan izin di dalam atribut

Atribut paling baik digunakan sebagaimemasukkanuntuk keputusan otorisasi. Jangan gunakan atribut untuk mewakili izin itu sendiri, seperti dengan mendeklarasikan atribut bernama “PermittedFolders” pada Pengguna:

```
// ANTI-PATTERN: comingling permissions into user attributes
{
    "id": "df82e4ad-949e-44cb-8acf-2d1acda71798",
    "name": "alice",
    "email": "alice@example.com",
    "permittedFolders": [
        "Folder::\"c943927f-d803-4f40-9a53-7740272cb969\"",
        "Folder::\"661817a9-d478-4096-943d-4ef1e082d19a\"",
        "Folder::\"b8ee140c-fa09-46c3-992e-099438930894\""
    ]
```

```
}
```

Dan, selanjutnya menggunakan atribut dalam kebijakan:

```
// ANTI-PATTERN
permit (
    principal,
    action == Action::"readFile",
    resource
)
when {
    resource in principal.permittedFolders
};
```

Pendekatan ini mengubah apa yang seharusnya menjadi model otorisasi sederhana, di mana prinsipal tertentu memiliki akses ke folder tertentu, menjadi model kontrol akses berbasis atribut (ABAC) dengan pengorbanan yang menyertainya. Salah satu tradeoff tersebut adalah menjadi lebih sulit untuk dengan cepat menentukan siapa yang memiliki izin untuk sumber daya. Dalam contoh sebelumnya, untuk menentukan siapa yang memiliki akses ke folder tertentu, perlu untuk mengulangi setiap pengguna untuk memeriksa apakah folder tersebut terdaftar dalam atribut mereka, dan melakukannya dengan kesadaran khusus bahwa ada kebijakan yang memberikan akses ketika mereka melakukannya.

Risiko lain dengan pendekatan ini adalah faktor penskalaan ketika izin dikemas bersama dalam satu`User`mencatat Jika pengguna memiliki akses ke banyak hal, ukuran kumulatif mereka`User`catatan akan tumbuh dan mungkin mendekati batas maksimum dari sistem apa pun yang menyimpan data.

Sebagai gantinya, sebaiknya Anda merepresentasikan skenario ini menggunakan beberapa kebijakan individual, mungkin menggunakan templat kebijakan untuk meminimalkan pengulangan.

```
//BETTER PATTERN
permit (
    principal == User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
    action == Action::"readFile",
    resource in Folder::"c943927f-d803-4f40-9a53-7740272cb969"
);

permit (
    principal == User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
```

```
    action == Action::"readFile",
    resource in Folder::"661817a9-d478-4096-943d-4ef1e082d19a"
);

permit (
    principal == User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
    action == Action::"readFile",
    resource in Folder::"b8ee140c-fa09-46c3-992e-099438930894"
);
```

Izin Terverifikasi dapat secara efisien menangani banyak kebijakan individual dan berbutir halus selama evaluasi otorisasi. Memodelkan hal-hal dengan cara ini lebih mudah dikelola dan diaudit dari waktu ke waktu.

Memilih izin berbutir halus dalam model dan izin agregat di antarmuka pengguna

Salah satu strategi yang sering disesali oleh desainer kemudian adalah merancang model otorisasi dengan tindakan yang sangat luas, sepertiReaddan`Write`, dan kemudian menyadari bahwa tindakan yang lebih halus diperlukan. Kebutuhan akan perincian yang lebih halus dapat didorong oleh umpan balik pelanggan untuk kontrol akses yang lebih terperinci, atau oleh auditor kepatuhan dan keamanan yang mendorong izin hak istimewa paling sedikit.

Jika izin berbutir halus tidak didefinisikan di muka, ini dapat memerlukan konversi yang rumit untuk memodifikasi kode aplikasi dan pernyataan kebijakan ke izin pengguna yang lebih halus. Misalnya, kode aplikasi yang sebelumnya diotorisasi terhadap tindakan berbutir kursus perlu dimodifikasi untuk menggunakan tindakan berbutir halus. Selain itu, kebijakan perlu diperbarui untuk mencerminkan migrasi:

```
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    // action  ==  Action::"read",              -- coarse-grained permission --
 commented out
    action in [                       //        -- finer grained permissions
        Action::"listFolderContents",
        Action::"viewFile"
    ],
    resource in Account::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

Untuk menghindari migrasi yang mahal ini, lebih baik menentukan izin berbutir halus di muka. Namun, ini dapat menghasilkan tradeoff jika pengguna akhir Anda kemudian dipaksa untuk memahami sejumlah besar izin halus, terutama jika sebagian besar pelanggan akan puas dengan kontrol berbutir kursus sepertiReaddanWrite. Untuk mencapai yang terbaik dari kedua dunia, Anda dapat mengelompokkan izin berbutir halus ke dalam koleksi yang telah ditentukan sepertiReaddanWritemenggunakan mekanisme seperti templat kebijakan atau grup tindakan. Dengan menggunakan pendekatan ini, pelanggan hanya melihat izin berbutir kursus. Namun di balik layar, Anda telah membuktikan aplikasi Anda di masa depan dengan memodelkan izin berbutir kursus sebagai kumpulan tindakan berbutir halus. Ketika pelanggan atau auditor memintanya, izin berbutir halus dapat diekspos.

Pertimbangkan alasan lain untuk meminta otorisasi

Kami biasanya mengaitkan pemeriksaan otorisasi dengan permintaan pengguna. Pemeriksaan adalah cara untuk menentukan apakah pengguna memiliki izin untuk melakukan permintaan itu. Namun, Anda juga dapat menggunakan data otorisasi untuk memengaruhi desain antarmuka aplikasi. Misalnya, Anda mungkin ingin menampilkan layar beranda yang hanya menampilkan daftar sumber daya yang dapat diakses pengguna akhir. Saat melihat detail sumber daya, Anda mungkin ingin antarmuka hanya menampilkan operasi yang dapat dilakukan pengguna pada sumber daya tersebut.

Situasi ini dapat memperkenalkan pengorbanan ke dalam model otorisasi. Misalnya, ketergantungan yang besar pada kebijakan kontrol akses berbasis atribusi (ABAC) dapat membuat lebih sulit untuk menjawab pertanyaan “siapa yang memiliki akses ke apa?” Ini karena menjawab pertanyaan itu memerlukan pemeriksaan setiap aturan terhadap setiap prinsip dan sumber daya untuk menentukan apakah ada kecocokan. Akibatnya, produk yang perlu dioptimalkan untuk mencantumkan hanya sumber daya yang dapat diakses oleh pengguna dapat memilih untuk menggunakan model kontrol akses berbasis peran (RBAC). Dengan menggunakan RBAC, akan lebih mudah untuk mengulangi semua kebijakan yang dilampirkan ke pengguna untuk menentukan akses sumber daya.

Bangku tes

Bangku uji Izin Terverifikasi memungkinkan Anda 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 gambaran visual dan demonstrasi bangku tes, lihat video ini.

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 di https://console.aws.amazon.com/verifiedpermissions/. 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 dari 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 untuk 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 di https://console.aws.amazon.com/verifiedpermissions/. 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 untuk 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.

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 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 API eksternal. 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 toko kebijakan sampel DigitalPetStore 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

Operasi API 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 memvalidasi entitas dalam permintaan Anda terhadap skema penyimpanan kebijakan Anda. Izin Terverifikasi kemudian mengevaluasi permintaan Anda terhadap semua kebijakan di toko kebijakan yang diminta yang berlaku untuk entitas dalam permintaan.

IsAuthorizedWithToken

`IsAuthorizedWithToken`Operasi ini menghasilkan permintaan otorisasi dari data pengguna di token web Amazon Cognito JSON (JWT). Izin Terverifikasi berfungsi langsung dengan 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 di kumpulan pengguna Amazon Cognito.

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.

BatchIsDiotorisasi

`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 kumpulan pengguna Amazon Cognito.

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 saat menerapkan aplikasi, Anda dapat mengevaluasi kebijakan saat mengembangkannya dengan Bangku tes dan dengan permintaan HTTPS REST API ke Izin Terverifikasi. Bangku tes adalah alat AWS Management Console 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 BatchIsDiotorisasi 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 https://console.aws.amazon.com/verifiedpermissions/. 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 IsAuthorizedreferensi API Izin Terverifikasi. Rincian berikut mereplikasi kondisi dalam Contoh 4 yang mereferensikan sampel DigitalPetStore.

    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 https://console.aws.amazon.com/verifiedpermissions/. Buat toko kebijakan dari toko kebijakan Sample dengan nama DigitalPetStore.
2. Perhatikan ID toko Kebijakan toko kebijakan baru Anda.
3. Dari IsAuthorizedreferensi API Izin Terverifikasi, salin badan permintaan Contoh 4 yang mereferensikan sampel DigitalPetStore.
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 toko kebijakan DigitalPetStore 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 menjadiBob.
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.

Integrasi dengan aplikasi dan AWS SDK

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 identitas Amazon Cognito yang ditautkan ke toko kebijakan Anda atau dari penyedia OpenID Connect (OIDC) lainnya.
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 SDK. Untuk menyertakan Izin Terverifikasi dalam aplikasi, integrasikan AWS SDK untuk bahasa pilihan Anda ke dalam paket aplikasi.

Untuk mempelajari lebih lanjut dan mengunduh AWS SDK, lihat Alat untuk Amazon Web Services.

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

- 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 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

Menambahkan konteks

Konteks adalah informasi yang relevan dengan keputusan kebijakan, tetapi bukan bagian dari identitas kepala sekolah, tindakan, atau sumber daya Anda. 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": {
      "MfaAuthorized": {
         "boolean": true
      },
      "AccountCodes": {
         "set": [
            {
               "long": 111122223333
            },
            {
               "long": 444455556666
            },
            {
               "long": 123456789012
            }
         ]
      },
      "UserAgent": {
         "string": "My UserAgent 1.12"
      },
      "RequestedOrderCount": {
         "long": 4
      },
      "NetworkInfo": {
         "record": {
```

```
            "IPAddress": {
               "string": "192.0.2.178"
            },
            "Country": {
               "string": "United States of America"
            },
            "SSL": {
               "boolean": true
            }
         }
      },
      "approvedBy": {
         "entityIdentifier": {
            "entityId": "Bob",
            "entityType": "DigitalPetStore::User"
         }
      }
   }
}
```

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 dari jenis 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 DigitalPetStoreaplikasi 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, DigitalPetStorebukan 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.

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

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

Untuk mereferensikan tipe `record` data yang dinamai `NetworkInfo` dalam konteks permintaan Anda, buat konstruksi CommonType dalam skema Anda sebagai berikut. `commonType`Konstruk adalah kumpulan atribut bersama yang dapat Anda terapkan ke entitas yang berbeda.

> **Note**
>
> Editor skema visual Izin Terverifikasi saat ini tidak mendukung `commonType` konstruksi. Ketika Anda menambahkannya ke skema Anda, Anda tidak dapat lagi melihat skema Anda dalam mode Visual.

```
"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.MfaAuthorized == true &&
    context.UserAgent like "*My UserAgent*" &&
    context.RequestedOrderCount <= 4 &&
    context.AccountCodes.contains(111122223333) &&
    context.NetworkInfo.Country like "*United States*" &&
    context.NetworkInfo.SSL == true &&
    context.NetworkInfo.IPAddress like "192.0.2.*" &&
    context.approvedBy in DigitalPetStore::Role::"Employee"
};
```

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 string`My UserAgent`.
3. Pengguna harus meminta untuk melihat 4 atau lebih sedikit pesanan.
4. Salah satu kode akun pengguna harus`111122223333`.
5. Alamat IP pengguna harus berasal dari Amerika Serikat, mereka harus berada 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 peran`Employee`.

```
{
   "principal": {
      "entityType": "DigitalPetStore::User",
      "entityId": "Alice"
   },
   "action": {
      "actionType": "DigitalPetStore::Action",
      "actionId": "GetOrder"
   },
   "resource": {
      "entityType": "DigitalPetStore::Order",
      "entityId": "1234"
   },
   "context": {
     "contextMap": {
      "MfaAuthorized": {
       "boolean": true
      },
      "UserAgent": {
       "string": "My UserAgent 1.12"
      },
      "RequestedOrderCount":{
       "long": 4
      },
      "AccountCodes": {
```

```
       "set": [
        {"long": 111122223333},
        {"long": 444455556666},
        {"long": 123456789012}
        ]
      },
      "NetworkInfo": {
        "record": {
          "IPAddress": {"string": "192.0.2.178"},
          "Country": {"string": "United States of America"},
          "SSL": {"boolean": true}
        }
      },
      "approvedBy": {
        "entityIdentifier": {
         "entityId": "Bob",
         "entityType": "DigitalPetStore::User"
        }
      }
     }
  },
  "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": "PSEXAMPLEabcdefg111111"
}
```

Keamanan di Izin Terverifikasi Amazon

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

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. Model tanggung jawab bersama menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan dari cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di Cloud AWS Cloud. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi secara berkala efektivitas keamanan kami sebagai bagian dari Program Kepatuhan AWS. Untuk mempelajari tentang program kepatuhan yang berlaku untuk Izin Terverifikasi Amazon, lihatAWSLayanan dalam Lingkup oleh Program Kepatuhan.
- Keamanan di cloud – Tanggung jawab Anda ditentukan menurut layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum 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 belajar cara menggunakanAWSlayanan yang membantu Anda memantau dan mengamankan sumber daya Izin Terverifikasi Anda.

Topik

Perlindungan data di Izin Terverifikasi Amazon

YangAWS model tanggung jawab bersamaberlaku untuk perlindungan data di Izin Terverifikasi Amazon. Sebagaimana diuraikan dalam model ini, AWS bertanggung jawab untuk memberikan

perlindungan terhadap infrastruktur global yang menjalankan semua AWS Cloud. Anda harus bertanggung jawab untuk memelihara kendali terhadap konten yang di-hosting pada infrastruktur ini. Konten ini meliputi konfigurasi keamanan dan tugas-tugas pengelolaan untuk berbagai layanan Layanan AWS yang Anda gunakan. Untuk informasi lebih lanjut tentang privasi data, lihat FAQ tentang Privasi Data. Untuk informasi tentang perlindungan data di Eropa, lihat postingan blog Model Tanggung Jawab Bersama AWS dan GDPR di Blog Keamanan AWS.

- Untuk tujuan perlindungan data, kami menyarankan Anda untuk melindungiAkun AWSkredensi dan mengatur pengguna individu denganAWS IAM Identity CenteratauAWS Identity and Access Management(IAM). Dengan cara tersebut, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tugas pekerjaan mereka.
- Kami menyarankan Anda mengamankan data Anda dengan cara berikut:
  - Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
  - Gunakan SSL/TLS untuk melakukan komunikasi dengan sumber daya AWS. Kami membutuhkan TLS 1.2.
  - Menyiapkan API dan log aktivitas pengguna dengan AWS CloudTrail.
  - GunakanAWSsolusi enkripsi, bersama dengan semua kontrol keamanan default di dalamnyaLayanan AWS.
  - Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
  - Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi lebih lanjut tentang titik akhir FIPS yang tersedia, lihat Standar Pemrosesan Informasi Federal (FIPS) 140-2.
- Kami sangat menyarankan agar Anda tidak memasukkan informasi rahasia atau sensitif, seperti alamat email pelanggan Anda, ke dalam tag atau bidang teks bentuk bebas sepertiNamabidang. Ini termasuk saat Anda bekerja dengan Izin Terverifikasi atau lainnyaLayanan AWSmenggunakan konsol, API,AWS CLI, atauAWSSDK. Setiap data yang Anda masukkan ke dalam tag atau kolom teks formulir bebas yang digunakan untuk nama dapat digunakan untuk penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, sebaiknya Anda tidak menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.
- 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`janeataubob`untuk nama entitas tipe`User`. Namun, dalam sistem produksi, sangat penting

untuk alasan keamanan bahwa Anda menggunakan nilai unik yang tidak dapat digunakan kembali. Sebaiknya gunakan nilai seperti pengenal unik universal (UUID). Misalnya, pertimbangkan penggunajaneyang meninggalkan perusahaan. Kemudian, Anda membiarkan orang lain menggunakan namajane. Pengguna baru itu mendapat akses secara otomatis ke segala sesuatu yang diberikan oleh kebijakan yang masih referensiUser::"jane". Izin Terverifikasi dan Cedar tidak dapat membedakan antara pengguna baru dan pengguna sebelumnya.

Panduan ini berlaku untuk pengidentifikasi pokok 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 pengenal lama dalam kebijakan.

- Pastikan bahwa string yang Anda berikan untuk mendefinisikanLongdanDecimalnilai berada dalam kisaran yang valid dari setiap jenis. Juga, pastikan bahwa penggunaan operator aritmatika apa pun tidak menghasilkan nilai di luar rentang yang valid. Jika rentang terlampaui, operasi menghasilkan pengecualian luapan. Kebijakan yang mengakibatkan kesalahan diabaikan, yang berarti bahwa kebijakan Izin mungkin tiba-tiba gagal mengizinkan akses, atau kebijakan Larang mungkin tiba-tiba gagal memblokir akses.

Enkripsi data

Izin Terverifikasi Amazon secara otomatis mengenkripsi semua data pelanggan seperti kebijakan denganKunci yang dikelola AWS, sehingga penggunaan kunci yang dikelola pelanggan tidak diperlukan atau didukung.

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

- Cara kerja Izin Terverifikasi Amazon IAM
- Contoh kebijakan berbasis identitas untuk Izin Terverifikasi Amazon
- Memecahkan masalah identitas dan akses Izin Terverifikasi Amazon

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Izin Terverifikasi.

Pengguna layanan — Jika Anda menggunakan layanan Izin Terverifikasi untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Izin Terverifikasi untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Izin Terverifikasi, lihatMemecahkan masalah identitas dan akses Izin Terverifikasi Amazon.

Administrator layanan — Jika Anda bertanggung jawab atas sumber daya Izin Terverifikasi di perusahaan Anda, Anda mungkin memiliki akses penuh ke Izin Terverifikasi. Tugas Anda adalah menentukan fitur dan sumber daya Izin Terverifikasi mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke IAM administrator Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang cara perusahaan Anda dapat menggunakan IAM Izin Terverifikasi, lihatCara kerja Izin Terverifikasi Amazon IAM.

IAM administrator — Jika Anda seorang IAM administrator, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Izin Terverifikasi. Untuk melihat contoh kebijakan berbasis identitas Izin Terverifikasi yang dapat Anda gunakan, lihat. IAMContoh kebijakan berbasis identitas untuk Izin Terverifikasi Amazon

Mengautentikasi dengan identitas

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

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas

federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan IAM peran. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

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

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

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

Akun AWS pengguna root

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

Identitas gabungan

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

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses

Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensi jangka panjang di IAM Panduan Pengguna.

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

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari lebih lanjut, lihat Kapan membuat pengguna IAM (bukan peran) 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 AWS Management Console 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 kredensyal 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 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 kredensional 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 adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Ringkasan kebijakan JSON di IAM Panduan Pengguna.

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

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

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

Kebijakan berbasis identitas

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 Membuat IAM kebijakan di Panduan Pengguna.IAM

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

Kebijakan berbasis sumber daya

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 prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

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 (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL) dalam Panduan Developer Amazon Simple Storage Service.

Jenis-jenis kebijakan lain

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

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

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

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

Berbagai jenis kebijakan

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

Cara kerja 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
ACL	Tidak

IAM fitur	Dukungan Izin Terverifikasi
ABAC (tanda dalam kebijakan)	Tidak
Kredensial sementara	Ya
Izin prinsipal	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 Membuat IAM kebijakan di Panduan Pengguna.IAM

Dengan kebijakan IAM berbasis identitas, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam 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 prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. 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. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, IAM administrator di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun IAM di Panduan IAM Pengguna.

Tindakan kebijakan untuk Izin Terverifikasi

Mendukung tindakan kebijakan	Ya

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

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

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

Untuk melihat daftar tindakan 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. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

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

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

Untuk melihat daftar jenis sumber daya Izin Terverifikasi dan ARNnya, 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. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

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

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

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat elemen IAM kebijakan: variabel dan tag di Panduan IAM Pengguna.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan IAM Pengguna.

ACL dalam Izin Terverifikasi

Mendukung ACL	Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Izin Terverifikasi

Mendukung ABAC (tanda dalam kebijakan)	Tidak

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke IAM entitas (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tag milik prinsipal cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

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

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

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

Menggunakan kredensyal sementara dengan Izin Terverifikasi

Mendukung penggunaan kredensial sementara	Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensyal sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensil sementara, lihat Layanan AWS yang berfungsi IAM di IAM Panduan Pengguna.

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

Anda dapat membuat kredensyal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensyal sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk Izin Terverifikasi

Mendukung izin pengguna utama	Ya

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

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, silakan lihat Membuat sebuah 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.

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 AWS Management Console, 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 ARN 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 Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat kebijakan AWSAWS terkelola atau kebijakan terkelola untuk fungsi pekerjaan di Panduan IAM Pengguna.
- Menerapkan izin hak istimewa paling sedikit — Saat Anda menetapkan izin dengan IAM kebijakan, berikan hanya izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat Kebijakan dan izin IAM di IAM Panduan Pengguna.
- Gunakan ketentuan dalam IAM kebijakan untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. 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 AWS 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 validasi kebijakan 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 Mengonfigurasi akses API yang dilindungi MFA di Panduan Pengguna.IAM

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:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
```

```
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

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 (ACL), 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, lihatCara 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. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

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

> **Note**
>
> Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat Referensi Layanan yang Memenuhi Syarat HIPAA.

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

Ketahanan dalam Izin Terverifikasi Amazon

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan jaringan yang sangat berlebihan. Dengan Availability Zone, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis mengalami fail over antar 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 toko kebijakan Izin Terverifikasi, toko tersebut dibuat dalam individuWilayah AWS, 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 Global AWS.

Memantau Izin Memantau Amazon Memantau Amazon Verified

Memantau adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan kinerja Amazon Verified Perizinan dan AWS solusi Anda lainnya. AWSmenyediakan alat pemantauan berikut untuk mengawasi Izin, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan:

- AWS CloudTrail merekam panggilan API dan peristiwa terkait yang dilakukan oleh atau atas nama akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang memanggil AWS, alamat IP sumber yang melakukan panggilan, dan kapan panggilan tersebut terjadi. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.

Mencatat panggilan API Izin Terverifikasi Amazon menggunakan AWS CloudTrail

Izin Terverifikasi Amazon terintegrasi denganAWS 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 bucket Amazon S3, termasuk peristiwa untuk Izin Terverifikasi. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk 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 AndaAkun AWS, termasuk acara untuk Izin Terverifikasi, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- 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 ke`CreateIdentitySource`,`DeletePolicy`, dan `ListPolicyStores` tindakan menghasilkan entri dalam file CloudTrail log.

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

- Baik permintaan tersebut dibuat dengan kredensial pengguna akar atau AWS Identity and Access Management (IAM).
- Jika permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Apakah permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Peristiwa data seperti IsAuthorizeddan IsAuthorizedWithTokentidak 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 bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

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": "PSEXAMPLEabcdefg111111"
    },
    "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": "PSEXAMPLEabcdefg111111"
    },
    "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": "PSEXAMPLEabcdefg111111",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-22T07:43:33.962794Z",
    "lastUpdatedDate": "2023-05-22T07:43:33.962794Z"
  },
  "requestID": "1dd9360e-e2dc-4554-ab65-b46d2cf45c29",
  "eventID": "b6edaeee-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-TOKEN1111111",
    "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": "PSEXAMPLEabcdefg111111"
    },
  "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/
PSEXAMPLEabcdefg111111"
    }
  ],
  "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": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "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/
PSEXAMPLEabcdefg111111"
    }
  ],
  "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": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "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/
PSEXAMPLEabcdefg111111"
    }
  ],
  "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 AWS CloudFormation ketentuan serta mengonfigurasi sumber daya tersebut untuk Anda.

Saat menggunakannya AWS 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 AWS CloudFormation terkait Identitas Amazon Cognito, sepertiUnrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient, kami menyarankan Anda membuat kumpulan pengguna dan klien Amazon Cognito di lokasi terdekat secara geografis Wilayah AWS tempat Identitas Amazon Cognito tersedia. Gunakan kumpulan pengguna yang baru dibuat ini saat membuat sumber identitas Izin Terverifikasi.

Izin dan AWS CloudFormation templat terverifikasi

Untuk menyediakan dan mengonfigurasi sumber daya untuk Izin Terverifikasi dan layanan terkait, Anda harus memahami AWS CloudFormation templat. Templat adalah file teks dengan format JSON atau YAML. Template ini menjelaskan sumber daya yang ingin Anda sediakan di AWS CloudFormation tumpukan Anda. Jika Anda tidak terbiasa dengan JSON atau YAMB, Anda dapat menggunakan AWS CloudFormation Designer untuk membantu Anda memulai dengan template. AWS CloudFormation Untuk informasi lebih lanjut, lihat Apa itu AWS CloudFormation Desainer? dalam AWS CloudFormation User Guide.

Izin Terverifikasi mendukung pembuatan sumber identitas, kebijakan, penyimpanan kebijakan, dan templat kebijakan di AWS 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. AWS CloudFormation Konstruksi, atau komponen cloud yang dapat digunakan kembali, dapat digunakan untuk membuat AWS 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 AWS CloudFormation

Untuk mempelajari selengkapnya AWS CloudFormation, lihat sumber daya berikut:

- AWS CloudFormation
- AWS CloudFormation Panduan Pengguna
- AWS CloudFormation Referensi API
- AWS CloudFormation Panduan Pengguna Antarmuka Baris Perintah

Mengakses Izin Terverifikasi Amazon menggunakan titik akhir antarmuka () AWS PrivateLink

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi privat antara VPC dan Amazon Verified Permissions. Anda dapat mengakses Verified Permissions seolah-olah VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau AWS Direct Connect koneksi. Instans dalam VPC Anda tidak memerlukan alamat IP publik untuk mengakses Izin Terverifikasi.

Anda membuat koneksi pribadi ini dengan membuat endpoint antarmuka, didukung olehAWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk endpoint antarmuka. Ini adalah antarmuka jaringan yang dikelola permintaan 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 Pertimbangan Verified

Sebelum menyiapkan titik akhir antarmuka untuk Izin Terverifikasi, tinjau Pertimbangan dalam Panduan. AWS PrivateLink

Izin Verified mendukung panggilan ke semua tindakan API melalui titik akhir antarmuka.

Kebijakan VPC endpoint tidak mendukung untuk Izin Terverifikasi. Secara default, akses penuh ke Titik akhir antarmuka 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.

Membuat titik akhir antarmuka untuk Izin Terverifikasi

Anda dapat membuat titik akhir antarmuka untuk Izin Terverifikasi menggunakan konsol Amazon VPC atau (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di AWS PrivateLinkPanduan.

Membuat titik akhir antarmuka untuk Izin Terverifikasi menggunakan nama layanan berikut:

```
com.amazonaws.region.verifiedpermissions
```

Jika Anda mengaktifkan DNS privat untuk titik akhir antarmuka, Anda dapat membuat permintaan API untuk Izin Terverifikasi menggunakan nama DNS Regional. Misalnya, `verifiedpermissions.us-east-1.amazonaws.com`.

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

Nama	Default	Dapat disesuaikan	Deskripsi
Toko kebijakan per Wilayah per akun	Setiap Wilayah yang didukung: 1.000	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 disesu an	Deskripsi
			dapat Anda tentukan untuk toko kebijakan.
Ukuran permintaan otorisasi[1]	1 MB	Tidak	Ukuran maksimum permintaan otorisasi.
Ukuran kebijakan	10.000 byte	Tidak	Ukuran maksimum kebijakan individu.
Ukuran skema	100.000 bita	Tidak	Ukuran maksimum skema toko kebijakan.
Ukuran kebijakan per sumber daya	200.000 bytes[2]	Tidak	Ukuran maksimum semua kebijakan yang mereferensikan sumber daya tertentu.

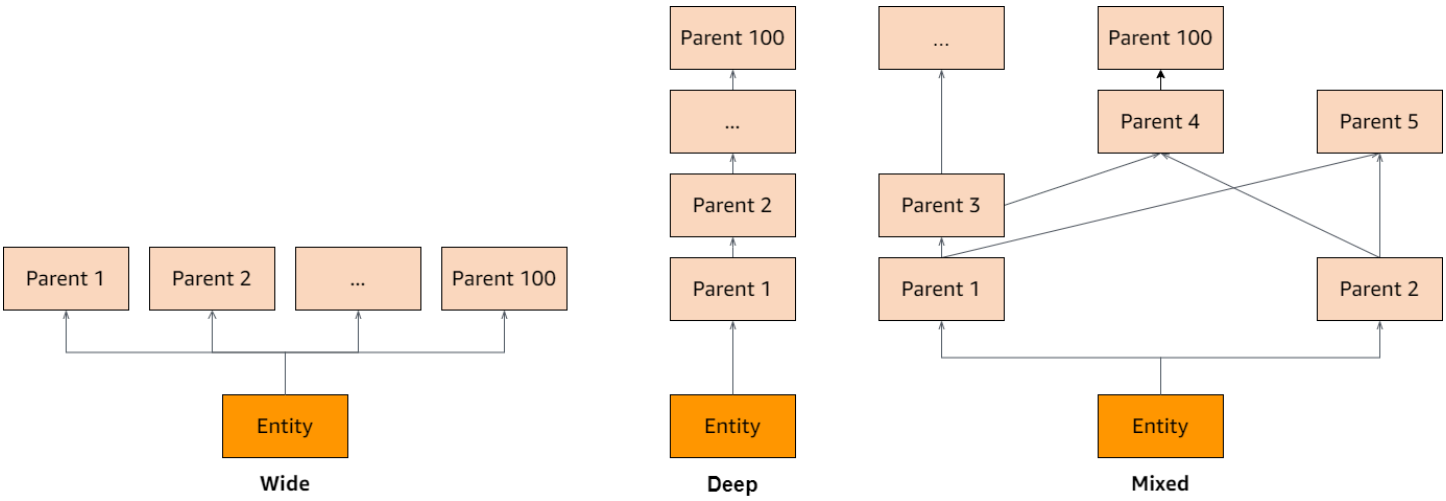
[1] Kuota untuk permintaan otorisasi adalah sama untuk keduanya dan IsAuthorized. IsAuthorizedWithToken

[2] 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.

Kuota untuk hierarki

Nama	Default	Dapat disesuaikan	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 Wilayah AWS 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 disesuaikan	Deskripsi
BatchIsAuthorized permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 30	Ya	Jumlah maksimum BatchIsAuthorized permintaan per detik.
BatchIsAuthorizedWithToken permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 30	Ya	Jumlah maksimum BatchIsAuthorizedWithToken permintaan per detik.
CreatePolicy permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum CreatePolicy permintaan per detik.
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 akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum CreatePolicyTemplate permintaan per detik.
DeletePolicy permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum DeletePolicy permintaan per detik.
DeletePolicyStore permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum DeletePolicyStore permintaan per detik.

Nama	Default	Dapat disesuaikan	Deskripsi
DeletePolicyTemplate permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum DeletePolicyTemplate permintaan per detik.
GetPolicy permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum GetPolicy permintaan per detik.
GetPolicyTemplate permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum GetPolicyTemplate permintaan per detik.
GetSchema permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum GetSchema permintaan per detik.
IsAuthorized permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 200	Ya	Jumlah maksimum IsAuthorized permintaan per detik.
IsAuthorizedWithToken permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 200	Ya	Jumlah maksimum IsAuthorizedWithToken permintaan per detik.
ListPolicies permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum ListPolicies permintaan per detik.
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 akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum ListPolicyTemplates permintaan per detik.

Nama	Default	Dapat disesuaikan	Deskripsi
PutSchema permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum PutSchema permintaan per detik.
UpdatePolicy permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum UpdatePolicy permintaan per detik.
UpdatePolicyTemplate permintaan per detik per Wilayah per akun	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum UpdatePolicyTemplate permintaan per detik.

Riwayat dokumen untuk Panduan Pengguna Izin Terverifikasi Amazon

Tabel berikut menjelaskan rilis dokumentasi untuk Izin Terverifikasi.

Perubahan	Deskripsi	Tanggal
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 pengguna Amazon Cognito dalam `BatchIsAuthorizedWithToken` satu permintaan API.	April 5, 2024
Membuat toko kebijakan dengan API Gateway	Anda sekarang dapat membuat toko kebijakan dari API yang ada dan kumpulan pengguna Amazon Cognito.	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, penyimpan an kebijakan, dan templat	Juni 30, 2023

	kebijakan di AWS CloudFormation.	
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.