aws

Guida per l'utente

Autorizzazioni verificate da Amazon

Autorizzazioni verificate da Amazon: Guida per l’utente

Table of Contents

Che cos'è Amazon Verified Permissions?

Amazon Verified Permissions è un servizio di gestione e autorizzazione scalabile e granulare delle autorizzazioni per applicazioni personalizzate create da te. Verified Permissions consente ai tuoi sviluppatori di creare applicazioni sicure più rapidamente esternalizzando le autorizzazioni e centralizzando la gestione e l'amministrazione delle policy. Verified Permissions utilizza il linguaggio di policy Cedar per definire autorizzazioni dettagliate per proteggere le risorse dell'applicazione.

Per indicazioni ed esempi sulla configurazione di un punto di decisione delle politiche (PDP) utilizzando le autorizzazioni verificate, consulta Implementazione di un PDP utilizzando Amazon Verified Permissions in Prescriptive Guidance.AWS

Argomenti

Autorizzazione nelle autorizzazioni verificate

Verified Permissions fornisce l'autorizzazione verificando se un principale è autorizzato a eseguire un'azione su una risorsa in un determinato contesto dell'applicazione. Verified Permissions presuppone che il principale sia stato precedentemente identificato e autenticato con altri mezzi, ad esempio utilizzando protocolli come OpenID Connect, un provider Amazon Cognito ospitato o un'altra soluzione di autenticazione. Verified Permissions non dipende da dove viene gestito il principale e dal modo in cui è stato autenticato.

Verified Permissions è un servizio che consente ai clienti di creare, mantenere e testare le policy in modo programmatico utilizzando le Console di gestione AWS Autorizzazioni Verificate o attraverso l'infrastruttura APIs, ad esempio soluzioni di codice. CloudFormation Le autorizzazioni sono espresse utilizzando il linguaggio di policy Cedar. L'applicazione client richiede l'autorizzazione APIs per valutare le politiche Cedar archiviate con il servizio e fornire una decisione di accesso sull'opportunità o meno di un'azione.

Linguaggio delle politiche Cedar

Le politiche di autorizzazione in Verified Permissions sono scritte utilizzando il linguaggio di policy Cedar. Cedar è un linguaggio open source per scrivere politiche di autorizzazione e prendere decisioni di autorizzazione basate su tali politiche. Quando crei un'applicazione, devi assicurarti che solo i responsabili autorizzati, gli utenti umani o le macchine possano accedere all'applicazione e possano fare solo ciò a cui sono autorizzati a fare. Utilizzando Cedar, è possibile disaccoppiare la logica aziendale dalla logica di autorizzazione. Nel codice dell'applicazione, inserite come prefazione alle vostre operazioni una chiamata al motore di autorizzazione Cedar, con la domanda «Questa richiesta è autorizzata?». Quindi, l'applicazione può eseguire l'operazione richiesta se la decisione è «consentire» o restituire un messaggio di errore se la decisione è «negare».

Verified Permissions attualmente utilizza la versione 4.7 di Cedar.

Per ulteriori informazioni su Cedar, consulta quanto segue:

- Guida di riferimento al linguaggio delle politiche Cedar
- Deposito Cedar GitHub

Vantaggi delle autorizzazioni verificate

Accelera lo sviluppo delle applicazioni

Accelera lo sviluppo delle applicazioni separando l'autorizzazione dalla logica aziendale.

Verified Permissions fornisce integrazioni con i framework di sviluppo più diffusi, semplificando l'implementazione dell'autorizzazione nelle applicazioni con modifiche minime al codice. Queste integrazioni ti consentono di concentrarti sulla logica aziendale principale, mentre Verified Permissions gestisce le decisioni di autorizzazione.

- Express.js: un'integrazione basata sul middleware che consente di proteggere gli endpoint API nelle applicazioni Express senza modificare i gestori di route esistenti. Per ulteriori informazioni, consulta the section called “Utilizzo di Express”.

Applicazioni più sicure

Le autorizzazioni verificate consentono agli sviluppatori di creare applicazioni più sicure.

Funzionalità per l'utente finale

Le autorizzazioni verificate consentono di fornire agli utenti finali funzionalità più complete per la gestione delle autorizzazioni.

Servizi correlati

- Amazon Cognito — Amazon Cognito è una piattaforma di identità per app Web e mobili. È un elenco utenti, un server di autenticazione e un servizio di autorizzazione per token e credenziali di accesso OAuth 2.0. AWS Quando crei un policy store, hai la possibilità di creare i tuoi principali e gruppi da un Amazon Cognito pool di utenti. Per ulteriori informazioni, consulta la Guida per sviluppatori di Amazon Cognito .
- Amazon API Gateway — Amazon API Gateway è un AWS servizio per la creazione, la pubblicazione, la manutenzione, il monitoraggio e la protezione di REST, HTTP e WebSocket APIs su qualsiasi scala. Quando crei un policy store, hai la possibilità di creare azioni e risorse a API Gateway partire da un'API. Per ulteriori informazioni in merito API Gateway, consulta la Guida per API Gateway gli sviluppatori.
- AWS IAM Identity Center— Con IAM Identity Center, puoi gestire la sicurezza degli accessi per le identità della tua forza lavoro, note anche come utenti della forza lavoro. IAM Identity Center offre un posto in cui è possibile creare o connettere gli utenti della forza lavoro e gestire centralmente il loro accesso a tutte le loro applicazioni. Account AWS Per ulteriori informazioni, consulta la Guida per l'utente AWS IAM Identity Center.

Accesso alle autorizzazioni verificate

Puoi utilizzare Amazon Verified Permissions in uno dei seguenti modi.

Console di gestione AWS

La console è un'interfaccia basata su browser per gestire le autorizzazioni e le risorse verificate. AWS Per ulteriori informazioni sull'accesso alle autorizzazioni verificate tramite la console, consulta Come accedere alla Guida per l' AWSutente.Accedi ad AWS

- Console Amazon Verified Autorizzations

AWS Strumenti da riga di comando

È possibile utilizzare gli strumenti della riga di AWS comando per impartire comandi dalla riga di comando del sistema per eseguire autorizzazioni e AWS attività verificate. L'utilizzo della riga di

comando può essere più veloce e semplice rispetto all'uso della console. Gli strumenti a riga di comando sono inoltre utili per creare script che eseguono le attività di AWS .

AWS fornisce due set di strumenti da riga di comando: the AWS Command Line Interface(AWS CLI) e the AWS Tools for Windows PowerShell. Per informazioni sull'installazione e l'utilizzo di AWS CLI, consulta la Guida AWS Command Line Interface per l'utente. Per informazioni sull'installazione e l'utilizzo degli strumenti per Windows PowerShell, consulta la Guida per AWS Strumenti per PowerShell l'utente.

- verifiedpermissions nel Command Reference AWS CLI
- Autorizzazioni verificate da Amazon in AWS Tools for Windows PowerShell

AWS SDKs

AWS fornisce SDKs (kit di sviluppo software) costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Python, Ruby, .NET, iOS, Android, ecc.). SDKs Forniscono un modo conveniente per creare l'accesso programmatico alle autorizzazioni verificate e. AWS Ad esempio, SDKs si occupano di attività come la firma crittografica delle richieste, la gestione degli errori e il ritentativo automatico delle richieste.

Per ulteriori informazioni e per il download AWS SDKs, consulta Strumenti per. Amazon Web Services

Di seguito sono riportati i collegamenti alla documentazione relativa alle risorse relative alle autorizzazioni verificate in varie AWS SDKs aree.

- AWS SDK per .NET
- AWS SDK per C++
- AWS SDK per Go
- AWS SDK per Java
- AWS SDK per JavaScript
- AWS SDK per PHP
- AWS SDK per Python (Boto)
- AWS SDK per Ruby
- AWS SDK per Rust

AWS Costrutti CDK

AWS Cloud Development Kit (AWS CDK) È un framework di sviluppo software open source per definire l'infrastruttura cloud in codice e fornirla tramite. CloudFormation I costrutti, o componenti

cloud riutilizzabili, possono essere utilizzati per creare modelli. CloudFormation Questi modelli possono quindi essere utilizzati per implementare l'infrastruttura cloud.

Per saperne di più e scaricare AWS CDK, consulta AWS Cloud Development Kit.

Di seguito sono riportati i collegamenti alla documentazione relativa alle AWS CDK risorse relative alle autorizzazioni verificate, ad esempio i costrutti.

- Autorizzazioni verificate Amazon L2 CDK Construct

API per le autorizzazioni verificate

Puoi accedere alle autorizzazioni verificate e in modo AWS programmatico utilizzando l'API Verified Permissions, che consente di inviare richieste HTTPS direttamente al servizio. Quando utilizzi le API , devi includere il codice per firmare in modo digitale le richieste utilizzando le tue credenziali.

- Guida di riferimento all'API Amazon Verified Permissions

Prezzi delle autorizzazioni verificate

Verified Permissions offre prezzi differenziati in base al numero di richieste di autorizzazione mensili inviate dalle richieste di autorizzazione alle autorizzazioni verificate. Sono inoltre previsti prezzi per le azioni di gestione delle politiche in base alla quantità di richieste API delle policy cURL (URL client) inviate ogni mese dalle tue applicazioni a Verified Permissions.

Per un elenco completo dei costi e dei prezzi per le autorizzazioni verificate, consulta i prezzi di Amazon Verified Permissions.

Per vedere la tua fattura, vai sul Pannello di controllo di gestione dei costi e della fatturazione nella console Gestione dei costi e fatturazione AWS. La fattura contiene collegamenti per passare ai report di utilizzo, che consentono di visualizzare i dettagli della fattura. Per ulteriori informazioni sulla Account AWS fatturazione, consulta la Guida per l'AWS Billing utente.

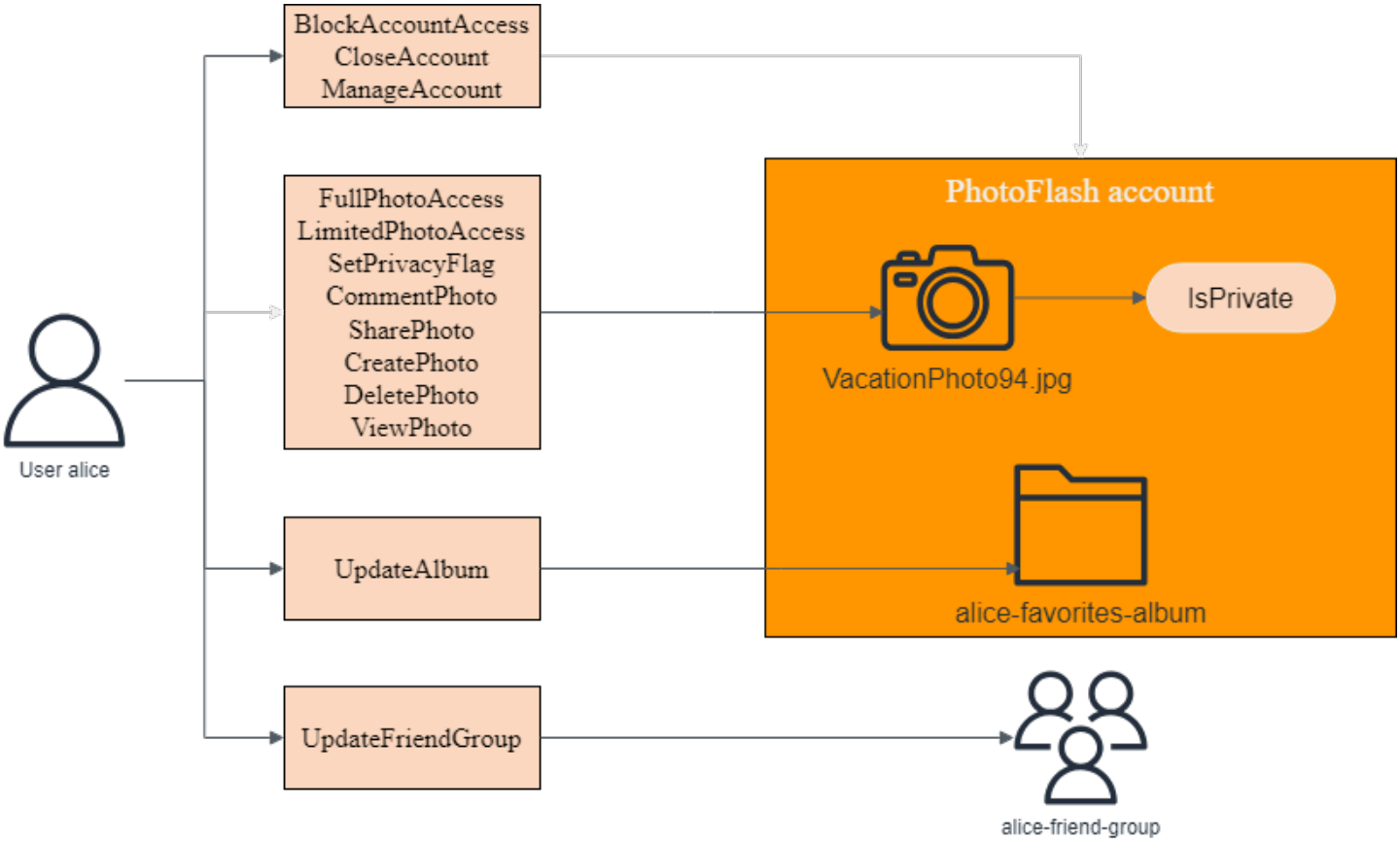
Se hai domande relative alla AWS fatturazione, agli account e agli eventi, contatta. Supporto

Crea il tuo primo Amazon Verified Permissions Policy Store

Per questo tutorial, supponiamo che tu sia lo sviluppatore di un'applicazione per la condivisione di foto e che tu stia cercando un modo per controllare le azioni che gli utenti dell'applicazione possono eseguire. Vuoi controllare chi può aggiungere, eliminare o visualizzare foto e album fotografici. Vuoi anche controllare le azioni che un utente può intraprendere sul proprio account. Possono gestire il proprio account, che ne dici dell'account di un amico? Per controllare queste azioni, è necessario creare politiche che consentano o vietino tali azioni in base all'identità dell'utente. Verified Permissions offre archivi di policy, o contenitori, per ospitare queste politiche.

In questo tutorial illustreremo come creare un archivio di policy di esempio utilizzando la console Amazon Verified Permissions. La console offre alcuni esempi di opzioni di policy store e creeremo un PhotoFlashpolicy store. Questo archivio delle norme consente ai responsabili, come gli utenti, di eseguire azioni, come la condivisione, su risorse, come foto o album.

Il diagramma seguente illustra le relazioni tra un responsabile e le azioni che può intraprendere su varie risorse, vale a dire il suo PhotoFlash account, il `VactionPhoto94.jpg` file, l'album `alice-favorites-album` di foto e il gruppo di utenti. `User::alice` `alice-friend-group`



Ora che hai una conoscenza del PhotoFlashpolicy store, creiamo il policy store ed esploriamolo.

Prerequisiti

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi Console di gestione AWScome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Accedere come utente root nella Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'IAM utente.

Crea un utente con accesso amministrativo

1. Abilita il Centro identità IAM.

   Per istruzioni, consulta Abilitazione del AWS IAM Identity Center nella Guida per l’utente di AWS IAM Identity Center .
2. Nel Centro identità IAM, assegna l’accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere come utente del Centro identità IAM, utilizza l’URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l’utente del Centro identità IAM.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegnazione dell’accesso ad altri utenti

1. Nel Centro identità IAM, crea un set di autorizzazioni conforme alla best practice per l’applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l’utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l’accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l’utente di AWS IAM Identity Center .

Fase 1: Creare un archivio delle politiche PhotoFlash

Nella procedura seguente creerai un archivio PhotoFlashdelle politiche utilizzando la AWS console.

Per creare un archivio PhotoFlash delle politiche

1. Nella console Autorizzazioni verificate, scegli Crea nuovo archivio di politiche.
2. Per le opzioni di avvio, scegli Inizia da un archivio di policy di esempio.
3. Per Progetto di esempio, scegli PhotoFlash.
4. Scegli Crea archivio di politiche.

Una volta visualizzato il messaggio «Archivio delle politiche creato e configurato», scegli Vai alla panoramica per esplorare il tuo archivio delle politiche.

Fase 2: Creare una policy

Al momento della creazione del policy store, è stata creata una policy predefinita che consente agli utenti di avere il pieno controllo sui propri account. Si tratta di una politica utile, ma per i nostri scopi, creiamo una politica più restrittiva per esplorare le sfumature delle autorizzazioni verificate. Se ricordate il diagramma che abbiamo visto in precedenza nel tutorial, avevamo un presideUser::alice, che poteva eseguire un'azione, su una risorsaUpdateAlbum,. alice-favorites-album Aggiungiamo la politica che permetterà ad Alice, e solo ad Alice, di gestire questo album.

Per creare una politica

1. Nella console Autorizzazioni verificate, scegli l'archivio delle politiche che hai creato nel passaggio 1.
2. Nella navigazione, scegli Politiche.
3. Scegli Crea politica, quindi scegli Crea politica statica.
4. Per Effetto policy, scegli Permetti.
5. Per l'ambito dei principali, scegli Principal specifico, quindi per Specificare il tipo di entità, scegli PhotoFlash: :User e per Specificare l'identificatore di entità, inserisci. **alice**
6. Per l'ambito delle risorse, scegli Risorsa specifica, quindi per Specificare il tipo di entità scegli PhotoFlash: :Album e per Specificare l'identificatore di entità, inserisci. **alice-favorites-album**
7. Per l'ambito delle azioni, scegli Set di azioni specifico, quindi per Azioni a cui deve applicarsi questa politica, seleziona. UpdateAlbum
8. Scegli Next (Successivo).

9. In Dettagli, per la descrizione della politica, facoltativo, inserisci**`Policy allowing alice to update alice-favorites-album.`**.
10. Selezionare Creare policy

Ora che hai creato una policy, puoi testarla nella console Autorizzazioni verificate.

Fase 3: Test di un archivio di policy

Dopo aver creato il policy store e la policy, puoi testarli eseguendo una richiesta di autorizzazione simulata utilizzando il test bench Verified Permissions.

Per testare le politiche del Policy Store

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Test bench.
3. Scegli la modalità Visual.
4. Per Principal, procedi come segue:
    a. Per agire da Principal scegli PhotoFlash: :User e per Specificare l'identificatore di entità, inserisci. **`alice`**
    b. In Attributi, per Account: Entity, assicurati che l'entità PhotoFlash: :Account sia selezionata e per Specificare l'identificatore di entità, inserisci. **`alice-account`**
5. In Risorsa, per la risorsa su cui agisce il principale, scegli il tipo di risorsa PhotoFlash: :Album e per Specificare l'identificatore di entità, inserisci. **`alice-favorites-album`**
6. Per Azione, scegli PhotoFlash: :Action:» UpdateAlbum "dall'elenco delle azioni valide.
7. Nella parte superiore della pagina, scegli Esegui richiesta di autorizzazione per simulare la richiesta di autorizzazione per le politiche Cedar nell'archivio di policy di esempio. Il banco di prova dovrebbe mostrare Decision: Allow, che indica che la nostra politica funziona come previsto.

La tabella seguente fornisce valori aggiuntivi per il principale, la risorsa e l'azione che puoi testare con il banco di prova Verified Permissions. La tabella include la decisione sulla richiesta di autorizzazione basata sulle politiche statiche incluse nell'archivio delle politiche di PhotoFlash esempio e sulla politica creata nel passaggio 2.

Valore principale	Conto principale: valore dell'entità	Valore della risorsa	Valore principale della risorsa	Azione	Decisione di autorizzazione
PhotoFlash: :Utente | bob	PhotoFlash: :Conto | alice-account	PhotoFlash: :Album | alice-favorites-album	N/D	PhotoFlash: :Azione:: » "UpdateAlbum	Rifiuta
PhotoFlash: :Utente | alice	PhotoFlash: :Conto | alice-account	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Conto | bob-account	PhotoFlash: :Azione::» "ViewPhoto	Rifiuta
PhotoFlash: :Utente | alice	PhotoFlash: :Conto | alice-account	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Conto | alice-account	PhotoFlash: :Azione::» "ViewPhoto	Consenso
PhotoFlash: :Utente | alice	PhotoFlash: :Conto | alice-account	PhotoFlash: :Foto | bob-photo.jpeg	PhotoFlash: :Album | Bob-Vacation-Album	PhotoFlash: :Azione::» "DeletePhoto	Rifiuta

Fase 4: Eliminazione delle risorse

Dopo aver finito di esplorare il tuo archivio delle polizze, eliminalo.

Come eliminare un archivio di policy

1. Nella console Autorizzazioni verificate, scegli il policy store che hai creato nel passaggio 1.
2. Nella navigazione, scegli Impostazioni.
3. In Elimina policy store, scegli Elimina questo policy store.
4. Nel file Eliminare questo archivio di politiche? nella finestra di dialogo, inserisci delete, quindi scegli Elimina.

Procedure consigliate per la progettazione di un modello di autorizzazione

Mentre ti prepari a utilizzare il servizio Amazon Verified Permissions all'interno di un'applicazione software, può essere difficile passare immediatamente alla stesura di dichiarazioni politiche come primo passo. Sarebbe come iniziare lo sviluppo di altre parti di un'applicazione scrivendo istruzioni SQL o specifiche API prima di decidere completamente cosa fare l'applicazione. Dovreste invece iniziare con un'esperienza utente. Quindi, procedi a ritroso da quell'esperienza per arrivare a un approccio di implementazione.

Mentre svolgi questo lavoro, ti ritroverai a porre domande come:

- Quali sono le mie risorse? Come sono organizzate? Ad esempio, i file si trovano all'interno di una cartella?
- L'organizzazione delle risorse ha un ruolo nel modello di autorizzazioni?
- Quali azioni possono eseguire i responsabili su ciascuna risorsa?
- In che modo i dirigenti acquisiscono tali autorizzazioni?
- Vuoi che i tuoi utenti finali possano scegliere tra autorizzazioni predefinite come «Amministratore», «Operatore» o «ReadOnly», o devono creare dichiarazioni politiche ad hoc? O entrambi?
- I ruoli sono globali o circoscritti? Ad esempio, un «operatore» è limitato a un singolo tenant o «operatore» significa operatore nell'intera applicazione?
- Quali tipi di query sono necessarie per rendere l'esperienza utente? Ad esempio, è necessario elencare tutte le risorse a cui un principale può accedere per visualizzare la home page di quell'utente?
- Gli utenti possono impedire accidentalmente l'accesso alle proprie risorse? È necessario evitarlo?

Il risultato finale di questo esercizio è denominato modello di autorizzazione; definisce i principi, le risorse, le azioni e il modo in cui interagiscono tra loro. La produzione di questo modello non richiede una conoscenza esclusiva di Cedar o del servizio Verified Permissions. Si tratta invece innanzitutto di un esercizio di progettazione dell'esperienza utente, molto simile a qualsiasi altro, e può manifestarsi in artefatti come prototipi di interfaccia, diagrammi logici e una descrizione generale di come le autorizzazioni influiscono su ciò che gli utenti possono fare nel prodotto. Cedar è progettato per essere sufficientemente flessibile da soddisfare i clienti secondo un modello, anziché forzare il modello a piegarsi in modo innaturale per conformarsi all'implementazione di Cedar. Di conseguenza,

acquisire una comprensione approfondita dell'esperienza utente desiderata è il modo migliore per arrivare a un modello ottimale.

Per contribuire a rispondere alle domande e giungere a un modello ottimale, procedi come segue:

- Consulta i modelli di progettazione di Cedar nella Guida di riferimento al linguaggio delle politiche Cedar.
- Prendi in considerazione le migliori pratiche nella Guida di riferimento al linguaggio delle politiche Cedar.
- Considerate le migliori pratiche incluse in questa pagina.

Best practice

- Non esiste un modello canonico «corretto»
- Restituisce 403 errori proibiti anziché 404 errori non trovati
- Concentrati sulle tue risorse oltre alle operazioni API
- Considerazioni sulla multi-tenancy

Non esiste un modello canonico «corretto»

Quando si progetta un modello di autorizzazione, non esiste un'unica risposta corretta. Applicazioni diverse possono utilizzare efficacemente modelli di autorizzazione diversi per concetti simili, e questo va bene. Si consideri ad esempio la rappresentazione del file system di un computer. Quando create un file in un sistema operativo simile a Unix, questo non eredita automaticamente le autorizzazioni dalla cartella principale. Al contrario, in molti altri sistemi operativi e nella maggior parte dei servizi di condivisione di file online, i file ereditano le autorizzazioni dalla cartella principale. Entrambe le scelte sono valide a seconda delle circostanze per cui l'applicazione è ottimizzata.

La correttezza di una soluzione di autorizzazione non è assoluta, ma deve essere vista in termini di come offre l'esperienza che i clienti desiderano e se protegge le loro risorse nel modo in cui si aspettano. Se il tuo modello di autorizzazione soddisfa questo obiettivo, allora ha successo.

Ecco perché iniziare la progettazione con l'esperienza utente desiderata è il prerequisito più utile per la creazione di un modello di autorizzazione efficace.

Restituisce 403 errori proibiti anziché 404 errori non trovati

È preferibile restituire un errore 403 Forbidden alle richieste che includono un'entità, in particolare una risorsa, che non corrisponde a nessuna politica piuttosto che un errore 404 Not found. In questo modo si garantisce il massimo livello di sicurezza, in quanto non si rivela l'esistenza o meno di un'entità, ma si indica solo che la richiesta non soddisfa le condizioni delle policy contenute in nessuna policy dell'archivio delle politiche.

Concentrati sulle tue risorse oltre alle operazioni API

Nella maggior parte delle applicazioni, le autorizzazioni sono modellate in base alle risorse supportate. Ad esempio, un'applicazione per la condivisione di file potrebbe rappresentare le autorizzazioni come azioni che possono essere eseguite su un file o una cartella. Si tratta di un modello semplice e valido che astrae l'implementazione sottostante e le operazioni dell'API di backend.

Al contrario, altri tipi di applicazioni, in particolare i servizi Web, spesso progettano le autorizzazioni in base alle operazioni API stesse. Ad esempio, se un servizio Web fornisce un'API denominata`createThing()`, il modello di autorizzazione potrebbe definire un'autorizzazione corrispondente o un nome `action` in Cedar. `createThing` Funziona in molte situazioni e semplifica la comprensione delle autorizzazioni. Per richiamare l'`createThing`operazione, è necessaria l'autorizzazione all'`createThing`azione. Sembra semplice, vero?

Scoprirai che la procedura introduttiva nella console Verified Permissions include la possibilità di creare risorse e azioni direttamente da un'API. Si tratta di una base utile: una mappatura diretta tra il tuo archivio di policy e l'API che autorizza.

Tuttavia, man mano che svilupperete ulteriormente il modello, questo approccio incentrato sulle API potrebbe non essere adatto alle applicazioni con modelli di autorizzazione molto granulari, perché funge semplicemente da indicatore di ciò che i clienti APIs stanno realmente cercando di proteggere: i dati e le risorse sottostanti. Se più utenti APIs controllano l'accesso alle stesse risorse, può essere difficile per gli amministratori ragionare sui percorsi verso tali risorse e gestire l'accesso di conseguenza.

Ad esempio, si consideri una rubrica di utenti che contiene i membri di un'organizzazione. Gli utenti possono essere organizzati in gruppi e uno degli obiettivi di sicurezza è vietare l'individuazione dell'appartenenza ai gruppi da parte di soggetti non autorizzati. Il servizio che gestisce questa directory di utenti fornisce due operazioni API:

- `listMembersOfGroup`
- `listGroupMembershipsForUser`

I clienti possono utilizzare una di queste operazioni per scoprire l'appartenenza al gruppo. Pertanto, l'amministratore delle autorizzazioni deve ricordarsi di coordinare l'accesso a entrambe le operazioni. Ciò si complica ulteriormente se in seguito si sceglie di aggiungere una nuova operazione API per risolvere casi d'uso aggiuntivi, come i seguenti.

- `isUserInGroups`(una nuova API per verificare rapidamente se un utente appartiene a uno o più gruppi)

Dal punto di vista della sicurezza, questa API apre un terzo percorso per scoprire l'appartenenza ai gruppi, interrompendo le autorizzazioni accuratamente predisposte dell'amministratore.

Ti consigliamo di concentrarti sui dati e sulle risorse sottostanti e sulle relative operazioni di associazione. L'applicazione di questo approccio all'esempio dell'appartenenza a un gruppo porterebbe a un'autorizzazione astrattaviewGroupMembership, ad esempio, che ciascuna delle tre operazioni API deve consultare.

Nome API	Permissions	
`listMembersOfGroup`	richiede `viewGroupMembership`	l'autorizzazione per il gruppo
`listGroupMembershipsForUser`	richiede `viewGroupMembership`	l'autorizzazione dell'utente
`isUserInGroups`	richiede `viewGroupMembership`	l'autorizzazione dell'utente

Definendo quest'unica autorizzazione, l'amministratore controlla con successo l'accesso alla scoperta delle appartenenze ai gruppi, ora e per sempre. Come compromesso, ogni operazione API deve ora documentare le eventuali diverse autorizzazioni richieste e l'amministratore deve consultare questa documentazione durante la creazione delle autorizzazioni. Questo può essere un compromesso valido se necessario per soddisfare i requisiti di sicurezza.

Considerazioni sulla multi-tenancy

Potresti voler sviluppare applicazioni che possano essere utilizzate da più clienti, aziende che utilizzano la tua applicazione o tenant, e integrarle con Amazon Verified Permissions. Prima di sviluppare il modello di autorizzazione, sviluppa una strategia multi-tenant. Puoi gestire le policy dei tuoi clienti in un unico archivio di policy condiviso o assegnare a ciascuno un archivio di policy per tenant. Per ulteriori informazioni, consulta Considerazioni sulla progettazione multi-tenant di Amazon Verified Permissions in Prescriptive Guidance.AWS

1. Un unico archivio di policy condiviso

   Tutti gli inquilini condividono un unico archivio di politiche. L'applicazione invia tutte le richieste di autorizzazione all'archivio delle politiche condiviso.

2. Archivio delle politiche per tenant

   Ogni inquilino dispone di un archivio di polizze dedicato. L'applicazione interrogherà diversi archivi di policy per una decisione di autorizzazione, a seconda del tenant che effettua la richiesta.

Nessuna delle due strategie avrà un grande impatto sulla AWS bolletta. Quindi, come dovresti progettare il tuo approccio? Le seguenti sono condizioni comuni che potrebbero contribuire alla strategia di autorizzazione multi-tenant con Autorizzazioni Verificate.

Isolamento delle politiche degli inquilini

L'isolamento delle politiche di ciascun inquilino dagli altri è importante per proteggere i dati degli inquilini. Quando ogni inquilino ha il proprio archivio delle polizze, ognuno ha il proprio set isolato di politiche.

Flusso di autorizzazione

È possibile identificare un tenant che effettua una richiesta di autorizzazione inserendo un Policy Store ID nella richiesta, utilizzando archivi di policy specifici per tenant. Con un policy store condiviso, tutte le richieste utilizzano lo stesso ID del policy store.

Gestione dei modelli e degli schemi

Quando l'applicazione dispone di più archivi di policy, i modelli di policy e uno schema di policy store aggiungono un livello di sovraccarico di progettazione e manutenzione in ogni archivio di policy.

Gestione delle politiche globali

Potresti voler applicare alcune politiche globali a ogni inquilino. Il livello di spese generali per la gestione delle politiche globali varia tra i modelli di archivio delle politiche condivisi e quelli per tenant.

Disimbarco da parte degli inquilini

Alcuni inquilini apporteranno al tuo schema e alle tue politiche elementi specifici per il loro caso. Quando un inquilino non è più attivo nell'organizzazione e desiderate rimuovere i suoi dati, il livello di impegno richiesto varia a seconda del suo livello di isolamento dagli altri inquilini.

Quote di risorse di servizio

Verified Permissions prevede quote di risorse e percentuali di richieste che potrebbero influire sulla decisione relativa alla locazione multipla. Per ulteriori informazioni sulle quote, consulta Quote per le risorse.

Confronto tra archivi di policy condivisi e archivi di policy per tenant

Ogni considerazione richiede il proprio livello di impegno in termini di tempo e risorse in modelli di archivio delle politiche condivisi e relativi ai singoli inquilini.

Considerazione	Livello di impegno in un archivio di policy condiviso	Livello di impegno negli archivi di policy relativi ai singoli inquilini
Isolamento delle politiche degli inquilini	Medio.È necessario includere gli identificatori degli inquilini nelle politiche e nelle richieste di autorizzazione.	Basso. L'isolamento è un comportamento predefinito. Le politiche specifiche degli inquilini sono inaccessibili agli altri inquilini.
Flusso di autorizzazione	Basso. Tutte le interrogazioni hanno come target un archivio di policy.	Medio. Deve mantenere le mappature tra ogni tenant e il relativo ID dell'archivio delle politiche.

Modelli e gestione degli schemi	Basso. Deve far funzionar e uno schema per tutti gli inquilini.	Alto. Gli schemi e i modelli potrebbero essere meno complessi singolarmente, ma le modifiche richiedono maggiore coordinamento e complessità.
Gestione delle politiche globali	Bassa. Tutte le politiche sono globali e possono essere aggiornate centralmente.	Alto. È necessario aggiunger e politiche globali a ciascun archivio di polizze in fase di onboarding. Replica gli aggiornamenti delle policy globali tra molti archivi di policy.
Disimbarco da parte del locatario	Alto. È necessario identificare ed eliminare solo le politiche specifiche del tenant.	Basso. Eliminare l'archivio delle politiche.
Quote di risorse di servizio	Alto. I tenant condividono le quote di risorse che influisco no sugli archivi delle politiche , come la dimensione dello schema, la dimensione dei criteri per risorsa e le fonti di identità per l'archivio delle politiche.	Basso. Ogni inquilino dispone di quote di risorse dedicate.

Come scegliere

Ogni applicazione multi-tenant è diversa. Confrontate attentamente i due approcci e le relative considerazioni prima di prendere una decisione architettonica.

Se l'applicazione non richiede policy specifiche per i tenant e utilizza un'unica fonte di identità, un archivio di policy condiviso per tutti i tenant è probabilmente la soluzione più efficace. Ciò si traduce in un flusso di autorizzazione più semplice e nella gestione delle policy globali. L'eliminazione di un

tenant utilizzando un archivio di policy condiviso richiede meno sforzi perché l'applicazione non deve eliminare le politiche specifiche del tenant.

Tuttavia, se l'applicazione richiede molte policy specifiche per il tenant o utilizza più fonti di identità, è probabile che gli archivi di policy per tenant siano i più efficaci. È possibile controllare l'accesso alle politiche dei tenant con politiche che concedono autorizzazioni per tenant a IAM ciascun archivio di politiche. L'esclusione di un tenant comporta l'eliminazione del relativo archivio delle politiche; in un shared-policy-store ambiente, è necessario trovare ed eliminare le politiche specifiche del tenant.

Archivi di policy di Amazon Verified Permissions

Un policy store è un contenitore per policy e modelli di policy. In ogni policy store, è possibile creare uno schema utilizzato per convalidare le policy aggiunte al policy store. Inoltre, è possibile attivare la convalida delle politiche. Se si aggiunge una policy a un policy store con la convalida delle policy abilitata, i tipi di entità, i tipi comuni e le azioni definiti nella policy vengono convalidati rispetto allo schema e le policy non valide vengono rifiutate.

La protezione dall'eliminazione impedisce l'eliminazione accidentale di un archivio di politiche. La protezione dall'eliminazione è abilitata su tutti i nuovi policy store creati tramite Console di gestione AWS. Al contrario, è disabilitata per tutti gli archivi di policy creati tramite una chiamata API o SDK.

Consigliamo di creare un archivio di policy per applicazione o un policy store per tenant per applicazioni multi-tenant. È necessario specificare un policy store quando si effettua una richiesta di autorizzazione. È inoltre possibile creare alias di policy store per fare riferimento ai policy store con nomi descrittivi. Per ulteriori informazioni, consulta Alias dell'archivio delle policy di Amazon Verified Permissions.

Consigliamo di utilizzare namespace per le entità Cedar nei vostri archivi di policy per evitare ambiguità. Un namespace è un prefisso di stringa per un tipo, separato da una coppia di due punti () come delimitatore. `::` Ad esempio, `MyApplicationNamespace::exampleType`. Verified Permissions supporta uno spazio dei nomi per archivio di politiche. Questi namespace aiutano a mantenere le cose chiare quando lavori con più applicazioni simili. Ad esempio, nelle applicazioni multi-tenant, l'utilizzo di uno spazio dei nomi per aggiungere il nome del tenant ai tipi definiti nello schema li distinguerà dalle controparti simili utilizzate dagli altri tenant. Esaminando i log delle richieste di autorizzazione, sarete in grado di identificare facilmente il tenant che ha elaborato la richiesta di autorizzazione. Per ulteriori informazioni, consulta Namespaces nella Cedar Policy Language Reference Guide.

Argomenti

Creazione di archivi di policy per le autorizzazioni verificate

È possibile creare un archivio delle politiche utilizzando i seguenti metodi:

- Segui una configurazione guidata: definirai un tipo di risorsa con azioni valide e un tipo principale prima di creare la tua prima politica.
- Configura con API Gateway e un'origine di identità: definisci le tue entità principali con gli utenti che accedono con un provider di identità (IdP) e le tue azioni e le entità di risorse da un'API Amazon API Gateway. Consigliamo questa opzione se desideri che l'applicazione autorizzi le richieste API con l'appartenenza al gruppo degli utenti o altri attributi.
- Inizia da un esempio di policy store: scegli un esempio di policy store di progetto predefinito. Ti consigliamo questa opzione se stai imparando a conoscere le autorizzazioni verificate e desideri visualizzare e testare politiche di esempio.
- Crea un archivio delle politiche vuoto: definirai tu stesso lo schema e tutte le politiche di accesso. Consigliamo questa opzione se avete già dimestichezza con la configurazione di un policy store.

Guided setup

Per creare un policy store utilizzando il metodo di configurazione con configurazione guidata

La procedura guidata di configurazione guida l'utente attraverso il processo di creazione della prima iterazione del policy store. Creerai uno schema per il tuo primo tipo di risorsa, descriverai le azioni applicabili a quel tipo di risorsa e il tipo principale per il quale concedi le autorizzazioni. Creerai quindi la tua prima politica. Una volta completata questa procedura guidata, sarà possibile aggiungerle al proprio archivio delle politiche, estendere lo schema per descrivere altri tipi di risorse e principali e creare criteri e modelli aggiuntivi.

1. Nella console Autorizzazioni verificate, seleziona Crea nuovo archivio di politiche.
2. Nella sezione Opzioni di avvio, scegli Configurazione guidata.
3. Inserisci una descrizione del Policy store. Questo testo può essere quello che più si addice all'organizzazione come riferimento esplicito alla funzione dell'attuale archivio delle politiche, ad esempio l'applicazione web Weather updates.
4. Nella sezione Dettagli, digitate un Namespace per lo schema. Per ulteriori informazioni sui namespace, vedere. Definizione dello spazio dei nomi
5. Scegli Next (Successivo).
6. Nella finestra Tipo di risorsa, digita un nome per il tipo di risorsa. Ad esempio, `currentTemperature` potrebbe essere una risorsa per l'applicazione web Weather updates.

7. (Facoltativo) Scegliete Aggiungi un attributo per aggiungere gli attributi della risorsa. Digitate il nome dell'attributo e scegliete un tipo di attributo per ogni attributo della risorsa. Scegli se ogni attributo è obbligatorio. Ad esempio, `temperatureFormat` potrebbe essere un attributo della `currentTemperature` risorsa ed essere Fahrenheit o Celsius. Per rimuovere un attributo che è stato aggiunto per il tipo di risorsa, scegli Rimuovi accanto all'attributo.
8. Nel campo Azioni, digita le azioni da autorizzare per il tipo di risorsa specificato. Per aggiungere azioni aggiuntive per il tipo di risorsa, scegli Aggiungi un'azione. Ad esempio, `viewTemperature` potrebbe essere un'azione nell'applicazione web Weather updates. Per rimuovere un'azione che è stata aggiunta per il tipo di risorsa, scegli Rimuovi accanto all'azione.
9. Nel campo Nome del tipo principale, digita il nome di un tipo di principale che utilizzerà le azioni specificate per il tipo di risorsa. Per impostazione predefinita, l'utente viene aggiunto a questo campo ma può essere sostituito.
10. Scegli Next (Successivo).
11. Nella finestra Tipo principale, scegli la fonte di identità per il tuo tipo principale.

    - Scegli Personalizzato se l'ID e gli attributi del principale verranno forniti direttamente dall'applicazione Autorizzazioni verificate. Scegli Aggiungi un attributo per aggiungere gli attributi principali. Autorizzazioni verificate utilizza i valori degli attributi specificati per verificare le politiche rispetto allo schema. Per rimuovere un attributo che è stato aggiunto per il tipo principale, scegli Rimuovi accanto all'attributo.
    - Scegli Cognito User Pool se l'ID e gli attributi del principale verranno forniti da un ID o da un token di accesso generato da. Amazon Cognito Scegli Connect user pool. Seleziona Regione AWSe digita l'ID del pool di Amazon Cognito utenti a cui connetterti. Scegli Connetti. Per ulteriori informazioni, consulta Authorization with Amazon Verified Permissions nella Amazon Cognito Developer Guide.
    - Scegli un provider OIDC esterno se l'ID e gli attributi del principale verranno estratti da un token ID and/or Access generato da un provider OIDC esterno e aggiungi i dettagli del provider e del token.
12. Scegli Next (Successivo).
13. Nella sezione Dettagli della politica, digita una descrizione facoltativa della politica per la tua prima politica Cedar.
14. Nel campo Ambito dei principi, scegli i principali a cui verranno concesse le autorizzazioni previste dalla politica.

- Scegli Principio specifico per applicare la politica a un principio specifico. Scegli il principale nel campo Principal a cui sarà consentito intraprendere azioni e digita un identificatore di entità per il principale. Ad esempio, `user-id` potrebbe essere un identificatore di entità nell'applicazione web Weather updates.

  > **Note**
  >
  > Se si utilizza Amazon Cognito, l'identificatore di entità deve essere formattato come. `<userpool-id>|<sub>`

- Scegli Tutti i mandanti per applicare la politica a tutti i mandanti del tuo archivio delle polizze.

15. Nel campo Ambito delle risorse, scegli su quali risorse i responsabili specificati saranno autorizzati ad agire.

    - Scegli Risorsa specifica per applicare la politica a una risorsa specifica. Scegli la risorsa nel campo Risorsa a cui questo criterio dovrebbe applicarsi e digita un identificatore di entità per la risorsa. Ad esempio, `temperature-id` potrebbe essere un identificatore di entità nell'applicazione web Weather updates.
    - Scegli Tutte le risorse per applicare la politica a tutte le risorse del tuo archivio delle politiche.

16. Nel campo Ambito delle azioni, scegli le azioni che i responsabili specificati saranno autorizzati a eseguire.

    - Scegli Set specifico di azioni per applicare la politica a azioni specifiche. Seleziona le caselle di controllo accanto alle azioni nel campo Azioni a cui questo criterio dovrebbe applicarsi.
    - Scegli Tutte le azioni per applicare la politica a tutte le azioni nel tuo archivio delle politiche.

17. Consulta la politica nella sezione Anteprima della politica. Scegli Crea archivio di politiche.

Set up with API Gateway and an identity source

Per creare un policy store utilizzando il metodo di configurazione Configura con API Gateway e un metodo di configurazione Identity Source

L' API Gateway opzione protegge APIs con politiche di autorizzazione verificate progettate per prendere decisioni di autorizzazione in base ai gruppi o ai ruoli degli utenti. Questa opzione crea un archivio di politiche per testare l'autorizzazione con gruppi di origini di identità e un'API con un autorizzatore Lambda.

Gli utenti e i relativi gruppi in un IdP diventano i tuoi principali (token ID) o il tuo contesto (token di accesso). I metodi e i percorsi di un' API Gateway API diventano le azioni autorizzate dalle policy. La tua applicazione diventa la risorsa. Come risultato di questo flusso di lavoro, Verified Permissions crea un archivio di politiche, una funzione Lambda e un autorizzatore API Lambda. È necessario assegnare l'autorizzatore Lambda all'API dopo aver completato questo flusso di lavoro.

1. Nella console Autorizzazioni verificate, seleziona Crea nuovo archivio di politiche.
2. Nella sezione Opzioni di avvio, scegli Configura con API Gateway e una fonte di identità e seleziona Avanti.
3. Nella fase Importa risorse e azioni, in API, scegli un'API che funga da modello per le risorse e le azioni del tuo policy store.
    a. Scegli una fase di implementazione tra le fasi configurate nella tua API e seleziona Importa API. Per ulteriori informazioni sulle fasi dell'API, consulta Configurazione di una fase per un'API REST nella Amazon API Gateway Developer Guide.
    b. Visualizza un'anteprima della mappa delle risorse e delle azioni importate.
    c. Per aggiornare risorse o azioni, modifica i percorsi o i metodi delle API nella API Gateway console e seleziona Importa API per visualizzare gli aggiornamenti.
    d. Quando sei soddisfatto delle tue scelte, scegli Avanti.
4. In Identity source, scegli un tipo di provider di identità. Puoi scegliere un pool di Amazon Cognito utenti o un tipo di IdP OpenID Connect (OIDC).
5. Se hai scelto: Amazon Cognito
    a. Scegli un pool di utenti nello stesso Regione AWS archivio delle politiche. Account AWS

b. Scegli il tipo di token da passare all'API che desideri inviare per l'autorizzazione. Entrambi i tipi di token contengono gruppi di utenti, la base di questo modello di autorizzazione collegato all'API.

c. In App client validation, puoi limitare l'ambito di un policy store a un sottoinsieme dei client di Amazon Cognito app in un pool di utenti multi-tenant. Per richiedere l'autenticazione dell'utente con uno o più client di app specificati nel tuo pool di utenti, seleziona Accetta token solo con il client di app previsto. IDs Per accettare qualsiasi utente che si autentichi con il pool di utenti, seleziona Don't validate app client. IDs

d. Scegli Next (Successivo).

6. Se hai scelto un provider OIDC esterno:

a. In URL dell'emittente, inserisci l'URL dell'emittente OIDC. Questo è l'endpoint del servizio che fornisce, ad esempio, il server di autorizzazione, le chiavi di firma e altre informazioni sul provider. `https://auth.example.com` L'URL dell'emittente deve ospitare un documento di rilevamento OIDC presso. `/.well-known/openid-configuration`

b. In Tipo di token, scegli il tipo di OIDC JWT che desideri che la tua applicazione invii per l'autorizzazione. Per ulteriori informazioni, vedere Mappatura dei token allo schema e Mappatura Amazon Cognito dei token OIDC allo schema.

c. (opzionale) In Dichiarazioni token: facoltativo, scegli Aggiungi un'attestazione token, inserisci un nome per il token e seleziona un tipo di valore.

d. In Reclami relativi ai token utente e di gruppo, procedi come segue:

i. Inserisci il nome dell'attestazione utente nel token per l'origine dell'identità. Si tratta in genere `sub` di un'attestazione derivante dal tuo ID o token di accesso che contiene l'identificatore univoco dell'entità da valutare. Le identità dell'IdP OIDC connesso verranno mappate al tipo di utente nel tuo policy store.

ii. Inserisci il nome di un claim di gruppo nel token per l'origine dell'identità. Si tratta in genere `groups` di un'attestazione derivante dal tuo ID o token di accesso che contiene un elenco dei gruppi dell'utente. Il tuo archivio delle politiche autorizzerà le richieste in base all'appartenenza al gruppo.

e. In Audience validation, scegli `Add value` e aggiungi un valore che desideri che il tuo policy store accetti nelle richieste di autorizzazione.

f. Scegli Next (Successivo).

7. Se hai scelto Amazon Cognito, Verified Permissions interroga il tuo pool di utenti per individuare i gruppi. Per i provider OIDC, inserisci i nomi dei gruppi manualmente. Il

passaggio Assegna azioni ai gruppi crea politiche per l'archivio delle politiche che consentono ai membri del gruppo di eseguire azioni.

a. Scegli o aggiungi i gruppi che desideri includere nelle tue politiche.

b. Assegna azioni a ciascuno dei gruppi selezionati.

c. Scegli Next (Successivo).

8. Nell'integrazione con l'app Deploy, scegli se desideri collegare manualmente l'autorizzatore Lambda manualmente in un secondo momento o se desideri che Verified Permissions lo faccia subito e rivedi i passaggi che Verified Permissions eseguirà per creare il tuo policy store e l'autorizzatore Lambda.

9. Quando sei pronto per creare le nuove risorse, scegli Create policy store.

10. Tieni aperta la fase di stato del Policy store nel browser per monitorare l'avanzamento della creazione delle risorse tramite Autorizzazioni verificate.

11. Dopo qualche tempo, in genere circa un'ora, o quando la fase di autorizzazione Deploy Lambda mostra l'esito positivo, se hai scelto di collegare l'autorizzatore manualmente, configura l'autorizzatore.

    Verified Permissions avrà creato una funzione Lambda e un autorizzatore Lambda nella tua API. Scegli Open API per accedere alla tua API.

    Per informazioni su come assegnare un'autorizzazione Lambda, consulta API Gateway Use Lambda authorizers nella Amazon API Gateway Developer Guide.

    a. Accedi a Authorizers per la tua API e annota il nome dell'autorizzatore creato da Verified Permissions.

    b. Vai a Risorse e seleziona un metodo di primo livello nella tua API.

    c. Seleziona Modifica in Impostazioni di richiesta del metodo.

    d. Imposta l'Autorizzatore in modo che sia il nome dell'autorizzatore che hai annotato in precedenza.

    e. Espandi le intestazioni delle richieste HTTP, inserisci un nome o e seleziona **`AUTHORIZATION`** Obbligatorio.

    f. Implementa la fase API.

    g. Salva le modifiche.

12. Testa il tuo sistema di autorizzazione con un token del pool di utenti del tipo Token selezionato nel passaggio Scegli l'origine dell'identità. Per ulteriori informazioni sull'accesso

al pool di utenti e sul recupero dei token, consulta Flusso di autenticazione del pool di utenti nella Amazon Cognito Developer Guide.

13. Prova nuovamente l'autenticazione con un token del pool di utenti nell'`AUTHORIZATION`intestazione di una richiesta alla tua API.
14. Esamina il tuo nuovo archivio di politiche. Aggiungi e perfeziona le politiche.

Sample policy store

Per creare un policy store utilizzando il metodo di configurazione Sample policy store

1. Nella sezione Opzioni di avvio, scegli Sample policy store.
2. Nella sezione Progetto di esempio, scegli il tipo di applicazione di esempio per le autorizzazioni verificate da utilizzare.

   - PhotoFlashè un'applicazione web di esempio rivolta ai clienti che consente agli utenti di condividere foto e album individuali con gli amici. Gli utenti possono impostare autorizzazioni dettagliate su chi è autorizzato a visualizzare, commentare e condividere nuovamente le proprie foto. I proprietari di account possono anche creare gruppi di amici e organizzare le foto in album.
   - DigitalPetStoreè un'applicazione di esempio in cui chiunque può registrarsi e diventare cliente. I clienti possono aggiungere animali domestici in vendita, cercare animali domestici ed effettuare ordini. I clienti che hanno aggiunto un animale domestico vengono registrati come proprietari dell'animale. I proprietari di animali domestici possono aggiornare i dettagli dell'animale, caricare un'immagine dell'animale o eliminare l'elenco degli animali domestici. I clienti che hanno effettuato un ordine vengono registrati come proprietari dell'ordine. I proprietari degli ordini possono ottenere dettagli sull'ordine o annullarlo. I gestori dei negozi di animali hanno accesso amministrativo.

     > **Note**
     >
     > L'archivio DigitalPetStoredi policy di esempio non include modelli di policy. Gli archivi TinyTododi policy PhotoFlashe di esempio includono modelli di policy.

   - TinyTodoè un'applicazione di esempio che consente agli utenti di creare attività ed elenchi di attività. I proprietari degli elenchi possono gestire e condividere i propri elenchi e specificare chi può visualizzare o modificare i propri elenchi.

3. Uno spazio dei nomi per lo schema del tuo archivio di policy di esempio viene generato automaticamente in base al progetto di esempio scelto.
4. Scegli Crea archivio di politiche.

   Il tuo policy store viene creato con criteri e uno schema per il policy store di esempio che hai scelto. Per ulteriori informazioni sulle politiche collegate ai modelli che è possibile creare per gli archivi di policy di esempio, consulta. Esempio di politiche collegate a modelli di Amazon Verified Permissions

Empty policy store

Per creare un policy store utilizzando il metodo di configurazione Empty policy store

1. Nella sezione Opzioni di avvio, scegli Empty policy store.
2. Scegli Crea archivio di politiche.

Un policy store vuoto viene creato senza uno schema, il che significa che i criteri non vengono convalidati. Per ulteriori informazioni sull'aggiornamento dello schema per il policy store, vedereSchema di archiviazione delle politiche di Amazon Verified Permissions.

Per ulteriori informazioni sulla creazione di policy per il tuo policy store, consulta Creazione di politiche statiche di Amazon Verified Permissions eCreazione di politiche collegate ai modelli di Amazon Verified Permissions.

AWS CLI

Per creare un archivio delle politiche vuoto utilizzando AWS CLI.

È possibile creare un archivio delle politiche utilizzando l'`create-policy-store`operazione.

> **Note**
>
> Un archivio delle politiche creato utilizzando il AWS CLI è vuoto.
>
> - Per aggiungere uno schema, vedereSchema di archiviazione delle politiche di Amazon Verified Permissions.
> - Per aggiungere politiche, vedereCreazione di politiche statiche di Amazon Verified Permissions.

- Per aggiungere modelli di policy, consultaCreazione di modelli di policy per le autorizzazioni verificate da Amazon.

```
$ 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

È possibile creare un archivio di politiche utilizzando l'`CreatePolicyStore`API. Per ulteriori informazioni, consulta CreatePolicyStorela Amazon Verified Permissions API Reference Guide.

Implementazione delle autorizzazioni verificate da Amazon in Rust con l'SDK AWS

Questo argomento fornisce un esempio pratico di implementazione di Amazon Verified Permissions in Rust con l' AWS SDK. Questo esempio mostra come sviluppare un modello di autorizzazione in grado di verificare se un utente è in grado di visualizzare una foto. Il codice di esempio utilizza il aws-sdk-verifiedpermissionscrate di AWS SDK per Rust, che offre un robusto set di strumenti con cui interagire. Servizi AWS

Prerequisiti

Prima di iniziare, assicurati di avere la AWS CLI configurata sul tuo sistema e di avere familiarità con Rust.

- Per istruzioni sull'installazione di AWS CLI, consulta la guida all'installazione della AWS CLI.
- Per istruzioni sulla configurazione di AWS CLI, vedere Configurazione delle impostazioni per e Impostazioni dei file di configurazione AWS CLI e credenziali in. AWS CLI
- Per ulteriori informazioni su Rust, consulta rust-lang.org e la AWS SDK for Rust Developer Guide.

Una volta preparato l'ambiente, esploriamo come implementare le autorizzazioni verificate in Rust.

Prova il codice di esempio

Il codice di esempio esegue le seguenti operazioni:

- Configura il client SDK con cui comunicare AWS
- Crea un archivio di politiche
- Definisce la struttura del policy store aggiungendo uno schema
- Aggiunge una politica per controllare le richieste di autorizzazione
- Invia una richiesta di autorizzazione di prova per verificare che tutto sia impostato correttamente

Per verificare il codice di esempio

1. Crea un progetto Rust.
2. Sostituisci qualsiasi codice esistente `main.rs` con il seguente codice:

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

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

```
//Function that adds a schema to the policy store in the client
async fn put_schema(client: &Client, ps_id: &str, schema: &str) -> PutSchemaOutput
 {
    let schema =
 client.put_schema().definition(SchemaDefinition::CedarJson(schema.to_string())).policy_sto
    return schema.unwrap();
}

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

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

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

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

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

```
//Add schema to policy store
    let schema= r#"{
        "PhotoFlash": {
            "actions": {
                "ViewPhoto": {
                    "appliesTo": {
                        "context": {
                            "type": "Record",
                            "attributes": {}
                        },
                        "principalTypes": [
                            "User"
                        ],
                        "resourceTypes": [
                            "Photo"
                        ]
                    },
                    "memberOf": []
                }
            },
            "entityTypes": {
                "Photo": {
                    "memberOfTypes": [],
                    "shape": {
                        "type": "Record",
                        "attributes": {
                            "IsPrivate": {
                                "type": "Boolean"
                            }
                        }
                    }
                },
                "User": {
                    "memberOfTypes": [],
                    "shape": {
                        "attributes": {},
                        "type": "Record"
                    }
                }
            }
        }
    }"#;
    let put_schema = put_schema(&client, &policy_store.policy_store_id,
 schema).await;
```

```
    println!(
        "Created Schema with Namespace: {:?}",
        put_schema.namespaces
    );

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

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

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

```
}
```

3. Esegui il codice `cargo run` inserendolo nel terminale.

Se il codice viene eseguito correttamente, verrà visualizzato il terminale `Decision: Allow` seguito dall'ID della politica determinante. Ciò significa che hai creato con successo un archivio di politiche e lo hai testato utilizzando l' AWS SDK per Rust.

Eseguire la pulizia delle risorse

Dopo aver finito di esplorare il tuo archivio delle politiche, eliminalo.

Come eliminare un archivio di policy

È possibile eliminare un policy store utilizzando l'`delete-policy-store`operazione, sostituendola *`PSEXAMPLEabcdefg111111`* con l'ID del policy store che si desidera eliminare.

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

Se il comando ha esito positivo, non produce alcun output.

Archivi di policy collegati alle API

Un caso d'uso comune consiste nell'utilizzare Amazon Verified Permissions per autorizzare l'accesso degli utenti all' APIs hosting su Amazon API Gateway. Utilizzando una procedura guidata nella AWS console, puoi creare policy di accesso basate sui ruoli per gli utenti gestiti in Amazon Cognito o in qualsiasi provider di identità OIDC (IdP) e distribuire AWS Lambda un Authorizer che richiama Autorizzazioni verificate per valutare queste politiche.

Per completare la procedura guidata, scegli Configura con API Gateway e un provider di identità quando crei un nuovo archivio di politiche e segui i passaggi.

Viene creato un policy store collegato all'API che fornisce il modello di autorizzazione e le risorse per le richieste di autorizzazione. Il policy store ha un'origine di identità e un autorizzatore Lambda che si connette API Gateway alle autorizzazioni verificate. Una volta creato il policy store, è possibile autorizzare le richieste API in base all'appartenenza ai gruppi degli utenti. Ad esempio, le autorizzazioni verificate possono concedere l'accesso solo agli utenti che sono membri del gruppo. `Directors`

Man mano che l'applicazione cresce, è possibile implementare autorizzazioni granulari con attributi utente e ambiti OAuth 2.0 utilizzando il linguaggio di policy Cedar. Ad esempio, le autorizzazioni verificate possono concedere l'accesso solo agli utenti che dispongono di un attributo nel dominio. `email` `mycompany.co.uk`

Dopo aver impostato il modello di autorizzazione per la vostra API, la vostra responsabilità restante è quella di autenticare gli utenti e generare richieste API nell'applicazione, nonché di gestire l'archivio delle policy.

Per vedere una demo, consulta Amazon Verified Permissions - Quick Start Overview and Demo sul Amazon Web Services YouTube canale.

Argomenti

- In che modo Verified Permissions autorizza le richieste API
- Considerazioni per gli archivi di policy collegati alle API
- Aggiungere il controllo degli accessi basato sugli attributi (ABAC)
- Passare alla produzione con AWS CloudFormation
- Risoluzione dei problemi relativi agli archivi di policy collegati alle API

> **Important**
>
> Gli archivi di policy creati con l'opzione Configura con API Gateway e un'origine di identità nella console Verified Permissions non sono destinati alla distribuzione immediata in produzione. Con il tuo archivio di policy iniziale, finalizza il tuo modello di autorizzazione ed esporta le risorse del Policy Store in. CloudFormation Implementa le autorizzazioni verificate alla produzione in modo programmatico con. AWS Cloud Development Kit (AWS CDK) Per ulteriori informazioni, consulta Passare alla produzione con AWS CloudFormation.

In un archivio di policy collegato a un'API e a una fonte di identità, l'applicazione presenta un token del pool di utenti in un'intestazione di autorizzazione quando effettua una richiesta all'API. La fonte di identità del tuo policy store fornisce la convalida dei token per le autorizzazioni verificate. Il token forma le richieste `principal` di autorizzazione con l'IsAuthorizedWithTokenAPI. Verified Permissions crea politiche relative all'appartenenza ai gruppi degli utenti, come illustrato in una dichiarazione di gruppo in termini di identità (ID) e token di accesso, ad esempio `cognito:groups` per i pool di utenti. L'API elabora il token dell'applicazione in un programma di autorizzazione

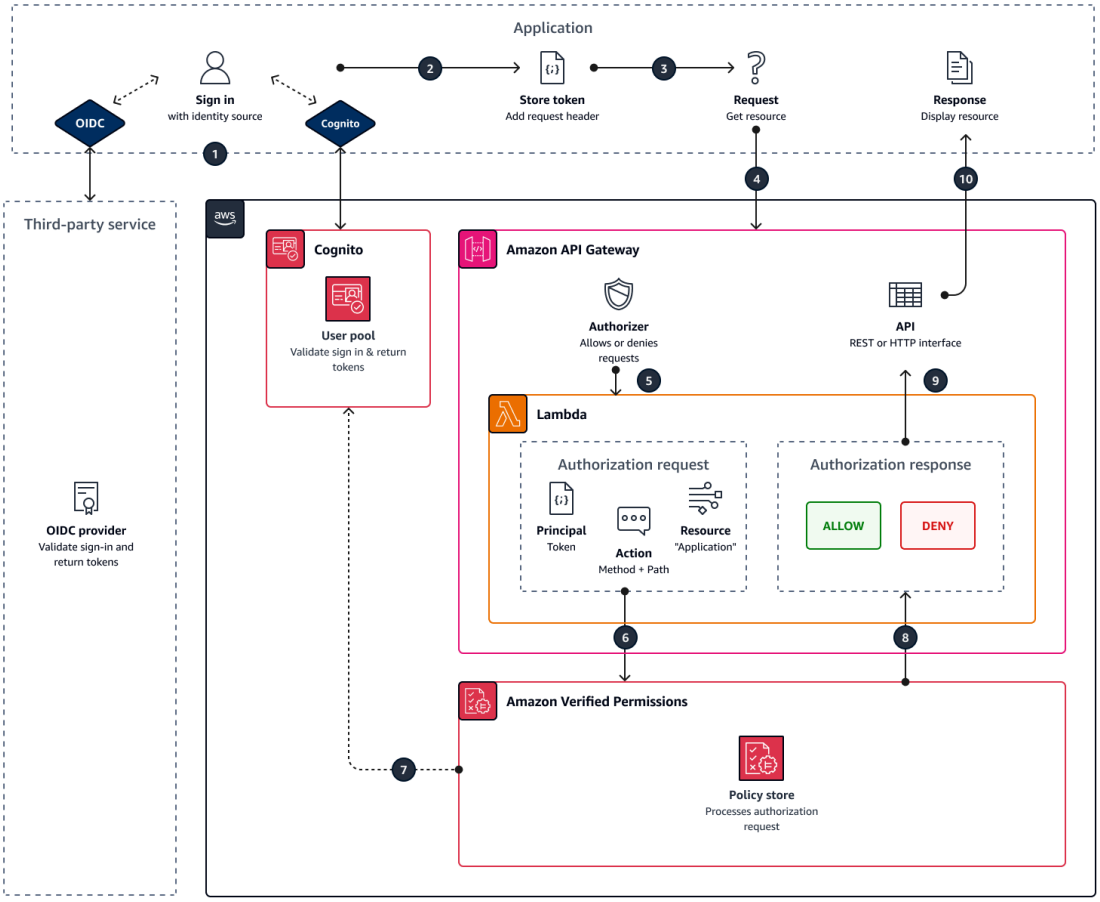
Lambda e lo invia a Verified Permissions per una decisione di autorizzazione. Quando l'API riceve la decisione di autorizzazione dall'autorizzatore Lambda, trasmette la richiesta all'origine dati o la nega.

Componenti dell'identità, dell'origine e dell' API Gateway autorizzazione con autorizzazioni verificate

- Un pool di Amazon Cognitoutenti o IdP OIDC che autentica e raggruppa gli utenti. I token degli utenti popolano l'appartenenza al gruppo e il principale o il contesto che Verified Permissions valuta nel tuo archivio delle politiche.
- API GatewayUn'API REST. Le autorizzazioni verificate definiscono le azioni dai percorsi API e dai metodi API, ad esempio`MyAPI::Action::get /photo`.
- Una funzione Lambda e un autorizzatore Lambda per la tua API. La funzione Lambda riceve i token portatori dal tuo pool di utenti, richiede l'autorizzazione da Verified Permissions e restituisce una decisione a. API Gateway Il flusso di lavoro Configura con API Gateway e un'origine di identità crea automaticamente questo autorizzatore Lambda per te.
- Un archivio di policy per le autorizzazioni verificate. L'origine dell'identità del Policy Store è il tuo pool di Amazon Cognito utenti o il gruppo di provider OIDC. Lo schema del policy store riflette la configurazione dell'API e le policy collegano i gruppi di utenti alle azioni API consentite.
- Un'applicazione che autentica gli utenti con il tuo IdP e aggiunge token alle richieste API.

In che modo Verified Permissions autorizza le richieste API

Quando si crea un nuovo policy store e si seleziona l'opzione Configura con API Gateway e un'origine di identità, Verified Permissions crea lo schema e le politiche del policy store. Lo schema e le politiche riflettono le azioni delle API e i gruppi di utenti che si desidera autorizzare a eseguire tali azioni. Verified Permissions crea anche la funzione e l'autorizzatore Lambda.



1. L'utente accede con l'applicazione tramite Amazon Cognito o un altro IdP OIDC. L'IdP emette ID e token di accesso con le informazioni dell'utente.
2. L'applicazione memorizza il. JWTs Per ulteriori informazioni, consulta Usare i token con i pool di utenti nella Guida per gli Amazon Cognito sviluppatori.
3. L'utente richiede i dati che l'applicazione deve recuperare da un'API esterna.
4. L'applicazione richiede dati da un'API REST in API Gateway. Aggiunge un ID o un token di accesso come intestazione della richiesta.
5. Se l'API dispone di una cache per la decisione di autorizzazione, restituisce la risposta precedente. Se la memorizzazione nella cache è disabilitata o l'API non ha una cache corrente, API Gateway passa i parametri della richiesta a un autorizzatore Lambda basato su token.
6. La funzione Lambda invia una richiesta di autorizzazione a un archivio di policy di autorizzazioni verificate con l'API. IsAuthorizedWithToken La funzione Lambda trasmette gli elementi di una decisione di autorizzazione:

   a. Il token dell'utente come principale.

b. Il metodo API combinato con il percorso dell'API, ad esempio`GetPhoto`, come azione.

c. Il termine `Application` come risorsa.

7. Verified Permissions convalida il token. Per ulteriori informazioni su come vengono convalidati Amazon Cognito i token, consulta Authorization with Amazon Verified Permissions nella Developer Guide.Amazon Cognito
8. Verified Permissions valuta la richiesta di autorizzazione rispetto alle politiche del tuo archivio di polizze e restituisce una decisione di autorizzazione.
9. L'autorizzatore Lambda restituisce una risposta `Allow` or `Deny` a. API Gateway
10. L'API restituisce dati o una `ACCESS_DENIED` risposta all'applicazione. L'applicazione elabora e visualizza i risultati della richiesta API.

Considerazioni per gli archivi di policy collegati alle API

Quando crei un policy store collegato alle API nella console Verified Permissions, stai creando un test per un'eventuale implementazione di produzione. Prima di passare alla produzione, stabilisci una configurazione fissa per l'API e il pool di utenti. Considera i fattori seguenti:

API Gateway memorizza nella cache le risposte

Negli archivi di policy collegati alle API, Verified Permissions crea un autorizzatore Lambda con un TTL di autorizzazione che memorizza nella cache di 120 secondi. Puoi modificare questo valore o disattivare la memorizzazione nella cache nel tuo programma di autorizzazione. In un autorizzatore con memorizzazione nella cache abilitata, l'autorizzatore restituisce la stessa risposta ogni volta fino alla scadenza del TTL. Ciò può prolungare la durata effettiva dei token del pool di utenti di una durata pari al TTL di memorizzazione nella cache della fase richiesta.

Amazon Cognito i gruppi possono essere riutilizzati

Amazon Verified Permissions determina l'appartenenza al gruppo per gli utenti del pool di utenti in base alla `cognito:groups` dichiarazione contenuta nell'ID o nel token di accesso di un utente. Il valore di questa dichiarazione è una matrice dei nomi descrittivi dei gruppi di pool di utenti a cui l'utente appartiene. Non è possibile associare i gruppi di pool di utenti a un identificatore univoco.

I gruppi di pool di utenti eliminati e ricreati con lo stesso nome presenti nel policy store come stesso gruppo. Quando elimini un gruppo da un pool di utenti, elimina tutti i riferimenti al gruppo dal tuo policy store.

Lo spazio dei nomi e lo schema derivati dall'API sono point-in-time

Verified Permissions acquisisce la tua API in un determinato momento: interroga la tua API solo quando crei il tuo archivio delle politiche. Quando lo schema o il nome dell'API cambia, devi aggiornare il policy store e l'autorizzatore Lambda o creare un nuovo policy store collegato all'API. Verified Permissions ricava lo spazio dei nomi del policy store dal nome dell'API.

La funzione Lambda non ha una configurazione VPC

La funzione Lambda creata da Verified Permissions per il tuo autorizzatore API viene lanciata nel VPC predefinito. Per impostazione predefinita. APIs con accesso alla rete limitato ai soli utenti privati non VPCs possono comunicare con la funzione Lambda che autorizza le richieste di accesso con autorizzazioni verificate.

Verified Permissions distribuisce le risorse di autorizzazione in CloudFormation

Per creare un policy store collegato all'API, è necessario accedere a un utente con privilegi elevati alla console Verified Permissions AWS . Questo utente distribuisce uno stack che crea risorse su diverse piattaforme CloudFormation . Servizi AWS Questo principale deve avere l'autorizzazione per aggiungere e modificare risorse in Autorizzazioni verificate IAM, Lambda e. API GatewayÈ consigliabile non condividere queste credenziali con altri amministratori dell'organizzazione.

Passare alla produzione con AWS CloudFormationPer una panoramica delle risorse create da Verified Permissions, vedi.

Aggiungere il controllo degli accessi basato sugli attributi (ABAC)

Una tipica sessione di autenticazione con un IdP restituisce ID e token di accesso. Puoi passare uno di questi tipi di token come token portante nelle richieste dell'applicazione alla tua API. A seconda delle scelte effettuate al momento della creazione del policy store, Verified Permissions prevede uno dei due tipi di token. Entrambi i tipi contengono informazioni sull'appartenenza al gruppo dell'utente. Per ulteriori informazioni sui tipi di token in Amazon Cognito, consulta Using tokens with user pool nella Amazon Cognito Developer Guide.

Dopo aver creato un archivio delle politiche, è possibile aggiungere ed estendere le politiche. Ad esempio, puoi aggiungere nuovi gruppi alle tue politiche man mano che le aggiungi al tuo pool di utenti. Poiché l'archivio delle politiche è già a conoscenza del modo in cui il pool di utenti presenta i gruppi in token, è possibile consentire una serie di azioni per ogni nuovo gruppo con una nuova politica.

Potresti anche voler estendere il modello di valutazione delle politiche basato sui gruppi in un modello più preciso basato sulle proprietà degli utenti. I token del pool di utenti contengono informazioni aggiuntive sugli utenti che possono contribuire alle decisioni di autorizzazione.

Token ID

I token ID rappresentano gli attributi di un utente e hanno un alto livello di controllo degli accessi preciso. Per valutare indirizzi e-mail, numeri di telefono o attributi personalizzati come reparto e responsabile, valuta il token ID.

Token di accesso

I token di accesso rappresentano le autorizzazioni di un utente con OAuth ambiti 2.0. Per aggiungere un livello di autorizzazione o impostare richieste di risorse aggiuntive, valuta il token di accesso. Ad esempio, puoi verificare che un utente appartenga ai gruppi appropriati e disponga di un ambito come `PetStore.read` quello che generalmente autorizza l'accesso all'API. I pool di utenti possono aggiungere ambiti personalizzati ai token con server di risorse e con personalizzazione dei token in fase di esecuzione.

Vedi Mappatura dei Amazon Cognito token allo schema e Mappatura dei token OIDC allo schema, ad esempio politiche che elaborano le dichiarazioni in ID e token di accesso.

Passare alla produzione con AWS CloudFormation

Gli archivi di policy collegati alle API sono un modo per creare rapidamente un modello di autorizzazione per un'API. API Gateway Sono progettati per fungere da ambiente di test per il componente di autorizzazione dell'applicazione. Dopo aver creato l'archivio delle politiche di test, dedica del tempo a perfezionare le politiche, lo schema e l'autorizzazione Lambda.

Potresti modificare l'architettura della tua API, richiedendo modifiche equivalenti allo schema e alle politiche del tuo Policy Store. Gli archivi di policy collegati alle API non aggiornano automaticamente il loro schema dall'architettura API: Verified Permissions interroga l'API solo al momento della creazione di un archivio di politiche. Se l'API cambia sufficientemente, potrebbe essere necessario ripetere la procedura con un nuovo policy store.

Quando l'applicazione e il modello di autorizzazione sono pronti per l'implementazione in produzione, integra il policy store collegato all'API che hai sviluppato con i tuoi processi di automazione. Come procedura consigliata, si consiglia di esportare lo schema e le policy del Policy Store in un AWS CloudFormation modello da distribuire su altri sistemi e. Account AWS Regioni AWS

I risultati del processo di archiviazione delle politiche collegato all'API sono un policy store iniziale e un autorizzatore Lambda. L'autorizzatore Lambda dispone di diverse risorse dipendenti. Verified Permissions distribuisce queste risorse in uno stack generato automaticamente. CloudFormation Per la distribuzione in produzione, è necessario raccogliere le risorse del Policy Store e dell'Autorizzatore Lambda in un modello. Un policy store collegato all'API è composto dalle seguenti risorse:

1. AWS::VerifiedPermissions::PolicyStore: Copia lo schema nell'`SchemaDefinitio`noggetto. Esci " dai personaggi come\".
2. AWS::VerifiedPermissions::IdentitySource: Copia i valori dall'output del GetIdentitySource tuo test policy store e modificali secondo necessità.
3. Uno o più dei AWS::VerifiedPermissions::Policyseguenti: copia la dichiarazione di policy nell'`Definitio`noggetto. Fuggi " dai personaggi come\".
4. AWS: :Lambda: :Funzione, AWS::: :Ruolo,IAM::: :Politica,AWS:IAM: :Autorizzatore,AWS ApiGateway AWS::Lambda::Permission

Il modello seguente è un esempio di policy store. Puoi aggiungere le risorse dell'autorizzazione Lambda dallo stack esistente a questo modello.

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

Risoluzione dei problemi relativi agli archivi di policy collegati alle API

Usa le informazioni qui per aiutarti a diagnosticare e risolvere i problemi più comuni quando crei archivi di policy collegati all'API di Amazon Verified Permissions.

Argomenti

- Ho aggiornato la mia politica ma la decisione di autorizzazione non è cambiata
- Ho collegato l'autorizzatore Lambda alla mia API ma non genera richieste di autorizzazione
- Ho ricevuto una decisione di autorizzazione inaspettata e desidero rivedere la logica di autorizzazione
- Voglio trovare i log del mio autorizzatore Lambda
- Il mio autorizzatore Lambda non esiste
- La mia API si trova in un VPC privato e non può richiamare l'autorizzatore
- Voglio elaborare attributi utente aggiuntivi nel mio modello di autorizzazione
- Voglio aggiungere nuove azioni, attributi del contesto dell'azione o attributi delle risorse

Ho aggiornato la mia politica ma la decisione di autorizzazione non è cambiata

Per impostazione predefinita, Verified Permissions configura l'autorizzatore Lambda per memorizzare nella cache le decisioni di autorizzazione per 120 secondi. Riprova dopo due minuti o disattiva la cache sull'autorizzatore. Per ulteriori informazioni, consulta Enabling API caching per migliorare la reattività nella Amazon API Gateway Developer Guide.

Ho collegato l'autorizzatore Lambda alla mia API ma non genera richieste di autorizzazione

Per iniziare a elaborare le richieste, devi implementare la fase API a cui hai collegato l'autorizzatore. Per ulteriori informazioni, consulta Deploying a REST API nella Amazon API Gateway Developer Guide.

Ho ricevuto una decisione di autorizzazione inaspettata e desidero rivedere la logica di autorizzazione

Il processo di archiviazione delle politiche collegato all'API crea una funzione Lambda per l'autorizzatore. Verified Permissions integra automaticamente la logica delle decisioni di autorizzazione nella funzione di autorizzazione. È possibile tornare indietro dopo aver creato l'archivio delle politiche per rivedere e aggiornare la logica della funzione.

Per individuare la funzione Lambda dalla AWS CloudFormation console, scegli il pulsante Verifica distribuzione nella pagina Panoramica del tuo nuovo archivio di politiche.

Puoi anche localizzare la tua funzione nella AWS Lambda console. Accedi alla console nel tuo archivio Regione AWS delle politiche e cerca il nome di una funzione con il prefisso di`AVPAuthorizerLambda`. Se avete creato più di un policy store collegato all'API, utilizzate l'ora dell'ultima modifica delle funzioni per correlarle alla creazione del policy store.

Voglio trovare i log del mio autorizzatore Lambda

Le funzioni Lambda raccolgono metriche e registrano i risultati delle chiamate in Amazon. CloudWatch Per esaminare i log, individua la funzione nella console Lambda e scegli la scheda Monitor. Seleziona Visualizza CloudWatch registri e rivedi le voci nel gruppo di log.

Per ulteriori informazioni sui log delle funzioni Lambda, consulta Using Amazon CloudWatch Logs with AWS Lambda nella Developer Guide.AWS Lambda

Il mio autorizzatore Lambda non esiste

Dopo aver completato la configurazione di un policy store collegato all'API, devi collegare l'autorizzatore Lambda all'API. Se non riesci a localizzare l'autorizzatore nella API Gateway console, è possibile che le risorse aggiuntive per il policy store abbiano avuto esito negativo o non siano ancora state distribuite. Gli archivi di policy collegati alle API distribuiscono queste risorse in uno stack. CloudFormation

Autorizzazioni verificate visualizza un link con l'etichetta Verifica distribuzione al termine del processo di creazione. Se hai già abbandonato questa schermata, vai alla CloudFormation console e cerca negli stack recenti un nome con il prefisso. `AVPAuthorizer-<policy store ID>` CloudFormation fornisce preziose informazioni sulla risoluzione dei problemi nell'output di una distribuzione di stack.

Per informazioni sulla risoluzione dei problemi relativi agli CloudFormation stack, consulta Troubleshooting CloudFormation nella Guida per l'AWS CloudFormation utente.

La mia API si trova in un VPC privato e non può richiamare l'autorizzatore

Verified Permissions non supporta l'accesso agli autorizzatori Lambda tramite endpoint VPC. È necessario aprire un percorso di rete tra l'API e la funzione Lambda che funge da autorizzatore.

Voglio elaborare attributi utente aggiuntivi nel mio modello di autorizzazione

Il processo di archiviazione delle politiche collegato all'API ricava le politiche di autorizzazione verificate dalle dichiarazioni dei gruppi nei token degli utenti. Per aggiornare il tuo modello di autorizzazione in modo da prendere in considerazione attributi utente aggiuntivi, integra tali attributi nelle tue politiche.

Puoi mappare molte attestazioni in ID e token di accesso dai pool di utenti di Amazon Cognito alle dichiarazioni politiche sulle autorizzazioni verificate. Ad esempio, la maggior parte degli utenti ha un `email` claim nel token ID. Per ulteriori informazioni sull'aggiunta di attestazioni dalla fonte di identità alle policy, consulta Mappatura dei Amazon Cognito token allo schema e Mappatura dei token OIDC allo schema.

Voglio aggiungere nuove azioni, attributi del contesto dell'azione o attributi delle risorse

Un policy store collegato all'API e l'autorizzatore Lambda che crea sono una risorsa. point-in-time Riflettono lo stato dell'API al momento della creazione. Lo schema del policy store non assegna alcun attributo di contesto alle azioni, né alcun attributo o genitore alla `Application` risorsa predefinita.

Quando aggiungi azioni, percorsi e metodi, alla tua API, devi aggiornare il policy store per essere a conoscenza delle nuove azioni. È inoltre necessario aggiornare l'autorizzatore Lambda per elaborare le richieste di autorizzazione per le nuove azioni. Puoi ricominciare da capo con un nuovo policy store o aggiornare il policy store esistente.

Per aggiornare il tuo archivio delle politiche esistente, individua la tua funzione. Esamina la logica nella funzione generata automaticamente e aggiornala per elaborare le nuove azioni, attributi o contesto. Quindi modifica lo schema per includere le nuove azioni e attributi.

Eliminazione degli archivi delle politiche

Puoi eliminare gli archivi di policy di Amazon Verified Permissions utilizzando Console di gestione AWS o il AWS CLI. L'eliminazione di un Policy Store elimina definitivamente lo schema e tutte le policy e i modelli di policy presenti nel Policy Store. Verranno eliminati anche tutti gli alias del policy store associato al policy store eliminato.

La protezione dall'eliminazione impedisce l'eliminazione accidentale di un policy store. La protezione dall'eliminazione è abilitata su tutti i nuovi policy store creati tramite Console di gestione AWS. Al contrario, è disabilitata per tutti gli archivi di policy creati tramite una chiamata API o SDK.

Potresti voler eliminare gli archivi delle politiche per i seguenti motivi:

- È stata raggiunta la quota di archivi delle politiche disponibili in una determinata regione. Per ulteriori informazioni, consulta Quote per le risorse.
- Non supportate più un tenant in un'applicazione multi-tenant e, pertanto, non avete più bisogno di quell'archivio di policy.

Console di gestione AWS

Come eliminare un archivio di policy

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione sinistro, seleziona Settings (Impostazioni).
3. Scegli Elimina questo archivio di polizze.
4. Digita `delete` nella casella di testo e scegli Elimina.

> **Note**
>
> Se la protezione da eliminazione è abilitata, dovrai disabilitarla prima di poter scegliere Elimina. Per disabilitarla, seleziona Disabilita la protezione da eliminazione.

AWS CLI

Come eliminare un archivio di policy

È possibile eliminare un policy store utilizzando l'`delete-policy-store`operazione, sostituendola *PSEXAMPLEabcdefg111111* con l'ID del policy store che si desidera eliminare.

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

Se il comando ha esito positivo, non produce alcun output.

> **Note**
>
> Se la protezione da eliminazione è abilitata per questo policy store, è necessario innanzitutto eseguire l'`update-policy-store`operazione e disabilitare la protezione da eliminazione.
>
> ```
> aws verifiedpermissions update-policy-store \
>     --deletion-protection "DISABLED" \
>     --policy-store-id PSEXAMPLEabcdefg111111
> ```

Alias dell'archivio delle policy di Amazon Verified Permissions

Un alias di policy store è un nome descrittivo per un policy store. Ad esempio, gli alias di policy store consentono di fare riferimento a un policy store utilizzando `policy-store-alias/example-policy-store` invece di. `PSEXAMPLEabcdefg111111` Gli alias del Policy Store possono essere utilizzati in qualsiasi operazione di Verified Permissions che accetta un `policyStoreId` parametro di input.

È possibile creare un alias di Policy Store per un Policy Store utilizzando l'`CreatePolicyStoreAlias`API o utilizzando la risorsa. `AWS::VerifiedPermissions::PolicyStoreAlias` CloudFormation

L'API Amazon Verified Permissions fornisce il controllo completo degli alias del Policy Store in ogni regione Account AWS . L'API include operazioni per creare un alias del Policy Store (`CreatePolicyStoreAlias`), visualizzare i nomi degli alias del Policy Store e gli alias del Policy Store ARNs (`GetPolicyStoreAlias`,`ListPolicyStoreAliases`) ed eliminare un alias del Policy Store (). `DeletePolicyStoreAlias`

Argomenti

Proprietà degli alias del policy store

Come funzionano gli alias del Policy Store in Amazon Verified Permissions.

Un alias del Policy Store è una risorsa indipendente AWS

Un alias del policy store non è una proprietà di un policy store. Le azioni eseguite sull'alias del policy store non influiscono sul policy store associato. È possibile eliminare l'alias del policy store senza

alcun effetto sul policy store associato. Se si elimina un policy store, vengono eliminati anche tutti gli alias del policy store associati a quel policy store.

Ogni alias del policy store ha un Amazon Resource Name (ARN) che identifica in modo univoco l'alias del policy store. Se specifichi un alias del policy store come risorsa in una policy IAM, la policy fa riferimento all'alias del policy store, non al policy store associato.

Ogni alias del policy store ha due formati

Quando si crea un alias del policy store, si specifica il nome dell'alias del policy store. Amazon Verified Permissions crea per te l'alias ARN del Policy Store.

- L'alias ARN di un policy store è un Amazon Resource Name (ARN) che identifica in modo univoco l'alias del policy store.

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

- Un nome alias del policy store che è unico nella regione and. Account AWS Nell'API Amazon Verified Permissions, il nome alias del policy store è sempre preceduto da. `policy-store-alias/`

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

Gli alias del Policy Store non sono segreti

Gli alias dell'archivio delle politiche possono essere visualizzati in testo semplice nei CloudTrail log e in altri output. Non includere informazioni riservate o sensibili nel nome alias del policy store.

Ogni alias del policy store è associato a un policy store alla volta

L'alias del policy store e il relativo policy store associato devono appartenere allo stesso criterio Account AWS e alla stessa regione. È possibile associare un alias del policy store a qualsiasi policy store della stessa regione Account AWS .

Ad esempio, questo `ListPolicyStoreAliases` output mostra che l'alias del `example-policy-store` policy store è associato esattamente a un policy store di destinazione, rappresentato dalla `policyStoreId` proprietà.

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

È possibile associare più alias allo stesso archivio delle politiche

Ad esempio, è possibile associare gli `example-policy-store-2` alias `example-policy-store` and allo stesso policy store.

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

Un alias di policy store deve essere univoco in una regione and Account AWS

Ad esempio, è possibile avere un solo alias del Policy Store con lo stesso nome `example-policy-store` in ciascuna regione Account AWS . Gli alias del Policy Store fanno distinzione tra maiuscole e minuscole. Non è possibile modificare il nome dell'alias di un Policy Store. Tuttavia, è possibile eliminare l'alias del policy store e creare un nuovo alias del policy store con il nome desiderato dopo la scadenza del periodo di prenotazione di 24 ore.

È possibile creare alias di policy store con lo stesso nome in diverse regioni. Ogni alias dell'archivio delle politiche avrà un ARN univoco. Se il codice fa riferimento a un alias del tipo Policy Store`policy-store-alias/example-policy-store`, puoi eseguirlo in più regioni. In ogni regione, utilizza un archivio di policy diverso.

Gli alias dell'archivio delle politiche vengono eliminati automaticamente

Quando un alias dell'archivio delle politiche viene eliminato, il nome dell'alias dell'archivio delle politiche viene riservato per un periodo di 24 ore. Se si tenta di creare un alias del policy store con lo stesso nome durante questo periodo, la richiesta verrà rifiutata. Durante questo periodo, `GetPolicyStoreAlias` restituisce l'alias del policy store con lo `PendingDeletion` stato.

È possibile utilizzare gli alias per identificare gli archivi delle politiche

È possibile utilizzare un alias del policy store per identificare un policy store in tutte le operazioni che accettano un `policyStoreId` (ad esempio,`IsAuthorized`). In questi casi, il nome dell'alias del policy store deve essere preceduto da. `policy-store-alias/` Gli alias del Policy Store non possono essere utilizzati per identificare un Policy Store per l'operazione. `DeletePolicyStore`

Non è possibile utilizzare un nome alias di un policy store o un alias di policy store ARN per identificare un policy store nell'`Resource`elemento di una policy IAM. Per controllare l'accesso a un policy store quando vi si fa riferimento tramite un alias di policy store, vedere. Controllo dell'accesso agli alias del policy store

Creazione di alias di Amazon Verified Permissions Policy Store

È possibile creare un alias del policy store per fare riferimento a un policy store utilizzando un nome descrittivo. Il nome di un alias del policy store deve essere univoco per regione Account AWS . Gli alias del Policy Store possono essere associati solo agli archivi delle policy store di proprietà dello stesso Account AWS e attivi nella stessa regione dell'alias del Policy Store. Gli alias del Policy Store sono risorse separate con autorizzazione propria ARNs e IAM.

Per impostazione predefinita, è possibile associare solo 10 alias del Policy Store allo stesso Policy Store.

> **Note**
>
> `CreatePolicyStoreAlias`è idempotente. Se si chiama l'`CreatePolicyStoreAlias`operazione con un nome alias di policy store

e un ID di policy store che corrispondono a un alias di policy store esistente, l'`CreatePolicyStoreAlias`operazione ha esito positivo e restituisce l'alias esistente del policy store. Tuttavia, se si chiama l'`CreatePolicyStoreAlias`operazione con un nome alias del policy store esistente ma un ID di policy store diverso, l'operazione restituisce un. `ConflictException`

AWS CLI

Per creare un alias del policy store

È possibile creare un alias del policy store utilizzando l'CreatePolicyStoreAliasoperazione. L'esempio seguente crea un alias del policy store con il nome. `example-policy-store`

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

Recupero degli alias dello store delle policy di Amazon Verified Permissions

È possibile recuperare informazioni sugli alias del Policy Store utilizzando l'`GetPolicyStoreAlias`operazione per ottenere dettagli su uno specifico alias del Policy Store oppure l'`ListPolicyStoreAliases`operazione per elencare tutti gli alias del Policy Store nella propria area geografica. Account AWS

Ottenere un alias del policy store

Utilizza l'`GetPolicyStoreAlias`operazione per recuperare i dettagli su uno specifico alias del policy store, incluso l'ID associato del policy store.

AWS CLI

Per recuperare i dettagli su un alias del policy store

È possibile recuperare un alias del policy store utilizzando l'operazione. GetPolicyStoreAlias L'esempio seguente recupera i dettagli per un alias del policy store con il nome. `example-policy-store`

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

Elenco degli alias del Policy Store

Usa l'`ListPolicyStoreAliases`operazione per elencare tutti gli alias del Policy Store nella tua regione Account AWS . È possibile utilizzare il `filter` parametro per elencare solo gli alias del policy store associati a uno specifico policy store.

AWS CLI

Per elencare tutti gli alias del policy store

È possibile elencare gli alias dell'archivio delle politiche utilizzando l'ListPolicyStoreAliasesoperazione. L'esempio seguente elenca tutti gli alias del policy store di proprietà del 123456789012 Account AWS nella regione us-west-2.

```
$ aws verifiedpermissions list-policy-store-aliases
{
    "policyStoreAliases": [
        {
            "aliasName": "policy-store-alias/example-policy-store",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-
store-alias/example-policy-store",
```

```
            "createdAt": "2024-01-15T12:30:00.000000+00:00",
            "state": "Active"
        },
        {
            "aliasName": "policy-store-alias/example-policy-store-2",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-
store-alias/example-policy-store-2",
            "createdAt": "2024-01-16T09:15:00.000000+00:00",
            "state": "Active"
        },
        {
            "aliasName": "policy-store-alias/example-policy-store-3",
            "policyStoreId": "PSEXAMPLEabcdefg222222",
            "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-
store-alias/example-policy-store-3",
            "createdAt": "2024-01-17T14:45:00.000000+00:00",
            "state": "Active"
        }
    ]
}
```

Per elencare gli alias dell'archivio delle politiche per uno specifico archivio delle politiche

Utilizzate il `filter` parametro per elencare solo gli alias associati a uno specifico policy store.

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

```
            "state": "Active"
        }
    ]
}
```

Eliminazione degli alias dello store delle policy di Amazon Verified Permissions

È possibile eliminare un alias del policy store quando non è più necessario. L'eliminazione di un alias del policy store non influisce sul policy store associato. L'eliminazione di un policy store elimina tutti gli alias del policy store associati a tale policy store.

Dopo aver eliminato un alias del policy store, il nome alias del policy store viene riservato per 24 ore e non può essere riutilizzato durante questo periodo.

AWS CLI

Per eliminare un alias del policy store

È possibile eliminare un alias del policy store utilizzando l'DeletePolicyStoreAliasoperazione. L'esempio seguente elimina un alias del policy store con il nome. `example-policy-store`

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

Utilizzando le policy di Amazon Verified Permissions, archivia gli alias nelle operazioni API

Qualsiasi operazione Amazon Verified Permissions che accetta un `policyStoreId` parametro, ad esempio IsAuthorized, e IsAuthorizedWithTokenGetPolicyStore, può accettare un alias del Policy Store al posto dell'ID del Policy Store.

> **Important**
>
> Quando utilizzi un alias del policy store come valore di un `policyStoreId` parametro, devi includere il prefisso. `policy-store-alias/` Ad esempio, usa`policy-store-alias/example-policy-store`, not. `example-policy-store`

Utilizzo degli alias del Policy Store in Operations

Il `IsAuthorized` comando seguente utilizza un alias del policy store con il nome `example-policy-store` per identificare un policy store.

AWS CLI

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

> **Note**
>
> Non è possibile utilizzare un alias del policy store al posto del `policyStoreId` campo per l'DeletePolicyStoreoperazione.

Utilizzo degli alias del Policy store Across Regioni AWS

Uno degli usi più efficaci degli alias è nelle applicazioni eseguite in più Regioni AWS. Ad esempio, potresti avere un'applicazione globale che utilizza archivi di policy diversi in ogni regione.

- In us-east-1, vuoi usare. `PSEXAMPLEabcdefg111111`
- In eu-west-1, si desidera utilizzare. `PSEXAMPLEabcdefg222222`

È possibile creare una versione diversa dell'applicazione in ciascuna regione o utilizzare un dizionario o un'istruzione switch per selezionare l'archivio di politiche giusto per ciascuna regione. Tuttavia,

è molto più semplice creare un alias di policy store con lo stesso nome alias di policy store in ogni regione. Ricorda che il nome alias del policy store fa distinzione tra maiuscole e minuscole.

AWS CLI

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

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

Quindi, usa l'alias del policy store nel tuo codice. Quando il codice viene eseguito in ogni regione, l'alias del policy store farà riferimento al policy store associato in quella regione.

AWS CLI

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

Tuttavia, esiste il rischio che l'alias del policy store venga eliminato. In tal caso, i tentativi dell'applicazione di utilizzare il nome alias del policy store falliranno e potrebbe essere necessario ricreare o aggiornare l'alias del policy store. Per mitigare questo rischio, fate attenzione a concedere ai responsabili l'autorizzazione a gestire gli alias del policy store utilizzati nell'applicazione.

Controllo dell'accesso agli alias del policy store

I responsabili che gestiscono gli alias del Policy Store devono disporre dell'autorizzazione a interagire con tali alias del Policy Store e, per alcune operazioni, con l'alias del Policy Store a cui è associato l'alias del Policy Store. È possibile fornire queste autorizzazioni utilizzando le politiche. IAM

Le sezioni seguenti descrivono le autorizzazioni necessarie per creare e gestire gli alias del Policy Store.

autorizzazioni verificate: CreatePolicyStoreAlias

Per creare un alias dell'archivio delle politiche, il principale necessita delle seguenti autorizzazioni sia per l'alias del policy store che per l'archivio delle politiche associato.

- `verifiedpermissions:CreatePolicyStoreAlias`per l'alias del policy store. Fornisci questa autorizzazione in una IAM policy allegata al principale autorizzato a creare l'alias del policy store.

  L'esempio seguente di dichiarazione politica specifica un particolare alias del policy store in un elemento. `Resource` È tuttavia possibile elencare più alias di policy store ARNs o specificare un pattern di alias di policy store, ad esempio. `"sample*"` È inoltre possibile specificare un `Resource` valore di `"*"` per consentire al principale di creare qualsiasi alias del policy store nella Account AWS regione and.

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

- `verifiedpermissions:CreatePolicyStoreAlias`per l'archivio delle politiche associato. Questa autorizzazione deve essere fornita in una IAM politica.

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

autorizzazioni verificate: GetPolicyStoreAlias

Per ottenere dettagli su uno specifico alias del policy store, il responsabile deve disporre dell'`verifiedpermissions:GetPolicyStoreAlias`autorizzazione per l'alias dell'archivio delle politiche in un criterio. IAM

L'istruzione politica di esempio seguente fornisce l'autorizzazione principale per ottenere un alias specifico del policy store.

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

autorizzazioni verificate: ListPolicyStoreAliases

Per elencare gli alias del policy store nella regione Account AWS and, il principale deve disporre dell'autorizzazione per una policy. `verifiedpermissions:ListPolicyStoreAliases` IAM Poiché questo criterio non è correlato a nessuna particolare risorsa Policy Store o Policy Store alias, il valore dell'elemento resource nella policy deve essere. `"*"`

Ad esempio, la seguente dichiarazione di IAM policy fornisce l'autorizzazione principale per elencare tutti gli alias del policy store in. Account AWS

```
{
  "Sid": "IAMPolicyForListingAliases",
  "Effect": "Allow",
  "Action": "verifiedpermissions:ListPolicyStoreAliases",
  "Resource": "*"
}
```

autorizzazioni verificate: DeletePolicyStoreAlias

Per eliminare un alias del policy store, il principale necessita dell'autorizzazione solo per l'alias del policy store.

> **Note**
>
> L'eliminazione di un alias del policy store non ha alcun effetto sul policy store associato, sebbene le applicazioni che fanno riferimento all'alias del policy store riceveranno errori. Se si elimina erroneamente un alias del policy store, è possibile ricrearlo dopo il periodo di prenotazione di 24 ore.

Il principale necessita dell'`verifiedpermissions:DeletePolicyStoreAlias`autorizzazione per l'alias del policy store. Fornisci questa autorizzazione in una IAM policy allegata al principale che è autorizzato a eliminare l'alias del policy store.

L'esempio seguente di dichiarazione politica specifica l'alias del policy store in un elemento. `Resource` È tuttavia possibile elencare più alias del policy store ARNs o specificare un modello di alias del policy store, ad esempio. `"sample*"` È inoltre possibile specificare un `Resource` valore di `"*"` per consentire al principale di eliminare qualsiasi alias del policy store nella Account AWS regione and.

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

Limitazione delle autorizzazioni degli alias del Policy Store

È possibile utilizzare un alias del policy store per fare riferimento a un policy store in qualsiasi operazione che accetta un `policyStoreId` campo come input. Quando lo fai, Amazon Verified Permissions autorizza l'`verifiedpermissions:GetPolicyStoreAlias`alias del Policy Store e l'operazione richiesta sul Policy Store associato.

Ad esempio, se l'`IsAuthorized`operazione viene eseguita utilizzando un alias del Policy Store, il principale necessita di entrambi:

- `verifiedpermissions:GetPolicyStoreAlias`autorizzazione per l'alias del policy store
- `verifiedpermissions:IsAuthorized`autorizzazione per il policy store associato

La policy di esempio seguente concede l'autorizzazione alla chiamata `IsAuthorized` utilizzando un alias specifico del policy store.

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

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

Per limitare gli alias del policy store che un principale può utilizzare, limita l'autorizzazione. `verifiedpermissions:GetPolicyStoreAlias` Ad esempio, il criterio seguente consente al principale di utilizzare qualsiasi alias dell'archivio delle politiche tranne quelli che iniziano con. `Restricted`

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

Schema di archiviazione delle politiche di Amazon Verified Permissions

Uno schema è una dichiarazione della struttura dei tipi di entità supportati dall'applicazione e delle azioni che l'applicazione può fornire nelle richieste di autorizzazione. Per vedere la differenza tra il modo in cui Verified Permissions e Cedar gestiscono gli schemi, consulta. Supporto dello schema

Per ulteriori informazioni, vedere il formato dello schema Cedar nella Guida di riferimento al linguaggio delle politiche Cedar.

> **Note**
>
> L'uso di schemi nelle autorizzazioni verificate è facoltativo, ma sono altamente consigliati per il software di produzione. Quando si crea una nuova politica, Verified Permissions può utilizzare lo schema per convalidare le entità e gli attributi a cui si fa riferimento nell'ambito e nelle condizioni, al fine di evitare errori di battitura ed errori nelle politiche che possono portare a un comportamento confuso del sistema. Se si attiva la convalida delle politiche, tutte le nuove politiche devono essere conformi allo schema.

Console di gestione AWS

Per creare uno schema

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Schema.
3. Scegliere Crea schema.

AWS CLI

Per inviare un nuovo schema o sovrascrivere uno schema esistente utilizzando il AWS CLI.

È possibile creare un policy store eseguendo un AWS CLI comando simile al seguente esempio.

Consideriamo uno schema che contenga il seguente contenuto Cedar:

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

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

Devi prima sfuggire al JSON in una stringa a riga singola, e prefigurarlo con una dichiarazione del suo tipo di dati:. `cedarJson` Il seguente esempio utilizza il seguente contenuto di un `schema.json` file che contiene la versione escape dello schema JSON.

> **Note**
>
> L'esempio qui è rifinito in righe per motivi di leggibilità. È necessario disporre dell'intero file su una sola riga affinché il comando lo accetti.

```
{"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

È possibile creare un archivio di politiche utilizzando l'`PutSchema`API. Per ulteriori informazioni, consulta PutSchemala Amazon Verified Permissions API Reference Guide.

Modifica degli schemi di archiviazione delle politiche

Quando selezioni Schema nella console Amazon Verified Permissions, vengono visualizzati i tipi di entità e le azioni che compongono lo schema. Puoi visualizzare e modificare lo schema in modalità Visual o JSON. La modalità visiva consente di aggiornare lo schema aggiungendo nuovi tipi e azioni utilizzando varie procedure guidate. Utilizzando la modalità JSON, puoi iniziare ad aggiornare il codice JSON dello schema direttamente nell'editor JSON.

Visual Mode

L'editor visivo dello schema inizia con una serie di diagrammi che illustrano le relazioni tra le entità dello schema. Scegli Espandi per massimizzare la visualizzazione dei diagrammi. Sono disponibili due diagrammi:

- Diagramma delle azioni: la visualizzazione del diagramma delle azioni elenca i tipi di Principal configurati nel Policy Store, le azioni che sono idonei a eseguire e le risorse su cui sono idonei a eseguire azioni. Le linee tra le entità indicano la possibilità di creare una politica che consenta a un responsabile di intraprendere un'azione su una risorsa. Se il diagramma delle azioni non indica una relazione tra due entità, è necessario creare tale relazione tra di esse prima di consentirla o negarla nelle politiche. Seleziona un'entità per visualizzare una panoramica delle proprietà ed espandi i dettagli per visualizzare tutti i dettagli. Scegli Filtra in base a questo [azione | tipo di risorsa | tipo principale] per vedere un'entità in una visualizzazione con solo le proprie connessioni.
- Diagramma dei tipi di entità: il diagramma dei tipi di entità si concentra sulle relazioni tra i principali e le risorse. Per comprendere le complesse relazioni principali annidate nello schema,

esaminate questo diagramma. Passa il mouse su un'entità per approfondire le relazioni principali che intrattiene.

Sotto i diagrammi sono elencate le visualizzazioni dei tipi di entità e delle azioni presenti nello schema. La visualizzazione elenco è utile quando si desidera visualizzare immediatamente i dettagli di un'azione o di un tipo di entità specifico. Seleziona qualsiasi entità per visualizzare i dettagli.

Per modificare uno schema di autorizzazioni verificate in modalità visiva

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Schema.
3. Scegli la modalità Visual. Esamina i diagrammi entità-relazione e pianifica le modifiche che desideri apportare allo schema. Facoltativamente, puoi filtrare in base a un'entità per esaminarne le connessioni individuali con altre entità.
4. Scegli Edit schema (Modifica schema).
5. Nella sezione Dettagli, digita un Namespace per lo schema.
6. Nella sezione Tipi di entità, scegli Aggiungi nuovo tipo di entità.
7. Digita il nome dell'entità.
8. (Facoltativo) Scegliete Aggiungi un genitore per aggiungere le entità principali di cui la nuova entità è membro. Per rimuovere un genitore che è stato aggiunto all'entità, scegli Rimuovi accanto al nome del genitore.
9. Scegli Aggiungi un attributo per aggiungere attributi all'entità. Digita il nome dell'attributo e scegli il tipo di attributo per ogni attributo dell'entità. Verified Permissions utilizza i valori degli attributi specificati per verificare le politiche rispetto allo schema. Seleziona se ogni attributo è obbligatorio. Per rimuovere un attributo che è stato aggiunto all'entità, scegli Rimuovi accanto all'attributo.
10. Scegli Aggiungi tipo di entità per aggiungere l'entità allo schema.
11. Nella sezione Azioni, scegli Aggiungi nuova azione.
12. Digita il nome dell'azione.
13. (Facoltativo) Scegliete Aggiungi una risorsa per aggiungere i tipi di risorse a cui si applica l'azione. Per rimuovere un tipo di risorsa che è stato aggiunto all'azione, scegli Rimuovi accanto al nome del tipo di risorsa.

14. (Facoltativo) Scegliete Aggiungi un principale per aggiungere un tipo principale a cui si applica l'azione. Per rimuovere un tipo principale che è stato aggiunto all'azione, scegliete Rimuovi accanto al nome del tipo principale.
15. Scegli Aggiungi un attributo per aggiungere attributi che possono essere aggiunti al contesto di un'azione nelle tue richieste di autorizzazione. Inserisci il nome dell'attributo e scegli il tipo di attributo per ogni attributo. Verified Permissions utilizza i valori degli attributi specificati per verificare le politiche rispetto allo schema. Seleziona se ogni attributo è obbligatorio. Per rimuovere un attributo che è stato aggiunto all'azione, scegli Rimuovi accanto all'attributo.
16. Selezionare Add action (Aggiungi operazione).
17. Dopo aver aggiunto tutti i tipi di entità e le azioni allo schema, scegli Salva modifiche.

JSON mode

Durante gli aggiornamenti, noterai che l'editor JSON convalida il codice in base alla sintassi JSON e identifica errori e avvisi durante le modifiche, facilitando la ricerca rapida dei problemi. Inoltre, non devi preoccuparti della formattazione del JSON, basta scegliere Format JSON dopo aver effettuato gli aggiornamenti e il formato verrà aggiornato in base alla formattazione JSON prevista.

Per modificare uno schema di autorizzazioni verificate in modalità JSON

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Schema.
3. Scegli la modalità JSON, quindi scegli Modifica schema.
4. Inserisci il contenuto dello schema JSON nel campo Contenuto. Non puoi salvare gli aggiornamenti dello schema finché non risolvi tutti gli errori di sintassi. Puoi scegliere Format JSON per formattare la sintassi JSON dello schema con la spaziatura e l'indentazione consigliate.
5. Scegli Save changes (Salva modifiche).

Attivazione della modalità di convalida delle policy di Amazon Verified Permissions

È possibile impostare la modalità di convalida delle politiche in Autorizzazioni verificate per controllare se le modifiche alle politiche vengono convalidate rispetto allo schema del proprio archivio delle politiche.

> **Important**
>
> Quando si attiva la convalida delle policy, tutti i tentativi di creare o aggiornare una policy o un modello di policy vengono convalidati in base allo schema presente nell'archivio delle policy. Verified Permissions rifiuta il tentativo di richiesta se la convalida fallisce. Per questo motivo, ti consigliamo di lasciare disattivata la convalida durante lo sviluppo dell'applicazione e di attivarla per il test e di lasciarla attiva mentre l'applicazione è in produzione.

Console di gestione AWS

Per impostare la modalità di convalida delle policy per un archivio di policy

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Seleziona Impostazioni.
3. Nella sezione Modalità di convalida della politica, scegli Modifica.
4. Esegui una di queste operazioni:
   - Per attivare la convalida delle politiche e imporre che tutte le modifiche alle politiche debbano essere convalidate rispetto allo schema, scegli il pulsante di opzione Strict (consigliato).
   - Per disattivare la convalida delle politiche per le modifiche alle politiche, scegli il pulsante di opzione Off. Digita `confirm` per confermare che gli aggiornamenti delle politiche non verranno più convalidati rispetto al tuo schema.
5. Scegli Save changes (Salva modifiche).

AWS CLI

Per impostare la modalità di convalida per un archivio di politiche

È possibile modificare la modalità di convalida per un policy store utilizzando l'UpdatePolicyStoreoperazione e specificando un valore diverso per il parametro. 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"
    }
}
```

Per ulteriori informazioni, vedere Convalida delle policy nella Cedar Policy Language Reference Guide.

Politiche di autorizzazione verificate di Amazon

Una politica è una dichiarazione che consente o proibisce a un preside di intraprendere una o più azioni su una risorsa. Ogni politica viene valutata indipendentemente da ogni altra politica. Per ulteriori informazioni su come sono strutturate e valutate le politiche Cedar, vedere la convalida delle politiche Cedar rispetto allo schema nella Guida di riferimento al linguaggio delle politiche Cedar.

Facoltativamente, è possibile assegnare un nome di politica a una politica. I nomi delle politiche devono essere univoci per tutte le politiche all'interno dell'archivio delle politiche e devono essere preceduti da. `name/` È possibile utilizzare il nome di una policy al posto dell'ID della policy nelle operazioni del piano di controllo che accettano un `policyId` parametro. L'esempio seguente utilizza il nome di una policy con `GetPolicy` cui recuperare una policy.

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

Important

Quando si scrivono politiche Cedar che fanno riferimento a principi, risorse e azioni, è possibile definire gli identificatori univoci utilizzati per ciascuno di questi elementi. Ti consigliamo vivamente di seguire queste best practice:

- Utilizzate identificatori universalmente univoci (UUIDs) per tutti gli identificatori principali e di risorse.

  Ad esempio, se un utente `jane` lascia l'azienda e in seguito consenti a qualcun altro di utilizzare il nome`jane`, quel nuovo utente ottiene automaticamente l'accesso a tutto ciò che è concesso dalle politiche che ancora fanno riferimento. `User::"jane"` Cedar non è in grado di distinguere tra il nuovo utente e il vecchio. Questo vale sia per gli identificatori principali che per quelli di risorse. Utilizza sempre identificatori che siano univoci garantiti e mai riutilizzati per assicurarti di non concedere involontariamente l'accesso a causa della presenza di un vecchio identificatore in una politica.

  Se utilizzi un UUID per un'entità, ti consigliamo di seguirlo con l'identificatore//comment e il nome «descrittivo» dell'entità. Questo aiuta a rendere le tue politiche più facili da capire. Ad esempio: principal == Role: :"a1b2c3d4-e5f6-a1b2-c3d4- «,//administrators EXAMPLE11111

- Non includete informazioni di identificazione personale, riservate o sensibili come parte dell'identificatore univoco dei vostri responsabili o delle vostre risorse. Questi identificatori sono inclusi nelle voci di registro condivise nei percorsi. AWS CloudTrail

Argomenti

Creazione di politiche statiche di Amazon Verified Permissions

È possibile creare una politica statica per consentire o vietare ai responsabili di eseguire azioni specifiche su risorse specifiche per l'applicazione. Una policy statica include valori specifici per `principal` `resource` e può essere utilizzata nelle decisioni di autorizzazione.

Console di gestione AWS

Per creare una politica statica

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy, quindi scegli Crea policy statica.

> **Note**
>
> Se hai una dichiarazione politica che desideri utilizzare, vai al passaggio 8 e incolla la politica nella sezione Politica nella pagina successiva.

4. Nella sezione Effetti delle politiche, scegli se la politica consentirà o proibirà quando una richiesta corrisponde alla politica. Se scegli Consenti, la politica consente ai mandanti di eseguire le azioni sulle risorse. Al contrario, se scegli Forbid, la politica non consente ai principali di eseguire le azioni sulle risorse.

5. Nel campo Ambito di applicazione dei principi, scegli l'ambito dei principi a cui verrà applicata la politica.

   - Scegli Principio specifico per applicare la politica a un principio specifico. Specificate il tipo di entità e l'identificatore del committente a cui sarà consentito o vietato intraprendere le azioni specificate nella politica.
   - Scegli Gruppo di responsabili per applicare la politica a un gruppo di responsabili. Digita il nome del gruppo principale nel campo Gruppo di dirigenti.
   - Scegli Tutti i responsabili per applicare la politica a tutti i mandanti del tuo archivio polizze.

6. Nel campo Ambito delle risorse, scegli l'ambito delle risorse a cui verrà applicata la politica.

   - Scegli Risorse specifiche per applicare la politica a una risorsa specifica. Specificate il tipo di entità e l'identificatore per la risorsa a cui deve essere applicata la politica.
   - Scegli Gruppo di risorse per applicare la politica a un gruppo di risorse. Digita il nome del gruppo di risorse nel campo Gruppo di risorse.
   - Scegli Tutte le risorse per applicare la politica a tutte le risorse del tuo archivio delle politiche.

7. Nella sezione Ambito delle azioni, scegli l'ambito delle risorse a cui verrà applicata la politica.

   - Scegli Set specifico di azioni per applicare la politica a un insieme di azioni. Seleziona le caselle di controllo accanto alle azioni per applicare la politica.
   - Scegli Tutte le azioni per applicare la policy a tutte le azioni nel tuo policy store.

8. Scegli Next (Successivo).
9. Nella sezione Politica, consulta la tua politica Cedar. Puoi scegliere Formato per formattare la sintassi della tua politica con la spaziatura e l'indentazione consigliate. Per ulteriori informazioni, vedere Costruzione delle politiche di base in Cedar nella Guida di riferimento al linguaggio delle politiche Cedar.
10. Nella sezione Dettagli, digita una descrizione facoltativa della politica.
11. Scegli Crea policy.

AWS CLI

Per creare una politica statica

È possibile creare una politica statica utilizzando l'CreatePolicyoperazione. L'esempio seguente crea una politica statica semplice.

```
$ 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"
}
```

Per creare una politica con un nome di politica

Facoltativamente, è possibile specificare un nome di politica durante la creazione di una politica. Il nome deve essere univoco per tutte le politiche all'interno dell'archivio delle politiche e deve essere preceduto da. `name/` È possibile utilizzare il nome al posto dell'ID della politica.

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

> **Note**
>
> Se si specifica un nome che è già associato a un'altra politica nel policy store, viene visualizzato un `ConflictException` errore.

Modifica delle politiche statiche di Amazon Verified Permissions

È possibile modificare una politica statica esistente nel proprio archivio delle politiche. È possibile aggiornare direttamente solo le politiche statiche. Per modificare una policy collegata a un modello, è necessario aggiornare il modello di policy. Per ulteriori informazioni, consulta Modifica dei modelli di policy di Amazon Verified Permissions.

È possibile modificare i seguenti elementi di una politica statica:

- A `action` cui fa riferimento la politica.
- Una clausola condizionale, ad esempio `when`. `unless`

Non è possibile modificare i seguenti elementi di una politica statica. Per modificare uno di questi elementi è necessario eliminare e ricreare la politica.

- Una politica da una politica statica a una politica collegata a un modello.
- L'effetto di una politica statica proveniente da o. `permit` `forbid`
- Il `principal` riferimento a cui fa riferimento una politica statica.
- Il `resource` referenziato da una politica statica.

Console di gestione AWS

Per modificare una politica statica

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli il pulsante di opzione accanto alla politica statica da modificare, quindi scegli Modifica.
4. Nella sezione Corpo della policy, aggiorna la clausola `action` or condition della policy statica. Non è possibile aggiornare l'effetto della politica o `resource` della politica. `principal`

5. Scegli Aggiorna policy.

> **Note**
>
> Se la convalida dei criteri è abilitata nel policy store, l'aggiornamento di un criterio statico fa sì che Verified Permissions convalidi la policy rispetto allo schema nel policy store. Se la policy statica aggiornata non supera la convalida, l'operazione ha esito negativo e l'aggiornamento non viene salvato.

AWS CLI

Per modificare una politica statica

È possibile modificare una politica statica utilizzando l'UpdatePolicyoperazione. L'esempio seguente modifica una politica statica semplice.

L'esempio utilizza il file `definition.txt` per contenere la definizione della politica.

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

Il comando seguente fa riferimento a quel file.

```
$ 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"
    }
}
```

Per aggiornare il nome di una politica

È possibile impostare o aggiornare il nome di una politica durante l'aggiornamento di una politica. Il nome deve essere univoco per tutte le politiche all'interno dell'archivio delle politiche e deve essere preceduto dan`ame/`. Se non includi il campo del nome nella richiesta di aggiornamento, il nome esistente rimane invariato. Per rimuovere un nome, impostalo su una stringa vuota.

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

Aggiungere un contesto

Il contesto è l'informazione rilevante per le decisioni politiche, ma non fa parte dell'identità del responsabile, dell'azione o della risorsa. Le rivendicazioni dei token di accesso sono contestuali.

Potresti voler consentire un'azione solo da un insieme di indirizzi IP di origine o solo se l'utente ha effettuato l'accesso con MFA. L'applicazione ha accesso a questi dati contestuali della sessione e deve inserirli nelle richieste di autorizzazione. I dati di contesto in una richiesta di autorizzazione Verified Permissions devono essere in formato JSON in un elemento. `contextMap`

Gli esempi che illustrano questo contenuto provengono da un esempio di policy store. A seguire, create il policy store DigitalPetStoredi esempio nel vostro ambiente di test.

Il seguente oggetto di contesto dichiara uno di ogni tipo di dati Cedar per un'applicazione basata sul DigitalPetStorepolicy store di esempio.

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

```
        "SSL": {
          "boolean": true
        }
    }
    },
    "RequestedOrderCount": {
      "long": 4
    },
    "UserAgent": {
      "string": "My UserAgent 1.12"
    }
  }
}
```

Tipi di dati nel contesto di autorizzazione

Booleano

Un binario `true` o un `false` valore. Nell'esempio, il valore booleano `true` for `MfaAuthenticated` indica che il cliente ha eseguito l'autenticazione a più fattori prima di richiedere la visualizzazione del proprio ordine.

Imposta

Una raccolta di elementi contestuali. I membri del set possono essere tutti dello stesso tipo, come in questo esempio, o di tipi diversi, incluso un set annidato. Nell'esempio, il cliente è associato a 3 account diversi.

Stringa

Una sequenza di lettere, numeri o simboli, racchiusa tra " caratteri. Nell'esempio, la `UserAgent` stringa rappresenta il browser utilizzato dal cliente per richiedere la visualizzazione dell'ordine.

Long

Come un intero, Nell'esempio, `RequestedOrderCount` indica che questa richiesta fa parte di un batch generato dalla richiesta del cliente di visualizzare quattro dei suoi ordini precedenti.

Registra

Una raccolta di attributi. È necessario dichiarare questi attributi nel contesto della richiesta. Un archivio di politiche con uno schema deve includere questa entità e gli attributi dell'entità nello schema. Nell'esempio, il `NetworkInfo` record contiene informazioni sull'IP di origine dell'utente, sulla geolocalizzazione di tale IP determinata dal client e sulla crittografia in transito.

EntityIdentifier

Un riferimento a un'entità e agli attributi dichiarati nell'`entities`elemento della richiesta. Nell'esempio, l'ordine dell'utente è stato approvato dal dipendenteBob.

Per testare questo contesto di esempio nell'DigitalPetStoreapp di esempio, è necessario aggiornare la richiesta`entities`, lo schema del policy store e la politica statica con la descrizione Customer Role - Get Order.

Modifica DigitalPetStore per accettare il contesto di autorizzazione

Inizialmente, non DigitalPetStoreè un archivio di policy molto complesso. Non include politiche o attributi di contesto preconfigurati per supportare il contesto che abbiamo presentato. Per valutare un esempio di richiesta di autorizzazione con queste informazioni di contesto, apporta le seguenti modifiche al tuo archivio delle politiche e alla tua richiesta di autorizzazione. Per esempi di contesto con informazioni sui token di accesso come contesto, vedere Mappatura dei token di Amazon Cognito accesso e Mappatura dei token di accesso OIDC.

Schema

Applica i seguenti aggiornamenti allo schema del tuo policy store per supportare i nuovi attributi di contesto. `GetOrder`Effettua l'aggiornamento `actions` come segue.

```
"GetOrder": {
  "memberOf": [],
  "appliesTo": {
    "resourceTypes": [
      "Order"
    ],
    "context": {
      "type": "Record",
      "attributes": {
        "AccountCodes": {
          "type": "Set",
          "required": true,
          "element": {
            "type": "Long"
          }
        },
        "approvedBy": {
          "name": "User",
```

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

Per fare riferimento al tipo di `record` dati indicato `NetworkInfo` nel contesto della richiesta, create un costrutto CommonType nello schema aggiungendo prima quanto segue allo schema. `actions` Un `commonType` costrutto è un insieme condiviso di attributi che puoi applicare a diverse entità.

```
"commonTypes": {
  "NetworkInfo": {
    "attributes": {
      "IPAddress": {
        "type": "String",
        "required": true
      },
      "SSL": {
        "required": true,
        "type": "Boolean"
      },
```

```
      "Country": {
        "required": true,
        "type": "String"
      }
    },
    "type": "Record"
  }
},
```

Policy

La seguente politica stabilisce le condizioni che devono essere soddisfatte da ciascuno degli elementi di contesto forniti. Si basa sulla politica statica esistente con la descrizione Customer Role - Get Order. Questa politica inizialmente richiede solo che il principale che effettua una richiesta sia il proprietario della risorsa.

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

Ora abbiamo richiesto che la richiesta di recupero di un ordine soddisfi le condizioni di contesto aggiuntive che abbiamo aggiunto alla richiesta.

1. L'utente deve aver effettuato l'accesso con MFA.
2. Il browser Web dell'utente `User-Agent` deve contenere la stringa`My UserAgent`.
3. L'utente deve aver richiesto di visualizzare 4 o meno ordini.
4. Uno dei codici dell'account dell'utente deve essere`111122223333`.

5. L'indirizzo IP dell'utente deve avere origine negli Stati Uniti d'America, deve trovarsi in una sessione crittografata e il suo indirizzo IP deve iniziare `192.0.2.` con.
6. Un dipendente deve aver approvato il proprio ordine. Nell'`entities`elemento della richiesta di autorizzazione, dichiareremo un utente `Bob` che ha il ruolo di`Employee`.

Request body

Dopo aver configurato l'archivio delle politiche con lo schema e la politica appropriati, puoi presentare questa richiesta di autorizzazione all'operazione dell'API Verified Permissions. IsAuthorized Tieni presente che il `entities` segmento contiene una definizione di`Bob`, un utente con un ruolo di`Employee`.

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

```
        "boolean": true
      },
      "NetworkInfo": {
        "record": {
          "Country": {"string": "United States of America"},
          "IPAddress": {"string": "192.0.2.178"},
          "SSL": {"boolean": true}
        }
      },
      "RequestedOrderCount":{
        "long": 4
      },
      "UserAgent": {
        "string": "My UserAgent 1.12"
      }
    }
  },
  "entities": {
    "entityList": [
      {
        "identifier": {
          "entityType": "DigitalPetStore::User",
          "entityId": "Alice"
        },
        "attributes": {
          "memberId": {
            "string": "801b87f2-1a5c-40b3-b580-eacad506d4e6"
          }
        },
        "parents": [
          {
            "entityType": "DigitalPetStore::Role",
            "entityId": "Customer"
          }
        ]
      },
      {
        "identifier": {
          "entityType": "DigitalPetStore::User",
          "entityId": "Bob"
        },
        "attributes": {
          "memberId": {
            "string": "49d9b81e-735d-429c-989d-93bec0bcfd8b"
```

```
          }
        },
        "parents": [
          {
            "entityType": "DigitalPetStore::Role",
            "entityId": "Employee"
          }
        ]
      },
      {
        "identifier": {
          "entityType": "DigitalPetStore::Order",
          "entityId": "1234"
        },
        "attributes": {
          "owner": {
            "entityIdentifier": {
              "entityType": "DigitalPetStore::User",
              "entityId": "Alice"
            }
          }
        },
        "parents": []
      }
     ]
   },
   "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Utilizzo del banco di prova Amazon Verified Permissions

Utilizza il banco di prova delle autorizzazioni verificate per testare e risolvere i problemi delle politiche di autorizzazione verificate eseguendo richieste di autorizzazione su di esse. Il banco di prova utilizza i parametri specificati dall'utente per determinare se le politiche Cedar presenti nell'archivio delle politiche autorizzerebbero la richiesta. È possibile passare dalla modalità Visual alla modalità JSON durante il test delle richieste di autorizzazione. Per ulteriori informazioni su come sono strutturate e valutate le politiche Cedar, vedere Costruzione delle politiche di base in Cedar nella Cedar Policy Language Reference Guide.

> **Note**
>
> Quando effettui una richiesta di autorizzazione utilizzando Verified Permissions, puoi fornire l'elenco dei principali e delle risorse come parte della richiesta nella sezione Entità aggiuntive. Tuttavia, non puoi includere i dettagli sulle azioni. Devono essere specificate nello schema o dedotte dalla richiesta. Non puoi inserire un'azione nella sezione Entità aggiuntive.

Per una panoramica visiva e una dimostrazione del banco di prova, consulta Amazon Verified Permissions - Policy Creation and Testing (Primer Series #3) sul AWS YouTube canale.

Visual mode

> **Note**
>
> È necessario disporre di uno schema definito nel proprio archivio di politiche per utilizzare la modalità visiva del banco di prova.

Per testare le politiche in modalità Visual

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Test bench.
3. Scegli la modalità Visual.
4. Nella sezione Principale, scegli il Principal che interviene tra i principali tipi del tuo schema. Digitate un identificatore per il principale nella casella di testo.
5. (Facoltativo) Scegliete Aggiungi un genitore per aggiungere entità principali per il principale specificato. Per rimuovere un genitore che è stato aggiunto al principale, scegli Rimuovi accanto al nome del genitore.
6. Specificate il valore dell'attributo per ogni attributo del principale specificato. Il banco di prova utilizza i valori degli attributi specificati nella richiesta di autorizzazione simulata.
7. Nella sezione Risorsa, scegli la risorsa su cui agisce il principale. Digita un identificatore per la risorsa nella casella di testo.
8. (Facoltativo) Scegliete Aggiungi un padre per aggiungere entità principali per la risorsa specificata. Per rimuovere un elemento principale che è stato aggiunto alla risorsa, scegliete Rimuovi accanto al nome del genitore.

9. Specificate il valore dell'attributo per ogni attributo della risorsa specificata. Il banco di prova utilizza i valori degli attributi specificati nella richiesta di autorizzazione simulata.
10. Nella sezione Azione, scegli l'azione che il principale sta eseguendo dall'elenco di azioni valide per il principale e la risorsa specificati.
11. Specificare il valore dell'attributo per ogni attributo dell'azione specificata. Il banco di prova utilizza i valori degli attributi specificati nella richiesta di autorizzazione simulata.
12. (Facoltativo) Nella sezione Entità aggiuntive, scegli Aggiungi entità per aggiungere entità da valutare per la decisione di autorizzazione.
13. Scegli l'identificatore dell'entità dall'elenco a discesa e digita l'identificatore dell'entità.
14. (Facoltativo) Scegli Aggiungi un genitore per aggiungere entità principali per l'entità specificata. Per rimuovere un padre che è stato aggiunto all'entità, scegli Rimuovi accanto al nome dell'entità principale.
15. Specificate il valore dell'attributo per ogni attributo dell'entità specificata. Il banco di prova utilizza i valori degli attributi specificati nella richiesta di autorizzazione simulata.
16. Scegli Conferma per aggiungere l'entità al banco di prova.
17. Scegli Esegui richiesta di autorizzazione per simulare la richiesta di autorizzazione per le politiche Cedar nel tuo policy store. Il banco di prova mostra la decisione di consentire o rifiutare la richiesta insieme alle informazioni sulle politiche soddisfatte o sugli errori riscontrati durante la valutazione.

JSON mode

Per testare le politiche in modalità JSON

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Test bench.
3. Scegli la modalità JSON.
4. Nella sezione Dettagli della richiesta, se hai definito uno schema, scegli il Principal che interviene tra i tipi principali del tuo schema. Digita un identificatore per il principale nella casella di testo.

   Se non avete definito uno schema, digitate il principale nella casella di testo Principal taking action.
5. Se hai definito uno schema, scegli la risorsa tra i tipi di risorse presenti nello schema. Digitate un identificatore per la risorsa nella casella di testo.

Se non avete uno schema definito, digitate la risorsa nella casella di testo Risorsa.

6. Se hai definito uno schema, scegli Azione dall'elenco di azioni valide per il principale e la risorsa specificati.

   Se non avete uno schema definito, digitate l'azione nella casella di testo Azione.

7. Immettete il contesto della richiesta da simulare nel campo Contesto. Il contesto della richiesta è costituito da informazioni aggiuntive che possono essere utilizzate per le decisioni di autorizzazione.
8. Nel campo Entità, inserisci la gerarchia delle entità e i relativi attributi da valutare per la decisione di autorizzazione.
9. Scegli Esegui richiesta di autorizzazione per simulare la richiesta di autorizzazione per le politiche Cedar nel tuo archivio di politiche. Il banco di prova mostra la decisione di consentire o rifiutare la richiesta insieme alle informazioni sulle politiche soddisfatte o sugli errori riscontrati durante la valutazione.

Esempi di politiche di Amazon Verified Permissions

Alcuni degli esempi di policy qui inclusi sono esempi di base di policy Cedar e altri sono specifici per Verified Permissions. Quelli di base si collegano alla Cedar Policy Language Reference Guide e vi sono inclusi. Per ulteriori informazioni sulla sintassi delle politiche Cedar, vedere Costruzione delle politiche di base in Cedar nella Cedar Policy Language Reference Guide.

Esempi di politiche

- Consente l'accesso a singole entità
- Consente l'accesso a gruppi di entità
- Consente l'accesso a qualsiasi entità
- Consente l'accesso agli attributi di un'entità (ABAC)
- Nega l'accesso
- Utilizza la notazione tra parentesi per fare riferimento agli attributi del token
- Utilizza la notazione a punti per fare riferimento agli attributi
- Riflette gli attributi del token ID Amazon Cognito
- Riflette gli attributi del token ID OIDC
- Riflette gli attributi dei token di accesso Amazon Cognito

- Riflette gli attributi del token di accesso OIDC

Utilizza la notazione tra parentesi per fare riferimento agli attributi del token

L'esempio seguente mostra come creare una politica che utilizzi la notazione tra parentesi per fare riferimento agli attributi del token.

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere Mappatura dei token allo schema e Mappatura Amazon Cognito dei token OIDC allo schema.

```
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.*"
};
```

Utilizza la notazione a punti per fare riferimento agli attributi

L'esempio seguente mostra come creare una politica che utilizzi la notazione a punti per fare riferimento agli attributi.

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere Mappatura dei token allo schema e Mappatura Amazon Cognito dei token OIDC allo schema.

```
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"
};
```

Riflette gli attributi del token ID Amazon Cognito

L'esempio seguente mostra come è possibile creare un riferimento di policy agli attributi del token ID Amazon Cognito.

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere Mappatura dei Amazon Cognito token allo schema e Mappatura dei token OIDC allo schema.

```
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"
};
```

Riflette gli attributi del token ID OIDC

L'esempio seguente mostra come è possibile creare una policy che faccia riferimento agli attributi del token ID da un provider OIDC.

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere Mappatura dei token allo schema e Mappatura Amazon Cognito dei token OIDC allo schema.

```
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*"
};
```

Riflette gli attributi dei token di accesso Amazon Cognito

L'esempio seguente mostra come è possibile creare una policy che faccia riferimento agli attributi del token di accesso Amazon Cognito.

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere Mappatura dei Amazon Cognito token allo schema e Mappatura dei token OIDC allo schema.

```
permit(principal, action in [MyApplication::Action::"Read",
 MyApplication::Action::"GetStoreInventory"], resource)
```

```
when {
    context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
    context.token.scope.contains("MyAPI/mydata.write")
};
```

Riflette gli attributi del token di accesso OIDC

L'esempio seguente mostra come è possibile creare una policy che faccia riferimento agli attributi del token di accesso da un provider OIDC.

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere Mappatura dei token allo schema e Mappatura Amazon Cognito dei token OIDC allo schema.

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

Modelli di policy di Amazon Verified Permissions e politiche collegate ai modelli

In Autorizzazioni verificate, i modelli di policy sono policy con segnaposto per o `principal` entrambi`resource`. I modelli di policy da soli non possono essere utilizzati per gestire le richieste di autorizzazione. Per gestire le richieste di autorizzazione, è necessario creare una policy collegata al modello basata su un modello di policy. I modelli di policy consentono di definire una policy una sola volta e di utilizzarla con più principi e risorse. Gli aggiornamenti al modello di policy si riflettono in tutte le policy che utilizzano il modello. Per ulteriori informazioni, consulta i modelli di policy Cedar nella Cedar Policy Language Reference Guide.

Facoltativamente, è possibile assegnare il nome di un modello di politica a un modello di politica. I nomi dei modelli di policy devono essere univoci all'interno dell'archivio delle politiche e devono essere preceduti da. `name/` È possibile utilizzare il nome di un modello di policy al posto dell'ID del modello di policy nelle operazioni del piano di controllo che accettano un `policyTemplateId` parametro. Solo `GetPolicyTemplate` e `ListPolicyTemplates` restituisce il nome nell'output. L'esempio seguente utilizza il nome di un modello di policy con `GetPolicyTemplate` cui recuperare un modello di policy.

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

Ad esempio, il seguente modello di policy fornisce `Read` e `Comment` autorizzazioni per il principale e la risorsa che utilizzano il modello di policy. `Edit`

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

Se si dovesse creare una politica denominata in `Editor` base a questo modello, quando un principale viene designato come editor per una risorsa specifica, l'applicazione creerebbe una politica che fornisce le autorizzazioni al principale per leggere, modificare e commentare la risorsa.

A differenza delle politiche statiche, le politiche collegate ai modelli sono dinamiche. Prendiamo l'esempio precedente, se si dovesse rimuovere l'`Comment`azione dal modello di policy, qualsiasi

policy collegata o basata su quel modello verrebbe aggiornata di conseguenza e i principi specificati nelle policy non sarebbero più in grado di commentare le risorse corrispondenti.

Per altri esempi di policy collegati ai modelli, consulta. Esempio di politiche collegate a modelli di Amazon Verified Permissions

Creazione di modelli di policy per le autorizzazioni verificate da Amazon

È possibile creare modelli di policy in Autorizzazioni verificate utilizzando il Console di gestione AWS, il AWS CLI, o il. AWS SDKs I modelli di policy consentono di definire una policy una sola volta e di utilizzarla con più principi e risorse. Una volta creato un modello di policy, è possibile creare policy collegate al modello per utilizzare i modelli di policy con principi e risorse specifici. Per ulteriori informazioni, consulta Creazione di politiche collegate ai modelli di Amazon Verified Permissions.

Console di gestione AWS

Come creare un modello di policy

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Modelli di policy.
3. Scegli Crea modello di policy.
4. Nella sezione Dettagli, digita una descrizione del modello di politica.
5. Nella sezione Corpo del modello di politica, utilizza i segnaposto `?principal` e consenti `?resource` alle politiche create sulla base di questo modello di personalizzare le autorizzazioni concesse. Puoi scegliere Formato per formattare la sintassi del tuo modello di policy con la spaziatura e l'indentazione consigliate.
6. Scegli Crea modello di policy.

AWS CLI

Come creare un modello di policy

È possibile creare un modello di policy utilizzando l'CreatePolicyTemplateoperazione. L'esempio seguente crea un modello di policy con un segnaposto per il principale.

Il file `template1.txt` contiene quanto segue.

```
"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"
}
```

Per creare un modello di policy con un nome di modello di policy

Facoltativamente, è possibile specificare il nome di un modello di politica durante la creazione di un modello di politica. Il nome deve essere univoco per tutti i modelli di policy all'interno dell'archivio delle politiche e deve essere preceduto da. `name/` È possibile utilizzare il nome al posto dell'ID del modello di policy.

```
$ aws verifiedpermissions create-policy-template \
    --description "Template for vacation picture access" \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111 \
    --name name/example-policy-template
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

> **Note**
>
> Se si specifica un nome che è già associato a un altro modello di policy nel Policy Store, viene visualizzato un `ConflictException` errore.

Creazione di politiche collegate ai modelli di Amazon Verified Permissions

È possibile creare policy collegate a un modello o politiche basate su un modello di policy utilizzando, o il. Console di gestione AWS AWS CLI AWS SDKs Le politiche collegate ai modelli rimangono collegate ai relativi modelli di policy. Se si modifica la dichiarazione di politica nel modello di politica, tutte le politiche collegate a tale modello utilizzano automaticamente la nuova dichiarazione per tutte le decisioni di autorizzazione prese da quel momento in poi.

Per esempi di policy collegati ai modelli, consulta. Esempio di politiche collegate a modelli di Amazon Verified Permissions

Console di gestione AWS

Per creare una policy collegata a un modello creando un'istanza di un modello di policy

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy, quindi scegli Crea policy collegata al modello.
4. Scegli il pulsante di opzione accanto al modello di policy da utilizzare, quindi scegli Avanti.
5. Digita il Principal e la Risorsa da utilizzare per questa istanza specifica della policy collegata al modello. I valori specificati vengono visualizzati nel campo di anteprima della dichiarazione politica.

   > **Note**
   >
   > I valori Principal e Resource devono avere la stessa formattazione delle politiche statiche. Ad esempio, per specificare il `AdminUsers` gruppo per il principale, digitate`Group::"AdminUsers"`. Se digitate`AdminUsers`, viene visualizzato un errore di convalida.

6. Scegli Crea politica collegata al modello.

   La nuova politica collegata al modello viene visualizzata in Politiche.

AWS CLI

Per creare una policy collegata al modello creando un'istanza di un modello di policy

È possibile creare una politica collegata a un modello che faccia riferimento a un modello di politica esistente e che specifichi i valori per tutti i segnaposto utilizzati dal modello.

L'esempio seguente crea una politica collegata al modello che utilizza un modello con la seguente dichiarazione:

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

Utilizza inoltre il seguente `definition.txt` file per fornire il valore per il parametro: `definition`

```
{
    "templateLinked": {
        "policyTemplateId": "PTEXAMPLEabcdefg111111",
        "principal": {
            "entityType": "PhotoFlash::User",
            "entityId": "alice"
        }
    }
}
```

L'output mostra sia la risorsa, ottenuta dal modello, sia la risorsa principale, che ottiene dal parametro di definizione

```
$ 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": "PhotoFlash::User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "PhotoFlash::Photo"
    }
}
```

Modifica dei modelli di policy di Amazon Verified Permissions

È possibile modificare o aggiornare i modelli di policy in Autorizzazioni verificate utilizzando il Console di gestione AWS, il AWS CLI, o il AWS SDKs. La modifica di un modello di policy aggiornerà automaticamente i criteri collegati o basati sul modello, quindi fai attenzione quando modifichi i modelli di policy e assicurati di non introdurre accidentalmente una modifica che danneggi l'applicazione.

È possibile modificare i seguenti elementi di un modello di policy:

- A cui `action` fa riferimento il modello di policy
- Una clausola condizionale, ad esempio e `when` `unless`

Non è possibile modificare i seguenti elementi di un modello di policy. Per modificare uno qualsiasi di questi elementi è necessario eliminare e ricreare il modello di policy.

- L'effetto di un modello di policy proveniente da `permit` o `forbid`
- Il `principal` riferimento a cui fa riferimento un modello di policy
- Il `resource` riferimento a cui fa riferimento un modello di policy

Console di gestione AWS

Per modificare i modelli di policy

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Modelli di policy. La console mostra tutti i modelli di policy che hai creato nell'archivio delle politiche corrente.
3. Scegli il pulsante di opzione accanto a un modello di policy per visualizzare i dettagli sul modello di policy, ad esempio quando il modello di policy è stato creato, aggiornato e il contenuto del modello di policy.
4. Scegli Modifica per modificare il modello di policy. Aggiorna la descrizione della politica e il corpo della politica secondo necessità, quindi scegli Aggiorna modello di politica.
5. È possibile eliminare un modello di politica selezionando il pulsante di opzione accanto a un modello di politica e quindi scegliendo Elimina. Scegli OK per confermare l'eliminazione del modello di policy.

AWS CLI

Per modificare un modello di policy

È possibile creare una politica statica utilizzando l'UpdatePolicyoperazione. L'esempio seguente aggiorna il modello di policy specificato sostituendo il relativo corpo della policy con un nuovo criterio definito in un file.

Contenuto del 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"
}
```

Per aggiornare il nome di un modello di policy

È possibile impostare o aggiornare il nome di un modello di policy durante l'aggiornamento di un modello di policy. Il nome deve essere univoco per tutti i modelli di policy all'interno dell'archivio delle politiche e deve essere preceduto dan`ame/`. Se non includi il campo del nome nella richiesta di aggiornamento, il nome esistente rimane invariato. Per rimuovere un nome, impostalo su una stringa vuota.

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

Esempio di politiche collegate a modelli di Amazon Verified Permissions

Quando crei un archivio delle politiche in Autorizzazioni verificate utilizzando il metodo Sample policy store, il tuo archivio delle politiche viene creato con politiche predefinite, modelli di policy e uno schema per il progetto di esempio che hai scelto. I seguenti esempi di policy collegati al modello Verified Permissions possono essere utilizzati con gli archivi di policy di esempio e i rispettivi criteri, modelli di policy e schemi.

PhotoFlash esempi

L'esempio seguente mostra come è possibile creare una politica collegata al modello che utilizza il modello di politica Garantire l'accesso limitato a foto condivise non private con un singolo utente e una foto.

> **Note**
>
> Cedar Policy Language considera un'entità come se stessa. `in` Pertanto, `principal in User::"Alice"` è equivalente a. `principal == User::"Alice"`

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

L'esempio seguente mostra come è possibile creare una politica collegata al modello che utilizza il modello di politica Garantire l'accesso limitato a foto condivise non private con un singolo utente e album.

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

L'esempio seguente mostra come è possibile creare una politica collegata al modello che utilizza il modello di politica Garantire l'accesso limitato a foto condivise non private con un gruppo di amici e una singola foto.

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

L'esempio seguente mostra come è possibile creare una politica collegata al modello che utilizza il modello di politica Garantire l'accesso limitato a foto condivise non private con un gruppo di amici e un album.

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

L'esempio seguente mostra come creare una politica collegata al modello che utilizza il modello di politica Garantire l'accesso completo alle foto condivise non private con un gruppo di amici e una singola foto.

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

L'esempio seguente mostra come creare una policy collegata a un modello che utilizza il modello di policy Blocca utente da un account.

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

DigitalPetStore esempi

L'archivio DigitalPetStore di policy di esempio non include alcun modello di policy. È possibile visualizzare le politiche incluse nel Policy Store scegliendo Policy nel riquadro di navigazione a sinistra dopo aver creato il Policy Store di DigitalPetStoreesempio.

TinyToDo esempi

L'esempio seguente mostra come è possibile creare una policy collegata al modello che utilizza il modello di policy che consente agli utenti di accedere a un singolo utente e a un elenco di attività.

```
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"
);
```

L'esempio seguente mostra come è possibile creare una politica collegata al modello che utilizza il modello di policy che consente l'accesso all'editor per un singolo utente e un elenco di attività.

```
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"
);
```

Proteggi le tue applicazioni con fonti di identità e token

Proteggi rapidamente le tue applicazioni creando una fonte di identità per rappresentare un provider di identità esterno (IdP) in Amazon Verified Permissions. Le fonti di identità forniscono informazioni su un utente che si è autenticato con un IdP che ha una relazione di fiducia con il tuo policy store. Quando l'applicazione effettua una richiesta di autorizzazione con un token proveniente da una fonte di identità, il policy store può prendere decisioni di autorizzazione sulla base delle proprietà dell'utente e delle autorizzazioni di accesso. Puoi aggiungere un pool di utenti Amazon Cognito o un IdP OpenID Connect (OIDC) personalizzato come fonte di identità.

Puoi utilizzare i provider di identità OpenID Connect (OIDC) () con autorizzazioni IdPs verificate. La tua applicazione può generare richieste di autorizzazione con token web JSON (JWTs) generati da un provider di identità conforme a OIDC. L'identità dell'utente nel token è mappata all'ID principale. Con i token ID, Verified Permissions associa le rivendicazioni degli attributi agli attributi principali. Con i token di accesso, queste affermazioni vengono mappate in base al contesto. Con entrambi i tipi di token, puoi mappare un claim come se fosse `groups` un gruppo principale e creare policy che valutino il controllo degli accessi basato sui ruoli (RBAC).

> **Note**
>
> Verified Permissions prende decisioni di autorizzazione sulla base delle informazioni di un token IdP ma non interagisce direttamente con l'IdP in alcun modo.

Per una step-by-step procedura dettagliata che crea una logica di autorizzazione per Amazon API Gateway REST APIs utilizzando un pool di Amazon Cognito utenti o un provider di identità OIDC, consulta Autorizza utilizzando API Gateway APIs Amazon Verified Permissions with Amazon Cognito or bring your own identity provider sul Security Blog.AWS

Argomenti

- Scelta del provider di identità giusto
- Lavorare con le fonti di Amazon Cognito identità
- Utilizzo delle fonti di identità OIDC

Scelta del provider di identità giusto

Sebbene Verified Permissions funzioni con una varietà di autorizzazioni IdPs, tieni presente quanto segue quando decidi quale utilizzare nella tua applicazione:

Da utilizzare quando Amazon Cognito :

- Stai creando nuove applicazioni senza l'infrastruttura di identità esistente
- Desideri pool AWS di utenti gestiti con funzionalità di sicurezza integrate
- È necessaria l'integrazione con un provider di identità social
- Desideri una gestione semplificata dei token

Utilizza i provider OIDC quando:

- Hai un'infrastruttura di identità esistente (Auth0, Okta, Azure AD)
- È necessario mantenere una gestione centralizzata degli utenti
- Hai requisiti di conformità per specifici IdPs

Lavorare con le fonti di Amazon Cognito identità

Verified Permissions lavora a stretto contatto con i pool di utenti di Amazon Cognito. Amazon Cognito JWTs hanno una struttura prevedibile. Verified Permissions riconosce questa struttura e trae il massimo beneficio dalle informazioni in essa contenute. Ad esempio, è possibile implementare un modello di autorizzazione per il controllo degli accessi basato sui ruoli (RBAC) con token ID o token di accesso.

Una nuova fonte di identità per i pool di utenti di Amazon Cognito richiede le seguenti informazioni:

- Il Regione AWS.
- L'ID pool di utenti.
- Il tipo di entità principale che desideri associare alla fonte della tua identità, ad esempio`MyCorp::User`.
- Il tipo di entità di gruppo principale che desideri associare alla tua fonte di identità, ad esempio`MyCorp::UserGroup`.
- Il client IDs del tuo pool di utenti che desideri autorizzare a effettuare richieste al tuo policy store.

Poiché Verified Permissions funziona solo con i pool di utenti di Amazon Cognito nello Account AWS stesso account, non puoi specificare una fonte di identità in un altro account. Verified Permissions imposta il prefisso dell'entità, l'identificatore dell'identità e della fonte a cui devi fare riferimento nelle politiche che agiscono sui principali del pool di utenti, all'ID del tuo pool di utenti, ad esempio. `us-west-2_EXAMPLE` In questo caso, faresti riferimento a un utente in quel pool di utenti con ID come `a1b2c3d4-5678-90ab-cdef-EXAMPLE22222` `us-west-2_EXAMPLE|a1b2c3d4-5678-90ab-cdef-EXAMPLE22222`

Le dichiarazioni relative ai token del pool di utenti possono contenere attributi, ambiti, gruppi IDs, client e dati personalizzati. Amazon Cognito JWTshanno la capacità di includere una varietà di informazioni che possono contribuire alle decisioni di autorizzazione nelle autorizzazioni verificate. Ciò include:

1. Nome utente e affermazioni di gruppo con prefisso `cognito:`
2. Attributi utente personalizzati con un `custom: prefix`
3. Affermazioni personalizzate aggiunte in fase di esecuzione
4. Dichiarazioni standard OIDC come e `sub` `email`

Tratteremo in dettaglio queste affermazioni e come gestirle nelle politiche sulle autorizzazioni verificate, in. Mappatura dei Amazon Cognito token sullo schema

> **Important**
>
> Sebbene sia possibile revocare Amazon Cognito i token prima della scadenza, JWTs sono considerati risorse stateless, autonome e dotate di firma e validità. I servizi conformi al token Web JSON RFC 7519 dovrebbero convalidare i token in remoto e non sono tenuti a convalidarli con l'emittente. Ciò significa che è possibile che Verified Permissions conceda l'accesso in base a un token revocato o rilasciato a un utente che è stato successivamente eliminato. Per mitigare questo rischio, ti consigliamo di creare i token con la durata di validità più breve possibile e di revocare i token di aggiornamento quando desideri rimuovere l'autorizzazione a continuare la sessione di un utente. Per ulteriori informazioni, consulta Terminare le sessioni utente con revoca dei token

L'esempio seguente mostra come creare una policy che faccia riferimento ad alcune delle dichiarazioni dei pool di utenti di Amazon Cognito associate a un'entità principale.

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

L'esempio seguente mostra come creare una politica che faccia riferimento a un principale che è un utente in un pool di utenti di Cognito. Nota che l'ID principale assume la forma di"`<userpool-id>|<sub>`".

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

Le politiche Cedar per le fonti di identità del pool di utenti in Verified Permissions utilizzano una sintassi speciale per i nomi delle rivendicazioni che contengono caratteri diversi da quelli alfanumerici e dal carattere di sottolineatura (). _ Ciò include le dichiarazioni di prefisso del pool di utenti che contengono un carattere, come e. `:` `cognito:username` `custom:department` Per scrivere una condizione politica che faccia riferimento al `custom:department` claim `cognito:username` o, scrivila rispettivamente come `principal["cognito:username"]` e`principal["custom:department"]`.

> **Note**
>
> Se un token contiene un'attestazione con un `custom:` prefisso `cognito:` or e un nome di attestazione con valore letterale `cognito` o`custom`, una richiesta di autorizzazione con IsAuthorizedWithTokenun. `ValidationException`

Per ulteriori informazioni sulla mappatura delle attestazioni, consulta. Mappatura dei Amazon Cognito token sullo schema Per ulteriori informazioni sull'autorizzazione per Amazon Cognito gli utenti, consulta Authorization with Amazon Verified Permissions nella Amazon Cognito Developer Guide.

Argomenti

Creazione di fonti di Amazon Cognito identità Amazon Verified Permissions

La procedura seguente aggiunge un'origine di identità a un archivio di politiche esistente.

È inoltre possibile creare un'origine di identità quando si crea un nuovo archivio di politiche nella console Autorizzazioni verificate. In questo processo, puoi importare automaticamente le attestazioni contenute nei token di origine dell'identità negli attributi dell'entità. Scegli l'opzione Configurazione guidata o Configura con API Gateway e un provider di identità. Queste opzioni creano anche politiche iniziali.

> **Note**
>
> Le fonti di identità non sono disponibili nel riquadro di navigazione a sinistra fino a quando non è stato creato un archivio delle politiche. Le fonti di identità create sono associate al policy store corrente.

Puoi omettere il tipo di entità principale quando crei una fonte di identità con AWS CLI o create-identity-sourceCreateIdentitySourcenell'API Verified Permissions. Tuttavia, un tipo di entità vuoto crea una fonte di identità con un tipo di entità di`AWS::Cognito`. Questo nome di entità non è compatibile con lo schema dell'archivio delle politiche. Per integrare Amazon Cognito le identità con lo schema dell'archivio delle politiche, è necessario impostare il tipo di entità principale su un'entità dell'archivio delle politiche supportata.

Console di gestione AWS

Per creare una fonte di identità per pool di utenti Amazon Cognito

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Identity sources.
3. Scegli Crea fonte di identità.

4. Nei dettagli del pool di utenti di Cognito, seleziona Regione AWS e inserisci l'ID del pool di utenti per la tua origine di identità.
5. Nella configurazione principale, per Tipo principale, scegli il tipo di entità per i principali da questa fonte. Le identità dei pool di utenti Amazon Cognito connessi verranno mappate sul tipo principale selezionato.
6. Nella configurazione del gruppo, seleziona Usa il gruppo Cognito se desideri mappare il claim del pool `cognito:groups` di utenti. Scegli un tipo di entità che sia padre del tipo principale.
7. In Convalida dell'applicazione client, scegli se convalidare l'applicazione client. IDs
   - Per convalidare l'applicazione client IDs, scegli Accetta solo token con l'applicazione client corrispondente. IDs Scegli Aggiungi nuovo ID dell'applicazione client per ogni ID dell'applicazione client da convalidare. Per rimuovere un ID dell'applicazione client che è stato aggiunto, scegli Rimuovi accanto all'ID dell'applicazione client.
   - Scegliete Non convalidare l'applicazione client IDs se non desiderate convalidare l'applicazione client. IDs
8. Scegli Crea origine di identità.
9. (Facoltativo) Se il vostro policy store dispone di uno schema, prima di poter fare riferimento agli attributi estratti dall'identità o dai token di accesso nelle policy Cedar, dovete aggiornare lo schema per rendere Cedar consapevole del tipo di principale creato dalla vostra fonte di identità. Tale aggiunta allo schema deve includere gli attributi a cui desiderate fare riferimento nelle vostre politiche Cedar. Per ulteriori informazioni sulla mappatura degli attributi dei Amazon Cognito token agli attributi principali di Cedar, vedere. Mappatura dei Amazon Cognito token sullo schema

   > **Note**
   >
   > Quando crei un policy store collegato all'API o utilizzi Set up with API Gateway e un provider di identità durante la creazione di policy store, Verified Permissions interroga il pool di utenti per gli attributi utente e crea uno schema in cui il tipo principale è popolato con gli attributi del pool di utenti.

10. Crea politiche che utilizzano le informazioni dei token per prendere decisioni di autorizzazione. Per ulteriori informazioni, consulta Creazione di politiche statiche di Amazon Verified Permissions.

Ora che hai creato una fonte di identità, aggiornato lo schema e creato le politiche, consenti `IsAuthorizedWithToken` alle Autorizzazioni Verificate di prendere decisioni di autorizzazione. Per ulteriori informazioni, consulta IsAuthorizedWithTokenla guida di riferimento dell'API Amazon Verified Permissions.

AWS CLI

Per creare una fonte di identità per pool di utenti Amazon Cognito

Puoi creare una fonte di identità utilizzando l'CreateIdentitySourceoperazione. L'esempio seguente crea un'origine di identità in grado di accedere alle identità autenticate da un pool di Amazon Cognito utenti.

1. Crea un `config.txt` file che contenga i seguenti dettagli del pool di Amazon Cognito utenti da utilizzare con il `--configuration` parametro del comando. `create-identity-source`

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

2. Esegui il comando seguente per creare una fonte di Amazon Cognito identità.

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

3. (Facoltativo) Se il vostro policy store dispone di uno schema, prima di poter fare riferimento agli attributi che estraete dall'identità o dai token di accesso nelle vostre policy Cedar, dovete aggiornare lo schema per rendere Cedar consapevole del tipo di principale creato dalla vostra fonte di identità. Tale aggiunta allo schema deve includere gli attributi a cui desiderate fare riferimento nelle vostre politiche Cedar. Per ulteriori informazioni sulla mappatura degli attributi dei Amazon Cognito token agli attributi principali di Cedar, vedere. Mappatura dei Amazon Cognito token sullo schema

   > **Note**
   >
   > Quando crei un policy store collegato all'API o utilizzi Set up with API Gateway e un provider di identità durante la creazione di policy store, Verified Permissions interroga il pool di utenti per gli attributi utente e crea uno schema in cui il tipo principale è popolato con gli attributi del pool di utenti.

4. Crea politiche che utilizzano le informazioni dei token per prendere decisioni di autorizzazione. Per ulteriori informazioni, consulta Creazione di politiche statiche di Amazon Verified Permissions.

Ora che hai creato una fonte di identità, aggiornato lo schema e creato le politiche, consenti `IsAuthorizedWithToken` alle Autorizzazioni Verificate di prendere decisioni di autorizzazione. Per ulteriori informazioni, consulta IsAuthorizedWithTokenla guida di riferimento dell'API Amazon Verified Permissions.

Per ulteriori informazioni sull'utilizzo dei token di accesso e identità di Amazon Cognito per gli utenti autenticati in Autorizzazioni verificate, consulta Authorization with Amazon Verified Permissions nella Amazon Cognito Developer Guide.

Modifica delle fonti di Amazon Cognito identità di Amazon Verified Permissions

Puoi modificare alcuni parametri della tua fonte di identità dopo averla creata. Non puoi cambiare il tipo di fonte di identità, devi eliminare l'origine dell'identità e crearne una nuova da cui passare Amazon Cognito a OIDC o OIDC. Amazon Cognito Se lo schema del policy store corrisponde agli attributi della fonte di identità, tieni presente che devi aggiornare lo schema separatamente per riflettere le modifiche apportate alla tua origine di identità.

Console di gestione AWS

Per aggiornare una fonte di Amazon Cognito identità

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Identity sources.
3. Scegli l'ID della fonte di identità da modificare.
4. Scegli Modifica.
5. Nei dettagli del pool di utenti di Cognito, seleziona Regione AWS e digita l'ID del pool di utenti per la tua origine di identità.
6. Nei dettagli del principale, puoi aggiornare il tipo di Principal per la fonte dell'identità. Le identità dei pool di utenti Amazon Cognito connessi verranno mappate sul tipo principale selezionato.
7. Nella configurazione del gruppo, seleziona Usa i gruppi di Cognito se desideri mappare la dichiarazione del pool `cognito:groups` di utenti. Scegli un tipo di entità che sia padre del tipo principale.
8. In Convalida dell'applicazione client, scegli se convalidare l'applicazione client. IDs
   - Per convalidare l'applicazione client IDs, scegli Accetta solo token con l'applicazione client corrispondente. IDs Scegli Aggiungi nuovo ID dell'applicazione client per ogni ID dell'applicazione client da convalidare. Per rimuovere un ID dell'applicazione client che è stato aggiunto, scegli Rimuovi accanto all'ID dell'applicazione client.
   - Scegliete Non convalidare l'applicazione client IDs se non desiderate convalidare l'applicazione client. IDs
9. Scegli Save changes (Salva modifiche).
10. Se hai modificato il tipo principale per l'origine dell'identità, devi aggiornare lo schema in modo che rifletta correttamente il tipo principale aggiornato.

È possibile eliminare una fonte di identità scegliendo il pulsante di opzione accanto a una fonte di identità e quindi scegliendo Elimina fonte di identità. Digita `delete` nella casella di testo, quindi scegli Elimina fonte di identità per confermare l'eliminazione della fonte di identità.

AWS CLI

Per aggiornare una fonte di Amazon Cognito identità

È possibile aggiornare una fonte di identità utilizzando l'UpdateIdentitySourceoperazione. L'esempio seguente aggiorna l'origine di identità specificata per utilizzare un pool di Amazon Cognito utenti diverso.

1. Create un `config.txt` file che contenga i seguenti dettagli del pool di Amazon Cognito utenti da utilizzare con il `--configuration` parametro del `update-identity-source` comando.

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

2. Esegui il comando seguente per aggiornare una fonte di Amazon Cognito identità.

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

> **Note**
>
> Se si modifica il tipo principale per l'origine dell'identità, è necessario aggiornare lo schema in modo che rifletta correttamente il tipo principale aggiornato.

Mappatura dei Amazon Cognito token sullo schema

Potresti scoprire di voler aggiungere una fonte di identità a un archivio delle politiche e mappare le attestazioni, o token, del provider allo schema del tuo archivio delle politiche. È possibile automatizzare questo processo utilizzando la configurazione guidata per creare il proprio Policy Store con una fonte di identità o aggiornare lo schema manualmente dopo la creazione del Policy Store. Dopo aver mappato i token allo schema, è possibile creare policy che vi fanno riferimento.

Questa sezione della guida per l'utente contiene le seguenti informazioni:

- Quando è possibile compilare automaticamente gli attributi in uno schema di policy store
- Come utilizzare le attestazioni relative ai Amazon Cognito token nelle politiche relative alle autorizzazioni verificate
- Come creare manualmente uno schema per una fonte di identità

Gli archivi di policy collegati alle API e gli archivi di policy con una fonte di identità creati tramite la configurazione guidata non richiedono la mappatura manuale degli attributi del token di identità (ID) allo schema. Puoi fornire autorizzazioni verificate con gli attributi del tuo pool di utenti e creare uno schema popolato con attributi utente. Nell'autorizzazione con token ID, Verified Permissions associa le rivendicazioni agli attributi di un'entità principale. Potrebbe essere necessario mappare manualmente Amazon Cognito i token allo schema nelle seguenti condizioni:

- È stato creato un policy store o un policy store vuoto a partire da un esempio.
- Desiderate estendere l'uso dei token di accesso oltre il controllo degli accessi basato sui ruoli (RBAC).
- Puoi creare archivi di policy con l'API REST di Verified Permissions, un SDK o il. AWS AWS CDK

Per utilizzarli Amazon Cognito come fonte di identità nel tuo archivio di policy per le autorizzazioni verificate, devi avere gli attributi del provider nello schema. Lo schema è fisso e deve corrispondere alle entità create dai token del provider IsAuthorizedWithTokeno alle richieste BatchIsAuthorizedWithTokenAPI. Se hai creato il tuo archivio delle politiche in modo da compilare automaticamente lo schema con le informazioni del provider in un token ID, sei pronto per scrivere le policy. Se crei un policy store senza uno schema per la tua origine di identità, devi aggiungere gli attributi del provider allo schema che corrispondono alle entità create utilizzando le richieste API. È quindi possibile scrivere politiche utilizzando gli attributi del token del provider.

Per ulteriori informazioni sull'utilizzo dell'ID Amazon Cognito e dei token di accesso per gli utenti autenticati in Autorizzazioni verificate, consulta Authorization with Amazon Verified Permissions nella Amazon Cognito Developer Guide.

Argomenti

- Mappatura dei token ID allo schema
- Mappatura dei token di accesso
- Notazione alternativa per Amazon Cognito affermazioni delimitate da due punti
- Cose da sapere sulla mappatura degli schemi

Mappatura dei token ID allo schema

Verified Permissions elabora le dichiarazioni relative ai token ID come attributi dell'utente: nomi e titoli, appartenenza al gruppo, informazioni di contatto. I token ID sono molto utili in un modello di autorizzazione ABAC (Attribute-Based Access Control). Se desideri che le autorizzazioni verificate analizzino l'accesso alle risorse in base a chi effettua la richiesta, scegli i token ID come fonte della tua identità.

Amazon Cognito I token ID funzionano con la maggior parte delle librerie relying-party OIDC. Estendono le funzionalità di OIDC con affermazioni aggiuntive. L'applicazione può autenticare l'utente con le operazioni API di autenticazione dei pool di utenti di Amazon Cognito o con l'interfaccia utente ospitata dal pool di utenti. Per ulteriori informazioni, consulta Using the API and endpoints nella Developer Guide.Amazon Cognito

Dichiarazioni utili nei token Amazon Cognito ID

`cognito:username` e *`preferred_username`*

Varianti del nome utente dell'utente.

`sub`

L'identificatore utente univoco (UUID) dell'utente

Affermazioni con un prefisso *`custom:`*

Un prefisso per attributi personalizzati del pool di utenti come. `custom:employmentStoreCode`

Affermazioni standard

Dichiarazioni OIDC standard come `email` e. `phone_number` Per ulteriori informazioni, consulta Dichiarazioni standard in OpenID Connect Core 1.0 che incorporano il set di errata 2.

`cognito:groups`

Appartenenze ai gruppi di un utente. In un modello di autorizzazione basato sul controllo degli accessi basato sui ruoli (RBAC), questa dichiarazione presenta i ruoli che è possibile valutare nelle politiche.

Reclami transitori

Affermazioni che non sono di proprietà dell'utente, ma vengono aggiunte in fase di esecuzione da un trigger Lambda prima della generazione di token del pool di utenti. Le affermazioni transitorie assomigliano alle affermazioni standard ma non rientrano nello standard, ad esempio o. `tenant` `department`

Nelle politiche che fanno riferimento Amazon Cognito agli attributi che dispongono di un `:` separatore, fate riferimento agli attributi nel formato. `principal["`*`cognito:username`*`"]` L'affermazione dei ruoli `cognito:groups` è un'eccezione a questa regola. Verified Permissions associa il contenuto di questa dichiarazione alle entità principali dell'entità utente.

Per ulteriori informazioni sulla struttura dei token ID dei pool di utenti di Amazon Cognito, consulta Using the ID token nella Amazon Cognito Developer Guide.

Il seguente esempio di token ID ha ciascuno dei quattro tipi di attributi. Include l' Amazon Cognito attestazione specifica`cognito:username`, l'attestazione personalizzata`custom:employmentStoreCode`, l'attestazione standard e l'`email`attestazione transitoria. `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"
}
```

Quando crei una fonte di identità con il tuo pool di Amazon Cognito utenti, specifichi il tipo di entità principale con cui Verified Permissions genera le richieste di autorizzazione. `IsAuthorizedWithToken` Le tue politiche possono quindi testare gli attributi di tale principale come parte della valutazione della richiesta. Lo schema definisce il tipo e gli attributi principali per una fonte di identità, quindi è possibile farvi riferimento nelle politiche Cedar.

Specificate anche il tipo di entità di gruppo che desiderate derivare dal claim ID Token Groups. Nelle richieste di autorizzazione, Verified Permissions associa ogni membro della dichiarazione di gruppo a quel tipo di entità di gruppo. Nelle politiche, puoi fare riferimento a quell'entità di gruppo come principale.

L'esempio seguente mostra come riflettere gli attributi del token di identità di esempio nello schema di autorizzazioni verificate. Per ulteriori informazioni sulla modifica dello schema, consultaModifica degli schemi di archiviazione delle politiche. Se la configurazione dell'origine dell'identità specifica il tipo principale`User`, puoi includere qualcosa di simile al seguente esempio per rendere tali attributi disponibili a 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
        }
     }
  }
}
```

Per un esempio di politica che verrà convalidata in base a questo schema, vedi. Riflette gli attributi del token ID Amazon Cognito

Mappatura dei token di accesso

Verified Permissions elabora le dichiarazioni dei token di accesso diverse da quelle dichiarate dai gruppi come attributi dell'azione o attributi di contesto. Oltre all'appartenenza al gruppo, i token di accesso del tuo IdP potrebbero contenere informazioni sull'accesso alle API. I token di accesso sono utili nei modelli di autorizzazione che utilizzano il controllo degli accessi basato sui ruoli (RBAC). I modelli di autorizzazione che si basano su richieste di token di accesso diverse dall'appartenenza al gruppo richiedono uno sforzo aggiuntivo nella configurazione dello schema.

Amazon Cognito i token di accesso hanno affermazioni che possono essere utilizzate per l'autorizzazione:

Affermazioni utili nei token di Amazon Cognito accesso

`client_id`

L'ID dell'applicazione client di un relying party dell'OIDC. Con l'ID client, Verified Permissions può verificare che la richiesta di autorizzazione provenga da un client autorizzato per il policy store. Nell'autorizzazione machine-to-machine (M2M), il sistema richiedente autorizza una richiesta con un segreto del cliente e fornisce l'ID e gli ambiti del client come prova dell'autorizzazione.

`scope`

Gli ambiti OAuth 2.0 che rappresentano i permessi di accesso del portatore del token.

cognito:groups

Appartenenze ai gruppi di un utente. In un modello di autorizzazione basato sul controllo degli accessi basato sui ruoli (RBAC), questa dichiarazione presenta i ruoli che è possibile valutare nelle politiche.

Reclami transitori

Affermazioni che non sono un'autorizzazione di accesso, ma vengono aggiunte in fase di esecuzione da un trigger Lambda di generazione precedente al token di un pool di utenti. Le affermazioni transitorie assomigliano alle affermazioni standard ma non rientrano nello standard, ad esempio o. `tenant` `department` La personalizzazione dei token di accesso aggiunge costi alla bolletta. AWS

Per ulteriori informazioni sulla struttura dei token di accesso dei pool di utenti di Amazon Cognito, consulta Using the access token nella Amazon Cognito Developer Guide.

Un token di Amazon Cognito accesso viene mappato su un oggetto di contesto quando viene passato a Verified Permissions. È possibile fare riferimento agli attributi del token di accesso utilizzando. `context.token.`*`attribute_name`* Il token di accesso di esempio seguente include sia le `client_id` `scope` attestazioni che.

```
{
    "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"
}
```

L'esempio seguente mostra come riflettere gli attributi del token di accesso di esempio nello schema di autorizzazioni verificate. Per ulteriori informazioni sulla modifica dello schema, consultaModifica degli schemi di archiviazione delle politiche.

```
{
   "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"
         }
```

```
      }
   }
}
```

Per un esempio di politica che verrà convalidata in base a questo schema, vediRiflette gli attributi dei token di accesso Amazon Cognito.

Notazione alternativa per Amazon Cognito affermazioni delimitate da due punti

Al momento del lancio di Verified Permissions, lo schema consigliato per le attestazioni dei Amazon Cognito token includeva `cognito:groups` e `custom:store` convertiva queste stringhe delimitate da due punti in modo che utilizzassero il carattere come delimitatore gerarchico. `.` Questo formato è chiamato notazione a punti. Ad esempio, un riferimento a `cognito:groups` became `principal.cognito.groups` nelle tue politiche. Sebbene sia possibile continuare a utilizzare questo formato, si consiglia di creare lo schema e le politiche utilizzando la notazione tra parentesi. In questo formato, un riferimento a `cognito:groups` diventa `principal["cognito:groups"]` nelle tue politiche. Gli schemi generati automaticamente per i token ID del pool di utenti dalla console Verified Permissions utilizzano la notazione tra parentesi.

Puoi continuare a utilizzare la notazione a punti negli schemi e nelle policy creati manualmente per le fonti di identità. Amazon Cognito Non puoi utilizzare la notazione a punti con `:` o qualsiasi altro carattere non alfanumerico nello schema o nelle politiche per nessun altro tipo di IdP OIDC.

Uno schema per la notazione a punti annida ogni istanza di un `:` carattere come elemento secondario della frase `cognito` o `custom` iniziale, come illustrato nell'esempio seguente:

```
"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
         }
      }
   }
}
```

Per un esempio di politica che verrà convalidata in base a questo schema e utilizzerà la notazione a punti, vedere. Utilizza la notazione a punti per fare riferimento agli attributi

Cose da sapere sulla mappatura degli schemi

La mappatura degli attributi differisce tra i tipi di token

Nell'autorizzazione del token di accesso, Verified Permissions mappa le rivendicazioni in base al contesto. Nell'autorizzazione tramite token ID, Verified Permissions associa le rivendicazioni agli attributi principali. Per i policy store creati nella console Verified Permissions, solo gli archivi di policy vuoti e di esempio non lasciano alcuna fonte di identità e richiedono di compilare lo schema con gli attributi del pool di utenti per l'autorizzazione del token ID. L'autorizzazione dei token di accesso si basa sul controllo degli accessi basato sui ruoli (RBAC) con attestazioni di appartenenza al gruppo e non associa automaticamente altre attestazioni allo schema del policy store.

Gli attributi di origine dell'identità non sono obbligatori

Quando crei una fonte di identità nella console Autorizzazioni verificate, nessun attributo viene contrassegnato come obbligatorio. In questo modo si evita che le attestazioni mancanti causino errori di convalida nelle richieste di autorizzazione. È possibile impostare gli attributi come obbligatori in base alle esigenze, ma devono essere presenti in tutte le richieste di autorizzazione.

RBAC non richiede attributi nello schema

Gli schemi per le fonti di identità dipendono dalle associazioni di entità che si creano quando si aggiunge la fonte di identità. Un'origine di identità associa un'attestazione a un tipo di entità utente e un'affermazione a un tipo di entità di gruppo. Queste mappature di entità sono il fulcro di una configurazione di origine dell'identità. Con queste informazioni minime, è possibile scrivere politiche che eseguano azioni di autorizzazione per utenti specifici e gruppi specifici di cui gli utenti potrebbero essere membri, in un modello di controllo degli accessi basato sul ruolo (RBAC). L'aggiunta di attestazioni di token allo schema estende l'ambito di autorizzazione del policy store. Gli attributi utente dei token ID contengono informazioni sugli utenti che possono contribuire all'autorizzazione del controllo degli accessi basato sugli attributi (ABAC). Gli attributi di contesto dei token di accesso contengono informazioni come gli ambiti OAuth 2.0 che possono fornire ulteriori informazioni sul controllo degli accessi fornite dal provider, ma richiedono ulteriori modifiche allo schema.

Le opzioni Configura con API Gateway e un provider di identità e Configurazione guidata nella console Autorizzazioni verificate assegnano le attestazioni del token ID allo schema. Questo non è il caso delle rivendicazioni relative ai token di accesso. Per aggiungere attestazioni di token di accesso non di gruppo allo schema, è necessario modificare lo schema in modalità JSON e aggiungere gli attributi CommonTypes. Per ulteriori informazioni, consulta Mappatura dei token di accesso.

Scegli un tipo di token

Il modo in cui il policy store funziona con la fonte di identità dipende da una decisione chiave nella configurazione dell'origine dell'identità: se elaborare gli ID o i token di accesso. Con un provider di Amazon Cognito identità, puoi scegliere il tipo di token quando crei un policy store collegato all'API. Quando crei un policy store collegato all'API, devi scegliere se configurare l'autorizzazione per l'ID o i token di accesso. Queste informazioni influiscono sugli attributi dello schema che Verified Permissions applica al tuo policy store e sulla sintassi dell'autorizzatore Lambda per la tua API. API Gateway Soprattutto se desideri trarre vantaggio dalla mappatura automatica delle dichiarazioni dei token ID agli attributi nella console Verified Permissions, decidi in anticipo il tipo di token che desideri elaborare prima di creare la fonte dell'identità. La modifica del tipo di token richiede uno sforzo significativo per rifattorizzare le politiche e lo schema. I seguenti argomenti descrivono l'uso degli ID e dei token di accesso con gli archivi delle politiche.

Cedar parser richiede parentesi per alcuni caratteri

Le politiche in genere fanno riferimento agli attributi dello schema in un formato simile. `principal.username` Nel caso della maggior parte dei caratteri non alfanumerici come `:`., o `/` che potrebbero apparire nei nomi delle rivendicazioni dei token, Verified Permissions non è in grado

di analizzare un valore di condizione come o. `principal.cognito:username` `context.ip-address` È invece necessario formattare queste condizioni con la notazione tra parentesi nel formato o, rispettivamente. `principal["cognito:username"]` `context["ip-address"]` Il carattere di sottolineatura _ è un carattere valido nei nomi delle rivendicazioni e rappresenta l'unica eccezione non alfanumerica a questo requisito.

Uno schema di esempio parziale per un attributo principale di questo tipo è simile al seguente:

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

Uno schema di esempio parziale per un attributo di contesto di questo tipo è simile al seguente:

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

```
            }
    }
    },
       "principalTypes": [
          "User"
       ]
    }
}
```

Per un esempio di politica che verrà convalidata rispetto a questo schema, vediUtilizza la notazione tra parentesi per fare riferimento agli attributi del token.

Convalida di clienti e destinatari per Amazon Cognito

Quando si aggiunge una fonte di identità a un policy store, Verified Permissions dispone di opzioni di configurazione che verificano che l'ID e i token di accesso vengano utilizzati come previsto. Questa convalida avviene durante l'elaborazione delle richieste API`IsAuthorizedWithToken`. `BatchIsAuthorizedWithToken` Il comportamento differisce tra ID e token di accesso Amazon Cognito e tra fonti di identità OIDC. Con i provider di pool di utenti di Amazon Cognito, Verified Permissions può convalidare l'ID client sia nell'ID che nei token di accesso. Con i provider OIDC, Verified Permissions può convalidare l'ID client nei token ID e il pubblico nei token di accesso.

Un ID client è un identificatore associato all'istanza del provider di identità utilizzata dall'applicazione, ad esempio. `1example23456789` Un pubblico è un percorso URL associato al relying party, o destinazione, previsto per il token di accesso, ad esempio. `https://mytoken.example.com` Quando si utilizzano i token di accesso, l'`aud`affermazione è sempre associata al pubblico.

Amazon Cognito I token ID hanno un `aud` claim che contiene l'ID client dell'app. I token di accesso hanno un `client_id` claim che contiene anche l'ID client dell'app.

Quando inserisci uno o più valori per la convalida dell'applicazione Client nella fonte della tua identità, Verified Permissions confronta questo elenco di client IDs dell'app con l'attestazione del token ID o l'`aud`attestazione del token di accesso. `client_id` Verified Permissions non convalida l'URL di un pubblico relying-party per le fonti di identità. Amazon Cognito

Autorizzazione lato client per JWTs

Potresti voler elaborare i token web JSON nella tua applicazione e passare le relative dichiarazioni a Verified Permissions senza utilizzare una fonte di identità del Policy Store. Puoi estrarre gli attributi della tua entità da un token Web JSON (JWT) e analizzarli in autorizzazioni verificate.

Questo esempio mostra come è possibile chiamare le autorizzazioni verificate da un'applicazione che utilizza un JWT.[1]

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

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

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

    return authResult;
```

```
}

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

  });

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

```
}
```

[1] Questo esempio di codice utilizza la aws-jwt-verifylibreria per la verifica JWTs della compatibilità con OIDC. IdPs

Utilizzo delle fonti di identità OIDC

Puoi anche configurare qualsiasi IdP OpenID Connect (OIDC) conforme come fonte di identità di un policy store. I provider OIDC sono simili ai pool di utenti di Amazon Cognito: JWTs producono come prodotto di autenticazione. Per aggiungere un provider OIDC, devi fornire un URL emittente

Una nuova fonte di identità OIDC richiede le seguenti informazioni:

- L'URL dell'emittente. Le autorizzazioni verificate devono essere in grado di rilevare un `.well-known/openid-configuration` endpoint in questo URL.
- Record CNAME che non includono wild card. Ad esempio, non `a.example.com` può essere mappato su. `*.example.net` Al contrario, non `*.example.com` può essere mappato su. `a.example.net`
- Il tipo di token che desideri utilizzare nelle richieste di autorizzazione. In questo caso, hai scelto Identity token.
- Il tipo di entità utente che desideri associare alla fonte della tua identità, ad esempio`MyCorp::User`.
- Il tipo di entità di gruppo che desideri associare alla tua fonte di identità, ad esempio`MyCorp::UserGroup`.
- Un esempio di token ID o una definizione delle attestazioni nel token ID.
- Il prefisso che desideri applicare all'entità IDs utente e di gruppo. Nella CLI e nell'API, puoi scegliere questo prefisso. Negli archivi di policy creati con l'opzione Configura con API Gateway e un provider di identità o l'opzione di configurazione guidata, Verified Permissions assegna un prefisso del nome dell'emittente meno`https://`, ad esempio. `MyCorp::User::"auth.example.com|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"`

Per ulteriori informazioni sull'utilizzo delle operazioni API per autorizzare le richieste provenienti da fonti OIDC, vedere. Operazioni API disponibili per l'autorizzazione

L'esempio seguente mostra come è possibile creare una politica che consenta l'accesso ai report di fine anno ai dipendenti del reparto contabilità, abbiano una classificazione riservata e non lavorino in

un ufficio secondario. Verified Permissions ricava questi attributi dalle attestazioni contenute nel token ID del preside.

Si noti che quando si fa riferimento a un gruppo nel principale, è necessario utilizzare l'`in`operatore affinché la politica venga valutata correttamente.

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

Argomenti

- Creazione di fonti di identità OIDC di Amazon Verified Permissions
- Modifica delle fonti di identità OIDC di Amazon Verified Permissions
- Mappatura dei token OIDC sullo schema
- Convalida di clienti e destinatari per i provider OIDC

Creazione di fonti di identità OIDC di Amazon Verified Permissions

La procedura seguente aggiunge un'origine di identità a un archivio di politiche esistente.

È inoltre possibile creare un'origine di identità quando si crea un nuovo archivio di politiche nella console Autorizzazioni verificate. In questo processo, puoi importare automaticamente le attestazioni contenute nei token di origine dell'identità negli attributi dell'entità. Scegli l'opzione Configurazione guidata o Configura con API Gateway e un provider di identità. Queste opzioni creano anche politiche iniziali.

> **Note**
>
> Le fonti di identità non sono disponibili nel riquadro di navigazione a sinistra fino a quando non è stato creato un archivio delle politiche. Le fonti di identità create sono associate al policy store corrente.

Puoi omettere il tipo di entità principale quando crei una fonte di identità con AWS CLI o create-identity-sourceCreateIdentitySourcenell'API Verified Permissions. Tuttavia, un tipo di entità vuoto crea una fonte di identità con un tipo di entità di`AWS::Cognito`. Questo nome di entità non è compatibile con lo schema dell'archivio delle politiche. Per integrare Amazon Cognito le identità con lo schema dell'archivio delle politiche, è necessario impostare il tipo di entità principale su un'entità dell'archivio delle politiche supportata.

Console di gestione AWS

Per creare una fonte di identità OpenID Connect (OIDC)

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Identity sources.
3. Scegli Crea fonte di identità.
4. Scegli un provider OIDC esterno.
5. In URL dell'emittente, inserisci l'URL dell'emittente OIDC. Questo è l'endpoint del servizio che fornisce, ad esempio, il server di autorizzazione, le chiavi di firma e altre informazioni sul provider. `https://auth.example.com` L'URL dell'emittente deve ospitare un documento di rilevamento OIDC presso. `/.well-known/openid-configuration`
6. In Tipo di token, scegli il tipo di OIDC JWT che desideri che la tua applicazione invii per l'autorizzazione. Per ulteriori informazioni, consulta Mappatura dei token OIDC sullo schema.
7. In Mappa le rivendicazioni dei token alle entità dello schema, scegli un'entità utente e un'attestazione utente per l'origine dell'identità. L'entità Utente è un'entità nel tuo archivio delle politiche a cui desideri fare riferimento agli utenti del tuo provider OIDC. L'attestazione Utente è un reclamo, in genere`sub`, derivante dal tuo ID o token di accesso che contiene l'identificatore univoco dell'entità da valutare. Le identità dell'IdP OIDC connesso verranno mappate sul tipo principale selezionato.
8. (Facoltativo) In Mappa le rivendicazioni dei token alle entità dello schema, scegli un'entità di gruppo e un'attestazione di gruppo come origine dell'identità. L'entità Gruppo è l'entità principale dell'entità Utente. Le rivendicazioni di gruppo vengono mappate su questa entità. L'attestazione di gruppo è in genere `groups` un'affermazione derivante dall'ID o dal token di accesso che contiene una stringa, JSON o una stringa di nomi di gruppi di utenti delimitata da spazi per l'entità da valutare. Le identità dell'IdP OIDC connesso verranno mappate sul tipo principale selezionato.
9. In fase di convalida: facoltativo, inserisci il cliente IDs o il pubblico URLs che desideri che l'archivio delle politiche accetti nelle eventuali richieste di autorizzazione.

10. Scegli Crea fonte di identità.
11. (Facoltativo) Se il vostro policy store dispone di uno schema, prima di poter fare riferimento agli attributi che estraete dall'identità o dai token di accesso nelle vostre policy Cedar, dovete aggiornare lo schema per rendere Cedar consapevole del tipo di principale creato dalla vostra fonte di identità. Tale aggiunta allo schema deve includere gli attributi a cui desiderate fare riferimento nelle vostre politiche Cedar. Per ulteriori informazioni sulla mappatura degli attributi del token OIDC agli attributi principali di Cedar, vedere. Mappatura dei token OIDC sullo schema
12. Crea politiche che utilizzano le informazioni dei token per prendere decisioni di autorizzazione. Per ulteriori informazioni, consulta Creazione di politiche statiche di Amazon Verified Permissions.

Ora che hai creato una fonte di identità, aggiornato lo schema e creato le politiche, consenti `IsAuthorizedWithToken` alle Autorizzazioni Verificate di prendere decisioni di autorizzazione. Per ulteriori informazioni, consulta IsAuthorizedWithTokenla guida di riferimento dell'API Amazon Verified Permissions.

AWS CLI

Per creare una fonte di identità OIDC

È possibile creare una fonte di identità utilizzando l'CreateIdentitySourceoperazione. L'esempio seguente crea un'origine di identità in grado di accedere alle identità autenticate da un provider di identità (IdP) OIDC.

1. Crea un `config.txt` file che contenga i seguenti dettagli di un IdP OIDC da utilizzare con il `--configuration` parametro del comando. `create-identity-source`

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

```
                "groupEntityType": "MyCorp::UserGroup"
            }
        }
}
```

2. Esegui il comando seguente per creare una fonte di identità OIDC.

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

3. (Facoltativo) Se il vostro policy store dispone di uno schema, prima di poter fare riferimento agli attributi che estraete dai token di identità o di accesso nelle vostre policy Cedar, dovete aggiornare lo schema per rendere Cedar consapevole del tipo di principale creato dalla vostra fonte di identità. Tale aggiunta allo schema deve includere gli attributi a cui desiderate fare riferimento nelle vostre politiche Cedar. Per ulteriori informazioni sulla mappatura degli attributi del token OIDC agli attributi principali di Cedar, vedere. Mappatura dei token OIDC sullo schema
4. Crea politiche che utilizzano le informazioni dei token per prendere decisioni di autorizzazione. Per ulteriori informazioni, consulta Creazione di politiche statiche di Amazon Verified Permissions.

Ora che hai creato una fonte di identità, aggiornato lo schema e creato le politiche, consenti `IsAuthorizedWithToken` alle Autorizzazioni Verificate di prendere decisioni di autorizzazione. Per ulteriori informazioni, consulta IsAuthorizedWithTokenla guida di riferimento dell'API Amazon Verified Permissions.

Modifica delle fonti di identità OIDC di Amazon Verified Permissions

Puoi modificare alcuni parametri della tua fonte di identità dopo averla creata. Non puoi cambiare il tipo di fonte di identità, devi eliminare l'origine dell'identità e crearne una nuova da cui passare Amazon Cognito a OIDC o OIDC. Amazon Cognito Se lo schema del policy store corrisponde agli

attributi della fonte di identità, tieni presente che devi aggiornare lo schema separatamente per riflettere le modifiche apportate alla tua origine di identità.

Console di gestione AWS

Per aggiornare una fonte di identità OIDC

1. Apri la console delle autorizzazioni verificate. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Identity sources.
3. Scegli l'ID della fonte di identità da modificare.
4. Scegli Modifica.
5. Nei dettagli del provider OIDC, modifica l'URL dell'emittente in base alle esigenze.
6. In Map token claim to schema, modifica le associazioni tra le attestazioni utente e di gruppo e i tipi di entità del Policy Store, se necessario. Dopo aver modificato i tipi di entità, è necessario aggiornare le politiche e gli attributi dello schema per applicarli ai nuovi tipi di entità.
7. Nella convalida dell'audience, aggiungi o rimuovi i valori di audience che desideri applicare.
8. Scegli Save changes (Salva modifiche).

Puoi eliminare una fonte di identità scegliendo il pulsante di opzione accanto a una fonte di identità e quindi scegliendo Elimina fonte di identità. Digita `delete` nella casella di testo, quindi scegli Elimina fonte di identità per confermare l'eliminazione della fonte di identità.

AWS CLI

Per aggiornare una fonte di identità OIDC

È possibile aggiornare una fonte di identità utilizzando l'UpdateIdentitySourceoperazione. L'esempio seguente aggiorna l'origine di identità specificata per utilizzare un provider OIDC diverso.

1. Crea un `config.txt` file che contenga i seguenti dettagli di un IdP OIDC da utilizzare con il `--configuration` parametro del comando. `update-identity-source`

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

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

2. Esegui il comando seguente per aggiornare una fonte di identità OIDC.

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

> **Note**
>
> Se si modifica il tipo principale per l'origine dell'identità, è necessario aggiornare lo schema in modo che rifletta correttamente il tipo principale aggiornato.

Mappatura dei token OIDC sullo schema

Potresti scoprire di voler aggiungere una fonte di identità a un archivio delle politiche e mappare le attestazioni, o token, del provider allo schema del tuo archivio delle politiche. È possibile automatizzare questo processo utilizzando la configurazione guidata per creare il proprio Policy Store con una fonte di identità o aggiornare lo schema manualmente dopo la creazione del Policy Store. Dopo aver mappato i token allo schema, è possibile creare policy che vi fanno riferimento.

Questa sezione della guida per l'utente contiene le seguenti informazioni:

- Quando è possibile compilare automaticamente gli attributi in uno schema di policy store
- Come creare manualmente uno schema per una fonte di identità

Gli archivi di policy collegati alle API e gli archivi di policy con una fonte di identità creati tramite la configurazione guidata non richiedono la mappatura manuale degli attributi del token di identità (ID) allo schema. Puoi fornire autorizzazioni verificate con gli attributi del tuo pool di utenti e creare uno schema popolato con attributi utente. Nell'autorizzazione con token ID, Verified Permissions associa le rivendicazioni agli attributi di un'entità principale.

Per utilizzare un provider di identità (IdP) OIDC come fonte di identità nel tuo archivio di policy per le autorizzazioni verificate, devi avere gli attributi del provider nello schema. Lo schema è fisso e deve corrispondere alle entità create dai token del provider o alle richieste API. IsAuthorizedWithTokenBatchIsAuthorizedWithToken Se hai creato il tuo archivio delle politiche in modo da compilare automaticamente lo schema con le informazioni del provider in un token ID, sei pronto per scrivere le policy. Se crei un policy store senza uno schema per la tua origine di identità, devi aggiungere gli attributi del provider allo schema che corrispondono alle entità create utilizzando le richieste API. Quindi puoi scrivere politiche utilizzando gli attributi del token del provider.

Argomenti

- Mappatura dei token ID sullo schema
- Mappatura dei token di accesso
- Cose da sapere sulla mappatura degli schemi

Mappatura dei token ID sullo schema

Verified Permissions elabora le dichiarazioni relative ai token ID come attributi dell'utente: nomi e titoli, appartenenza al gruppo, informazioni di contatto. I token ID sono molto utili in un modello di autorizzazione ABAC (Attribute-Based Access Control). Se desideri che le autorizzazioni verificate analizzino l'accesso alle risorse in base a chi effettua la richiesta, scegli i token ID come fonte della tua identità.

Lavorare con i token ID di un provider OIDC è molto simile a lavorare con i token ID. Amazon Cognito La differenza sta nelle affermazioni. Il tuo IdP potrebbe presentare attributi OIDC standard o avere uno schema personalizzato. Quando crei un nuovo archivio di politiche nella console Verified Permissions, puoi aggiungere una fonte di identità OIDC con un token ID di esempio oppure puoi mappare manualmente le attestazioni dei token agli attributi utente. Poiché Verified Permissions non conosce lo schema degli attributi del tuo IdP, devi fornire queste informazioni.

Per ulteriori informazioni, consulta Creazione di archivi di policy per le autorizzazioni verificate.

Di seguito è riportato uno schema di esempio per un policy store con una fonte di identità 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"
         }
      }
   }
}
```

Per un esempio di politica che verrà convalidata in base a questo schema, vedere. Riflette gli attributi del token ID OIDC

Mappatura dei token di accesso

Verified Permissions elabora le dichiarazioni dei token di accesso diverse da quelle dichiarate dai gruppi come attributi dell'azione o attributi di contesto. Oltre all'appartenenza al gruppo, i token di accesso del tuo IdP potrebbero contenere informazioni sull'accesso alle API. I token di accesso sono utili nei modelli di autorizzazione che utilizzano il controllo degli accessi basato sui ruoli (RBAC). I modelli di autorizzazione che si basano su richieste di token di accesso diverse dall'appartenenza al gruppo richiedono uno sforzo aggiuntivo nella configurazione dello schema.

La maggior parte dei token di accesso di fornitori OIDC esterni si allinea strettamente ai token di accesso. Amazon Cognito Un token di accesso OIDC viene mappato su un oggetto di contesto

quando viene passato a Verified Permissions. È possibile fare riferimento agli attributi del token di accesso utilizzando. `context.token.`*`attribute_name`* Il seguente token di accesso OIDC include esempi di attestazioni di base.

```
{
    "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"
}
```

L'esempio seguente mostra come riflettere gli attributi del token di accesso di esempio nello schema Verified Permissions. Per ulteriori informazioni sulla modifica dello schema, consultaModifica degli schemi di archiviazione delle politiche.

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

Per un esempio di politica che verrà convalidata rispetto a questo schema, vediRiflette gli attributi del token di accesso OIDC.

Cose da sapere sulla mappatura degli schemi

La mappatura degli attributi differisce tra i tipi di token

Nell'autorizzazione del token di accesso, Verified Permissions mappa le rivendicazioni in base al contesto. Nell'autorizzazione tramite token ID, Verified Permissions associa le rivendicazioni agli attributi principali. Per i policy store creati nella console Verified Permissions, solo gli archivi di policy vuoti e di esempio non lasciano alcuna fonte di identità e richiedono di compilare lo schema con gli attributi del pool di utenti per l'autorizzazione del token ID. L'autorizzazione dei token di accesso si basa sul controllo degli accessi basato sui ruoli (RBAC) con attestazioni di appartenenza al gruppo e non associa automaticamente altre attestazioni allo schema del policy store.

Gli attributi di origine dell'identità non sono obbligatori

Quando crei una fonte di identità nella console Autorizzazioni verificate, nessun attributo viene contrassegnato come obbligatorio. In questo modo si evita che le attestazioni mancanti causino errori di convalida nelle richieste di autorizzazione. È possibile impostare gli attributi come obbligatori in base alle esigenze, ma devono essere presenti in tutte le richieste di autorizzazione.

RBAC non richiede attributi nello schema

Gli schemi per le fonti di identità dipendono dalle associazioni di entità che si creano quando si aggiunge la fonte di identità. Un'origine di identità associa un'attestazione a un tipo di entità utente e un'affermazione a un tipo di entità di gruppo. Queste mappature di entità sono il fulcro di una configurazione di origine dell'identità. Con queste informazioni minime, è possibile scrivere politiche che eseguano azioni di autorizzazione per utenti specifici e gruppi specifici di cui gli utenti potrebbero essere membri, in un modello di controllo degli accessi basato sul ruolo (RBAC). L'aggiunta di attestazioni di token allo schema estende l'ambito di autorizzazione del policy store. Gli attributi utente dei token ID contengono informazioni sugli utenti che possono contribuire all'autorizzazione del controllo degli accessi basato sugli attributi (ABAC). Gli attributi di contesto dei token di accesso contengono informazioni come gli ambiti OAuth 2.0 che possono fornire ulteriori informazioni sul controllo degli accessi fornite dal provider, ma richiedono ulteriori modifiche allo schema.

Le opzioni Configura con API Gateway e un provider di identità e Configurazione guidata nella console Autorizzazioni verificate assegnano le attestazioni del token ID allo schema. Questo non è il caso delle rivendicazioni relative ai token di accesso. Per aggiungere attestazioni di token di accesso non di gruppo allo schema, è necessario modificare lo schema in modalità JSON e aggiungere gli attributi CommonTypes. Per ulteriori informazioni, consulta Mappatura dei token di accesso.

L'affermazione dei gruppi OIDC supporta più formati

Quando aggiungi un provider OIDC, puoi scegliere il nome della dichiarazione di gruppo in ID o i token di accesso che desideri associare all'appartenenza al gruppo di un utente nel tuo archivio delle politiche. Le autorizzazioni verificate riconoscono le dichiarazioni dei gruppi nei seguenti formati:

1. Stringa senza spazi: `"groups": "MyGroup"`
2. Elenco delimitato da spazi:. `"groups": "MyGroup1 MyGroup2 MyGroup3"` Ogni stringa è un gruppo.
3. Elenco JSON (delimitato da virgole): `"groups": ["MyGroup1", "MyGroup2", "MyGroup3"]`

> **Note**
>
> Verified Permissions interpreta ogni stringa contenuta in una dichiarazione di gruppi separati da spazi come un gruppo separato. Per interpretare il nome di un gruppo con un carattere di spazio come un singolo gruppo, sostituisci o rimuovi lo spazio nell'attestazione. Ad esempio, formatta un gruppo denominato `My Group` come`MyGroup`.

Scegli un tipo di token

Il modo in cui il policy store funziona con la fonte di identità dipende da una decisione chiave nella configurazione dell'origine dell'identità: se elaborare gli ID o i token di accesso. Con un provider OIDC, è necessario scegliere un tipo di token quando si aggiunge la fonte di identità. Puoi scegliere ID o token di accesso e la tua scelta esclude che il tipo di token non scelto venga elaborato nel tuo archivio delle politiche. Soprattutto se desideri trarre vantaggio dalla mappatura automatica delle rivendicazioni dei token ID agli attributi nella console Verified Permissions, decidi in anticipo il tipo di token che desideri elaborare prima di creare la tua fonte di identità. La modifica del tipo di token richiede uno sforzo significativo per rifattorizzare le politiche e lo schema. I seguenti argomenti descrivono l'uso degli ID e dei token di accesso con gli archivi delle politiche.

Cedar parser richiede parentesi per alcuni caratteri

Le politiche in genere fanno riferimento agli attributi dello schema in un formato simile. `principal.username` Nel caso della maggior parte dei caratteri non alfanumerici come `:.`, o `/` che potrebbero apparire nei nomi delle rivendicazioni dei token, Verified Permissions non è in grado di analizzare un valore di condizione come o. `principal.cognito:username` `context.ip-address` È invece necessario formattare queste condizioni con la notazione tra parentesi nel formato o, rispettivamente. `principal["cognito:username"]` `context["ip-address"]` Il carattere di sottolineatura `_` è un carattere valido nei nomi delle rivendicazioni e rappresenta l'unica eccezione non alfanumerica a questo requisito.

Uno schema di esempio parziale per un attributo principale di questo tipo è simile al seguente:

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

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

Uno schema di esempio parziale per un attributo di contesto di questo tipo è simile al seguente:

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

Per un esempio di politica che verrà convalidata rispetto a questo schema, vediUtilizza la notazione tra parentesi per fare riferimento agli attributi del token.

Convalida di clienti e destinatari per i provider OIDC

Quando si aggiunge una fonte di identità a un policy store, Verified Permissions dispone di opzioni di configurazione che verificano che l'ID e i token di accesso vengano utilizzati come previsto.

Questa convalida avviene durante l'elaborazione delle richieste API`IsAuthorizedWithToken`. `BatchIsAuthorizedWithToken` Il comportamento differisce tra ID e token di accesso Amazon Cognito e tra fonti di identità OIDC. Con i provider di pool di utenti di Amazon Cognito, Verified Permissions può convalidare l'ID client sia nell'ID che nei token di accesso. Con i provider OIDC, Verified Permissions può convalidare l'ID client nei token ID e il pubblico nei token di accesso.

Un ID client è un identificatore associato all'istanza del provider di identità utilizzata dall'applicazione, ad esempio. `1example23456789` Un pubblico è un percorso URL associato al relying party, o destinazione, previsto per il token di accesso, ad esempio. `https://mytoken.example.com` Quando si utilizzano i token di accesso, l'`aud`affermazione è sempre associata al pubblico.

I token ID OIDC hanno un `aud` claim che contiene client IDs, ad esempio. `1example23456789`

I token di accesso OIDC hanno un'`aud`attestazione che contiene l'URL del pubblico per il token, ad esempio, e un'`client_id`attestazione che contiene client`https://myapplication.example.com`, ad esempio. IDs `1example23456789`

Quando configuri il tuo policy store, inserisci uno o più valori per la convalida dell'audience che il tuo policy store utilizzerà per convalidare il pubblico di un token.

- Token ID: Verified Permissions convalida l'ID client verificando che almeno un membro del client IDs nell'`aud`attestazione corrisponda a un valore di convalida del pubblico.
- Token di accesso: le autorizzazioni verificate convalidano il pubblico verificando che l'URL nell'attestazione corrisponda a un valore di convalida del `aud` pubblico. Se non esiste alcuna `aud` affermazione, il pubblico può essere convalidato utilizzando le attestazioni o. `cid` `client_id` Rivolgiti al tuo provider di identità per conoscere la dichiarazione e il formato corretti relativi al pubblico.

Autorizzazione lato client per JWTs

Potresti voler elaborare i token web JSON nella tua applicazione e passare le relative dichiarazioni a Verified Permissions senza utilizzare una fonte di identità del Policy Store. Puoi estrarre gli attributi della tua entità da un token Web JSON (JWT) e analizzarli in autorizzazioni verificate.

Questo esempio mostra come è possibile chiamare le autorizzazioni verificate da un'applicazione che utilizza un JWT.[1]

```
async function authorizeUsingJwtToken(jwtToken) {
```

```
    const payload = await verifier.verify(jwtToken);

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

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

    return authResult;

}

function getUserEntitiesFromToken(payload) {
  let attributes = {};
  let claimsNotPassedInEntities = ['aud', 'sub', 'exp', 'jti', 'iss'];
  Object.entries(payload).forEach(([key, value]) => {
```

```
    if (claimsNotPassedInEntities.includes(key)) {
        return;
    }
    if (Array.isArray(value)) {
      var 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 === 'boolean') {
        attributes[key] = {
            boolean: value,
       }
    }

  });

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

[1] Questo esempio di codice utilizza la aws-jwt-verifylibreria per la verifica JWTs della compatibilità con OIDC. IdPs

Integrazioni per Amazon Verified Permissions

Le integrazioni di Amazon Verified Permissions ti aiutano a implementare autorizzazioni granulari nelle tue applicazioni, riducendo al minimo il codice e seguendo le best practice specifiche del framework. Queste integrazioni forniscono componenti e utilità middleware che collegano perfettamente l'applicazione con le autorizzazioni verificate.

Con le integrazioni, puoi:

- Implementa l'autorizzazione in pochi minuti
- Segui modelli e convenzioni specifici del framework
- Riduci il sovraccarico di manutenzione
- Ridurre al minimo i potenziali errori di implementazione della sicurezza
- Concentrati sulla logica aziendale piuttosto che sul codice di autorizzazione

Quando vengono aggiunte all'applicazione, le integrazioni eseguono le seguenti operazioni:

1. Intercetta le richieste in arrivo tramite middleware specifico del framework
2. Estrai il contesto di autorizzazione pertinente dalle richieste
3. Determina le decisioni di autorizzazione utilizzando le autorizzazioni verificate
4. Applica il controllo degli accessi in base ai risultati delle autorizzazioni

Verified Permissions attualmente supporta i seguenti framework:

- Express.js per applicazioni Node.js

Integrazione di Express con le autorizzazioni verificate di Amazon

L'integrazione Verified Permissions Express fornisce un approccio basato sul middleware per l'implementazione dell'autorizzazione nelle applicazioni Express.js. Con questa integrazione, puoi proteggere gli endpoint delle API utilizzando politiche di autorizzazione granulari senza modificare i gestori di route esistenti. L'integrazione gestisce automaticamente i controlli di autorizzazione intercettando le richieste, valutandole rispetto alle policy definite e garantendo che solo gli utenti autorizzati possano accedere alle risorse protette.

Questo argomento illustra la configurazione dell'integrazione Express, dalla creazione di un archivio di politiche all'implementazione e al test del middleware di autorizzazione. Seguendo questi passaggi, è possibile aggiungere solidi controlli di autorizzazione all'applicazione Express con modifiche minime al codice.

In questo argomento GitHub si fa riferimento ai seguenti repository:

- cedar-policy/ authorization-for-expressjs - Il middleware di autorizzazione Cedar per Express.js
- verifiedpermissions/ - I client di autorizzazione Verified Permissions authorization-clients-js per JavaScript
- verifiedpermissions/examples/express-petstore - Esempio di implementazione utilizzando il middleware Express.js

Prerequisiti

Prima di implementare l'integrazione con Express, assicurati di avere:

- Un AWS account con accesso alle autorizzazioni verificate
- Node.js e npm installati
- Un'applicazione Express.js
- Un provider di identità OpenID Connect (OIDC) (ad esempio) Amazon Cognito
- AWS CLIconfigurato con le autorizzazioni appropriate

Configurazione dell'integrazione

Fase 1: Creare un archivio delle politiche

Creare un archivio delle politiche utilizzando AWS CLI:

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

> **Note**
>
> Salva l'ID del policy store restituito nella risposta per utilizzarlo nei passaggi successivi.

Passaggio 2: installare le dipendenze

Installa i pacchetti richiesti nella tua applicazione Express:

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

Configurazione dell'autorizzazione

Passaggio 1: generare e caricare lo schema Cedar

Uno schema definisce il modello di autorizzazione per un'applicazione, inclusi i tipi di entità nell'applicazione e le azioni che gli utenti possono intraprendere. Si consiglia di definire uno spazio dei nomi per lo schema. In questo esempio viene utilizzato `YourNamespace`. Alleghi lo schema agli archivi dei criteri di autorizzazione verificata e, quando i criteri vengono aggiunti o modificati, il servizio li convalida automaticamente rispetto allo schema.

Il `@cedar-policy/authorization-for-expressjs` pacchetto può analizzare le specifiche OpenAPI dell'applicazione e generare uno schema Cedar. In particolare, l'oggetto paths è richiesto nelle vostre specifiche.

Se non disponi di una specifica OpenAPI, puoi seguire le istruzioni rapide del express-openapi-generatorpacchetto per generare una specifica OpenAPI.

Genera uno schema dalla tua specifica OpenAPI:

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

Quindi, formatta lo schema Cedar da utilizzare con. AWS CLI Per ulteriori informazioni sul formato specifico richiesto, vedereSchema del Policy Store. Se hai bisogno di aiuto per formattare lo schema, c'è uno script chiamato `prepare-cedar-schema.sh` nel repository GitHubverifiedpermissions/examples. Di seguito è riportato un esempio di chiamata a quello script che restituisce lo schema formattato Verified Permissions nel file. `v2.cedarschema.forAVP.json`

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

Carica lo schema formattato nel tuo policy store, sostituendolo `policy-store-id` con il tuo ID del policy store:

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

Fase 2: Creare politiche di autorizzazione

Se non è configurata alcuna politica, Cedar nega tutte le richieste di autorizzazione. L'integrazione del framework Express aiuta ad avviare questo processo generando politiche di esempio basate sullo schema generato in precedenza.

Quando si utilizza questa integrazione nelle applicazioni di produzione, si consiglia di creare nuove politiche utilizzando gli strumenti Infrastructure as a code (IaaC). Per ulteriori informazioni, consulta Lavorare con AWS CloudFormation.

Genera esempi di policy Cedar:

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

Questo genererà politiche di esempio nella `/policies` directory. È quindi possibile personalizzare queste politiche in base ai casi d'uso. Esempio:

```
// Defines permitted administrator user group actions
permit (
    principal in YourNamespace::UserGroup::"<userPoolId>|administrator",
    action,
    resource
);

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

Formatta le politiche da utilizzare con AWS CLI. Per ulteriori informazioni sul formato richiesto, consulta create-policy nel AWS CLI riferimento. Se hai bisogno di aiuto per formattare le politiche, c'è uno script chiamato `convert_cedar_policies.sh` nel repository verifiedpermissions/examples. GitHub Quella che segue è una chiamata a quello script:

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

Carica le politiche formattate in Autorizzazioni verificate, sostituendole `policy_1.json` con il percorso e il nome del file delle politiche e `policy-store-id` con l'ID del tuo archivio delle politiche:

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

Passaggio 3: Connect un provider di identità

Per impostazione predefinita, il middleware Verified Permissions Authorizer legge un JSON Web Token (JWT) fornito nell'intestazione di autorizzazione della richiesta API per ottenere informazioni sull'utente. Le autorizzazioni verificate possono convalidare il token oltre a eseguire la valutazione della politica di autorizzazione.

Crea un file di configurazione dell'origine dell'identità denominato `identity-source-configuration.txt` come segue con il tuo `userPoolArn` e: `clientId`

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

Crea l'origine dell'identità eseguendo il AWS CLI comando seguente, sostituendolo `policy-store-id` con il tuo Policy Store ID:

```
aws verifiedpermissions create-identity-source \
```

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

Implementazione del middleware di autorizzazione

Aggiorna l'applicazione Express per includere il middleware di autorizzazione. In questo esempio utilizziamo token di identità, ma puoi anche utilizzare token di accesso. Per ulteriori informazioni, vedere authorization-for-expressjssu. GitHub

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

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

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

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

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

Test dell'integrazione

Puoi testare l'implementazione dell'autorizzazione effettuando richieste agli endpoint API con diversi token utente. Il middleware di autorizzazione valuterà automaticamente ogni richiesta rispetto alle politiche definite.

Ad esempio, se hai impostato diversi gruppi di utenti con autorizzazioni diverse:

- Amministratori: accesso completo a tutte le risorse e le funzioni di gestione
- Dipendenti: possono visualizzare, creare e aggiornare le risorse
- Clienti: possono solo visualizzare le risorse

È possibile verificare che le politiche di autorizzazione funzionino come previsto accedendo con utenti diversi e tentando varie operazioni. Nel terminale dell'applicazione Express, è possibile visualizzare l'output del registro che fornisce ulteriori dettagli sulle decisioni di autorizzazione.

Risoluzione dei problemi

In caso di errori di autorizzazione, prova quanto segue:

- Verifica che il Policy Store ID sia corretto
- Assicurati che la tua fonte di identità sia configurata correttamente
- Verifica che le tue politiche siano formattate correttamente
- Verifica che i tuoi token JWT siano validi

Fasi successive

Dopo aver implementato l'integrazione di base, considera:

- Implementazione di mappatori personalizzati per scenari di autorizzazione specifici
- Configurazione del monitoraggio e della registrazione per le decisioni di autorizzazione
- Creazione di politiche aggiuntive per diversi ruoli utente

Implementazione dell'autorizzazione in Amazon Verified Permissions

Dopo aver creato l'archivio delle politiche, le politiche, i modelli, lo schema e il modello di autorizzazione, sei pronto per iniziare ad autorizzare le richieste utilizzando Amazon Verified Permissions. Per implementare l'autorizzazione Verified Permissions, devi combinare la configurazione delle politiche di autorizzazione AWS con l'integrazione in un'applicazione. Per integrare Verified Permissions con la tua applicazione, aggiungi un AWS SDK e implementa i metodi che richiamano l'API Verified Permissions e generano decisioni di autorizzazione in base al tuo archivio di policy.

L'autorizzazione con autorizzazioni verificate è utile per le autorizzazioni UX e le autorizzazioni API nelle applicazioni.

Autorizzazioni UX

Controlla l'accesso degli utenti alla UX della tua applicazione. Puoi consentire a un utente di visualizzare solo i moduli, i pulsanti, la grafica e le altre risorse esatte a cui deve accedere. Ad esempio, quando un utente effettua l'accesso, potresti voler determinare se il pulsante «Trasferisci fondi» è visibile nel suo account. Puoi anche controllare le azioni che un utente può intraprendere. Ad esempio, nella stessa app bancaria potresti voler determinare se il tuo utente è autorizzato a modificare la categoria di una transazione.

Autorizzazioni API

Controlla l'accesso degli utenti ai dati. Le applicazioni fanno spesso parte di un sistema distribuito e importano informazioni dall'esterno APIs. Nell'esempio dell'app bancaria in cui Verified Permissions ha consentito la visualizzazione del pulsante «Trasferisci fondi», è necessario prendere una decisione di autorizzazione più complessa quando l'utente avvia un trasferimento. Verified Permissions può autorizzare la richiesta API che elenca gli account di destinazione idonei al trasferimento, quindi la richiesta di inoltrare il trasferimento all'altro account.

Gli esempi che illustrano questo contenuto provengono da un esempio di policy store. A seguire, create il policy store DigitalPetStoredi esempio nel vostro ambiente di test.

Per un'applicazione di esempio end-to-end che implementa le autorizzazioni UX utilizzando l'autorizzazione in batch, consulta Use Amazon Verified Permissions per un'autorizzazione granulare su larga scala sul Security Blog.AWS

Argomenti

Operazioni API disponibili per l'autorizzazione

L'API Verified Permissions prevede le seguenti operazioni di autorizzazione.

IsAuthorized

L'operazione `IsAuthorized` API è il punto di accesso alle richieste di autorizzazione con autorizzazioni verificate. È necessario inviare gli elementi principali, di azione, di risorsa, di contesto ed entità. Verified Permissions valuta la richiesta rispetto a tutte le politiche dell'archivio delle politiche richiesto che si applicano alle entità incluse nella richiesta.

IsAuthorizedWithToken

L'`IsAuthorizedWithToken`operazione genera una richiesta di autorizzazione dai dati utente nei token web JSON (). JWTs Verified Permissions funziona direttamente con i provider OIDC, ad esempio Amazon Cognito come fonte di identità nell'archivio delle politiche. Verified Permissions compila tutti gli attributi relativi all'indirizzo principale della richiesta utilizzando le attestazioni contenute nell'ID degli utenti o nei token di accesso. È possibile autorizzare azioni e risorse dagli attributi degli utenti o dall'appartenenza a gruppi in una fonte di identità.

Non è possibile includere informazioni sui tipi principali di gruppi o utenti in una `IsAuthorizedWithToken` richiesta. È necessario inserire tutti i dati principali nel JWT fornito.

BatchIsAuthorized

L'`BatchIsAuthorized`operazione elabora più decisioni di autorizzazione per un singolo principale o risorsa in un'unica richiesta API. Questa operazione raggruppa le richieste in un'unica operazione batch che riduce al minimo l'utilizzo delle quote e restituisce le decisioni di autorizzazione per ciascuna delle 30 azioni annidate complesse. Con l'autorizzazione in batch per una singola risorsa, è possibile filtrare le azioni che un utente può eseguire su una risorsa. Con l'autorizzazione in batch per un singolo principale, puoi filtrare in base alle risorse su cui un utente può intervenire.

BatchIsAuthorizedWithToken

L'`BatchIsAuthorizedWithToken`operazione elabora più decisioni di autorizzazione per un singolo principale in un'unica richiesta API. Il principale viene fornito dalla fonte di identità del Policy Store in un ID o token di accesso. Questa operazione raggruppa le richieste in un'unica operazione batch che riduce al minimo l'utilizzo delle quote e restituisce le decisioni di autorizzazione per ciascuna delle 30 richieste di azioni e risorse. Nelle vostre politiche, potete autorizzare il loro accesso dai loro attributi o dalla loro appartenenza al gruppo in una directory di utenti.

Ad esempio`IsAuthorizedWithToken`, non è possibile includere informazioni sui tipi principali di gruppi o utenti in una `BatchIsAuthorizedWithToken` richiesta. È necessario inserire tutti i dati principali nel JWT fornito.

Verifica del tuo modello di autorizzazione

Per comprendere l'effetto della decisione di autorizzazione di Amazon Verified Permissions quando distribuisci la tua applicazione, puoi valutare le tue politiche man mano che le sviluppi con Utilizzo del banco di prova Amazon Verified Permissions e con le richieste API REST HTTPS alle autorizzazioni verificate. Il banco di prova è uno strumento Console di gestione AWS per valutare le richieste e le risposte di autorizzazione nel tuo archivio di policy.

L'API REST di Verified Permissions è il passo successivo dello sviluppo, che prevede il passaggio da una comprensione concettuale alla progettazione di applicazioni. L'API Verified Permissions accetta richieste di autorizzazione con IsAuthorizede BatchIsAuthorizedcome richieste AWS API firmate agli endpoint di servizio regionali. IsAuthorizedWithToken Per testare il tuo modello di autorizzazione, puoi generare richieste con qualsiasi client API e verificare che le tue politiche restituiscano le decisioni di autorizzazione come previsto.

Ad esempio, è possibile eseguire il test `IsAuthorized` in un archivio di policy di esempio con la seguente procedura.

Test bench

1. Apri la console delle autorizzazioni verificate nella console delle autorizzazioni verificate. Crea un policy store dal Policy Store di Sample con il nome. DigitalPetStore
2. Seleziona Test bench nel tuo nuovo policy store.

3. Compila la tua richiesta di test bench dal IsAuthorizedriferimento all'API Verified Permissions. I seguenti dettagli replicano le condizioni dell'Esempio 4 che fanno riferimento all'esempio. DigitalPetStore

   a. Imposta Alice come principale. Affinché il preside agisca, scegli `DigitalPetStore::User` ed entra`Alice`.

   b. Imposta il ruolo di Alice come cliente. Scegli Aggiungi un genitore`DigitalPetStore::Role`, scegli e inserisci Cliente.

   c. Imposta la risorsa come ordine «1234». Per la risorsa su cui agisce il principale, scegli `DigitalPetStore::Order` ed inserisci`1234`.

   d. La `DigitalPetStore::Order` risorsa richiede un `owner` attributo. Imposta Alice come proprietaria dell'ordine. Scegli `DigitalPetStore::User` ed entra `Alice`

   e. Alice ha richiesto di visualizzare l'ordine. Per Azione che il preside sta intraprendendo, scegli`DigitalPetStore::Action::"GetOrder"`.

4. Scegli Esegui richiesta di autorizzazione. In un archivio di policy non modificato, questa richiesta genera una `ALLOW` decisione. Nota la politica di soddisfazione che ha restituito la decisione.

5. Scegli Politiche dalla barra di navigazione a sinistra. Consulta la politica statica con la descrizione Customer Role - Get Order.

6. Tieni presente che Verified Permissions ha consentito la richiesta perché il responsabile ricopriva il ruolo di cliente ed era il proprietario della risorsa.

REST API

1. Apri la console delle autorizzazioni verificate nella console delle autorizzazioni verificate. Crea un policy store dal Policy Store di Sample con il nome. DigitalPetStore

2. Annota l'ID del Policy Store del tuo nuovo Policy Store.

3. Dal riferimento IsAuthorizedall'API Verified Permissions, copia il corpo della richiesta dell'Esempio 4 che fa riferimento all'DigitalPetStoreesempio.

4. Apri il tuo client API e crea una richiesta all'endpoint di servizio regionale per il tuo policy store. Compila le intestazioni come mostrato nell'esempio.

5. Incolla il corpo della richiesta di esempio e modifica il valore di nell'ID del `policyStoreId` Policy Store che hai annotato in precedenza.

6. Invia la richiesta ed esamina i risultati. In un archivio di DigitalPetStorepolicy predefinito, questa richiesta restituisce una `ALLOW` decisione.

È possibile apportare modifiche alle politiche, allo schema e alle richieste nell'ambiente di test per modificare i risultati e produrre decisioni più complesse.

1. Modifica la richiesta in modo da modificare la decisione presa in Autorizzazioni verificate. Ad esempio, modifica il ruolo di Alice `Employee` o modifica l'`owner`attributo dell'ordine 1234 in`Bob`.
2. Modifica le politiche in modo da influire sulle decisioni di autorizzazione. Ad esempio, modifica la politica con la descrizione Customer Role - Get Order per rimuovere la condizione che `User` deve essere il proprietario della `Resource` e modifica la richiesta in modo che `Bob` desideri visualizzare l'ordine.
3. Modifica lo schema per consentire alle politiche di prendere decisioni più complesse. Aggiorna le entità della richiesta in modo che Alice possa soddisfare i nuovi requisiti. Ad esempio, modifica lo schema `User` per consentire di essere membro di `ActiveUsers` o`InactiveUsers`. Aggiorna la politica in modo che solo gli utenti attivi possano visualizzare i propri ordini. Aggiorna le entità della richiesta in modo che Alice sia un utente attivo o inattivo.

Integrazione dei modelli di autorizzazione con le applicazioni

Per implementare Amazon Verified Permissions nella tua applicazione, devi definire le politiche e lo schema che desideri che l'app applichi. Una volta implementato e testato il modello di autorizzazione, il passo successivo è iniziare a generare richieste API dal punto di applicazione. A tale scopo, è necessario configurare la logica dell'applicazione per raccogliere i dati degli utenti e inserirli nelle richieste di autorizzazione.

In che modo un'app autorizza le richieste con autorizzazioni verificate

1. Raccogli informazioni sull'utente corrente. In genere, i dettagli di un utente vengono forniti nei dettagli di una sessione autenticata, come un cookie JWT o di sessione web. Questi dati utente potrebbero provenire da una fonte di Amazon Cognito identità collegata al tuo policy store o da un altro provider OpenID Connect (OIDC).
2. Raccogli informazioni sulla risorsa a cui un utente desidera accedere. In genere, l'applicazione riceve informazioni sulla risorsa quando un utente effettua una selezione che richiede all'app di caricare una nuova risorsa.
3. Determina l'azione che l'utente desidera intraprendere.

4. Genera una richiesta di autorizzazione a Verified Permissions con il principale, l'azione, la risorsa e le entità per il tentativo di operazione dell'utente. Verified Permissions valuta la richiesta rispetto alle politiche dell'archivio delle politiche e restituisce una decisione di autorizzazione.
5. L'applicazione legge la risposta di autorizzazione o rifiuto di Verified Permissions e applica la decisione sulla richiesta dell'utente.

Le operazioni dell'API Verified Permissions sono integrate. AWS SDKs Per includere le autorizzazioni verificate in un'app, integra l' AWS SDK per la lingua scelta nel pacchetto dell'app.

Per saperne di più e per effettuare il download AWS SDKs, consulta Strumenti per. Amazon Web Services

Di seguito sono riportati i collegamenti alla documentazione per varie AWS SDKs risorse relative alle autorizzazioni verificate.

- AWS SDK per .NET
- AWS SDK per C++
- AWS SDK per Go
- AWS SDK per Java
- AWS SDK per JavaScript
- AWS SDK per PHP
- AWS SDK per Python (Boto)
- AWS SDK per Ruby
- AWS SDK per Rust

Il seguente AWS SDK per JavaScript esempio di `IsAuthorized` proviene da Simplifica l'autorizzazione granulare con Amazon Verified Permissions e 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"
               }
           }
        }
      ]
   }
});
```

Altre risorse per gli sviluppatori

- Workshop sulle autorizzazioni verificate di Amazon
- Autorizzazioni verificate da Amazon - Risorse
- Implementa un provider di policy di autorizzazione personalizzato per le app ASP.NET Core utilizzando Amazon Verified Permissions
- Crea un servizio di autorizzazione per le applicazioni aziendali utilizzando Amazon Verified Permissions
- Semplifica l'autorizzazione granulare con Amazon Verified Permissions e Amazon Cognito

Sicurezza nelle autorizzazioni verificate da Amazon

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per maggiori informazioni sui programmi di conformità che si applicano alle autorizzazioni verificate di Amazon, consulta AWS Services in Scope by Compliance Program AWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando utilizzi le autorizzazioni verificate. I seguenti argomenti mostrano come configurare le autorizzazioni verificate per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse relative alle autorizzazioni verificate.

Argomenti

Protezione dei dati in Amazon Verified Permissions

Il modello di responsabilità AWS condivisa modello si applica alla protezione dei dati in Amazon Verified Permissions. Come descritto in questo modello, AWS è responsabile della protezione

dell'infrastruttura globale che esegue tutto l' Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Questo contenuto include le attività di configurazione e gestione della sicurezza per Servizi AWS quello che utilizzi. Per maggiori informazioni sulla privacy dei dati, consulta le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

- Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti.
- Ti consigliamo di proteggere i tuoi dati nei seguenti modi:
  - Utilizza l'autenticazione a più fattori (MFA) con ogni account.
  - SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2.
  - Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail.
  - Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
  - Utilizza servizi di sicurezza gestiti avanzati come Amazon Macie, che aiutano a scoprire e proteggere i dati sensibili archiviati in. Amazon S3
  - Se necessiti di moduli crittografici convalidati FIPS 140-2 quando accedi ad AWS attraverso un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.
- Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con autorizzazioni verificate o altro Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.
- I nomi delle tue azioni non devono includere informazioni sensibili.
- Inoltre, ti consigliamo vivamente di utilizzare sempre identificatori unici, non modificabili e non riutilizzabili per le tue entità (risorse e principali). In un ambiente di test, puoi scegliere di utilizzare identificatori di entità semplici, come `jane` o `bob` per il nome di un'entità di tipo. `User` Tuttavia, in un sistema di produzione, è fondamentale per motivi di sicurezza utilizzare valori univoci che non

possano essere riutilizzati. Ti consigliamo di utilizzare valori come identificatori univoci universali (). UUIDs Ad esempio, si consideri l'utente `jane` che lascia l'azienda. Successivamente, consenti a qualcun altro di usare il nome`jane`. Quel nuovo utente ottiene automaticamente l'accesso a tutto ciò che è concesso dalle politiche a cui fanno ancora riferimento`User::"jane"`. Verified Permissions e Cedar non riescono a distinguere tra il nuovo utente e l'utente precedente.

Questa guida si applica sia agli identificatori principali che a quelli di risorse. Utilizza sempre identificatori che siano univoci garantiti e mai riutilizzati per assicurarti di non concedere l'accesso involontariamente a causa della presenza di un vecchio identificatore in una politica.

- Assicurati che le stringhe che fornisci per definire `Long` e `Decimal` i valori rientrino nell'intervallo valido di ogni tipo. Inoltre, assicuratevi che l'utilizzo di qualsiasi operatore aritmetico non produca un valore al di fuori dell'intervallo valido. Se l'intervallo viene superato, l'operazione genera un'eccezione di overflow. Una politica che genera un errore viene ignorata, il che significa che una politica di autorizzazione potrebbe inaspettatamente non consentire l'accesso o una politica di divieto potrebbe inaspettatamente non riuscire a bloccare l'accesso.

Crittografia dei dati

Amazon Verified Permissions crittografa automaticamente tutti i dati dei clienti, ad esempio le politiche, con un. Chiave gestita da AWS Amazon Verified Permissions consente inoltre ai clienti di utilizzare un per chiave gestita dal cliente crittografare i propri dati.

Per informazioni dettagliate sull'utilizzo delle chiavi gestite dal cliente per la crittografia, consulta. the section called “Chiavi gestite dal cliente”

Risorse di crittografia in Amazon Verified Permissions

Amazon Verified Permissions fornisce la crittografia di default per proteggere i dati sensibili dei clienti archiviati utilizzando chiavi AWS di crittografia proprietarie. Come ulteriore livello di protezione, Amazon Verified Permissions ti consente di crittografare i tuoi archivi di polizze utilizzando AWS Key Management Service (AWS KMS) chiave gestita dal cliente s (CMK). Questa funzionalità garantisce la protezione dei dati sensibili tramite la crittografia inattiva, che ti aiuta a:

- Riduci l'onere operativo sull'applicazione per proteggere i dati sensibili
- Mantieni il controllo su chi può vedere i dettagli delle tue politiche di autorizzazione tramite AWS KMS chiave gestita dal cliente i tuoi

- Creare applicazioni ad alto livello di sicurezza che rispettano rigorosi requisiti normativi e di conformità per la crittografia

Le sezioni seguenti spiegano come configurare la crittografia per i nuovi archivi di policy e come gestire le chiavi di crittografia.

AWS KMS Tipi di chiave per le autorizzazioni Amazon Verified

Amazon Verified Permissions si integra con la gestione delle chiavi AWS KMS di crittografia utilizzate per i dati encrypting/decrypting dei clienti. Per maggiori informazioni sui tipi e gli stati delle chiavi, consulta Concetti di AWS Key Management Service nella Guida per gli sviluppatori di AWS KMS . Quando crei un nuovo archivio di politiche, puoi scegliere tra i seguenti tipi di AWS KMS chiave per crittografare i tuoi dati:

AWS Chiave di proprietà

Il tipo di crittografia predefinito. Amazon Verified Permissions possiede la chiave senza costi aggiuntivi e crittografa i dati delle risorse inattivi al momento della creazione. Non è richiesta alcuna configurazione aggiuntiva nel codice o nelle applicazioni dei dati utilizzando encrypt/decrypt la chiave di proprietà di Verified Permissions.

Chiave gestita dal cliente

Crei, possiedi e gestisci la chiave nel tuo AWS account. Hai il pieno controllo della AWS KMS chiave. AWS KMS si applicano costi per chiave gestita dal cliente s. Per maggiori informazioni, consulta la pagina Prezzi di AWS KMS. Per ulteriori informazioni sui tipi di chiave, consulta Customer managed keys nella AWS KMS Developer Guide.

Quando si specifica un crittografiaggio chiave gestita dal cliente per risorse di primo livello (ad esempio, Policy Store), Verified Permissions crittografa la risorsa, così come le relative risorse secondarie, con quella chiave. Per crittografare un archivio di politiche utilizzando un chiave gestita dal cliente, è necessario concedere l'accesso alle autorizzazioni verificate nella politica chiave. Una politica chiave è una politica basata sulle risorse che alleghi alla tua per controllarne l' chiave gestita dal cliente accesso. Per ulteriori dettagli, consulta the section called "Autorizzazione all'uso della tua AWS KMS chiave per Amazon Verified Permissions".

Inoltre, per creare un policy store crittografato con a o per effettuare chiamate API a un policy store crittografato da a chiave gestita dal cliente, anche l'utente o il ruolo IAM che effettua la chiamata deve avere accesso alla chiave. chiave gestita dal cliente Se Verified Permissions non è in grado di

accedere alla chiave, qualsiasi decisione di autorizzazione che coinvolge risorse crittografate da tale chiave potrebbe essere obsoleta o imprecisa. Se non hai accesso alla chiave, non sarai in grado di crittografare read/update/delete le risorse con quella chiave e qualsiasi chiamata di creazione per utilizzare la chiave per la crittografia avrà esito negativo.

> **Note**
>
> La crittografia inattiva delle autorizzazioni verificate è disponibile in tutte le AWS regioni in cui sono disponibili le autorizzazioni verificate.

> **Important**
>
> Una volta utilizzato chiave gestita dal cliente un Policy Store per crittografare un Policy Store, NON è POSSIBILE aggiornare la risorsa per utilizzare una chiave di crittografia diversa o rimuovere la chiave da tale Policy Store.

Utilizzo AWS KMS e chiavi dati con Amazon Verified Permissions

La funzionalità di crittografia a riposo di Amazon Verified Permissions utilizza una AWS KMS chiave e una gerarchia di chiavi dati per proteggere i dati delle risorse.

> **Note**
>
> Amazon Verified Permissions supporta solo chiavi simmetriche AWS KMS . Non puoi utilizzare una AWS KMS chiave asimmetrica per crittografare le tue risorse Amazon Verified Permissions.

Utilizzo di chiavi possedute AWS

Amazon Verified Permissions crittografa tutte le risorse per impostazione predefinita con chiavi AWS di proprietà. Queste chiavi possono essere utilizzate gratuitamente e ruotano ogni anno per proteggere le risorse degli account. Non è necessario visualizzare, gestire, utilizzare o controllare queste chiavi, quindi non è necessaria alcuna azione per la protezione dei dati. Per ulteriori informazioni sulle chiavi di AWS proprietà, consulta le chiavi AWS possedute nella Guida per gli AWS KMS sviluppatori.

Utilizzo delle chiavi gestite dal cliente

La scelta di un chiave gestita dal cliente metodo per la crittografia offre i seguenti vantaggi:

- È possibile creare e gestire la AWS KMS chiave, inclusa l'impostazione delle politiche e delle IAM politiche chiave per controllare l'accesso alla AWS KMS chiave. È possibile abilitare e disabilitare la AWS KMS chiave, abilitare e disabilitare la rotazione automatica dei tasti ed eliminare la AWS KMS chiave quando non è più in uso.
- Puoi utilizzarne una chiave gestita dal cliente con materiale chiave importato o una chiave gestita dal cliente in un archivio chiavi personalizzato di tua proprietà e gestione.
- Puoi controllare la crittografia e la decrittografia delle tue risorse Verified Permissions esaminando le chiamate dell'API Amazon Verified Permissions ai log. AWS KMS AWS CloudTrail

Affinché Amazon Verified Permissions utilizzi chiave gestita dal cliente i tuoi messaggi per encryption/decryption, you will need to add specific key policies to allow Amazon Verified Permissions to encrypt/decrypt risorse per tuo conto.

Autorizzazione all'uso della tua AWS KMS chiave per Amazon Verified Permissions

Amazon Verified Permissions richiede almeno le seguenti autorizzazioni su un: chiave gestita dal cliente

- `kms:Encrypt`
- `kms:GenerateDataKeyWithoutPlaintext`
- `kms:DescribeKey`
- `kms:ReEncryptTo`
- `kms:ReEncryptFrom`
- `kms:Decrypt`

Di seguito è riportato un esempio di politica chiave:

```
{
    "Sid": "Enable AVP to use the KMS key for encrypting project J.A.K. policy
 resources",
    "Effect": "Allow",
    "Principal": {
```

```
        "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Encrypt",
        "kms:ReEncryptFrom",
        "kms:ReEncryptTo",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Comprensione del contesto di origine

Il contesto di origine fornisce informazioni sul tentativo del chiamante di origine di eseguire AWS KMS azioni su una determinata chiave. Ciò impedisce la confusione o l'uso improprio dei dati crittografati associando il contesto alla fonte dei dati.

I clienti possono utilizzare il contesto di origine come condizioni aggiuntive per la loro politica chiave, ad esempio le seguenti dichiarazioni politiche chiave:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable this account full access to this key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable AVP to retrieve this key's metadata",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
                }
            }
        },
        {
            "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:Encrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
                }
            }
        }
    ]
}
```

Questa politica chiave consente a Verified Permissions di effettuare AWS KMS chiamate per conto dell'utente, se l'account di origine è lo stesso dell'account in cui risiede la AWS KMS chiave. Questi valori devono essere verificabili quando si controllano i registri di AWS CloudTrail controllo per la chiave CMK. Per ulteriori informazioni sulle chiavi di AWS condizione globali, consulta Utilizzo `aws:SourceArn` delle chiavi di condizione o di condizione. `aws:SourceAccount`

Comprensione del contesto di crittografia

Il contesto di crittografia è un insieme di coppie chiave-valore che contengono dati autenticati aggiuntivi per i controlli di integrità della crittografia. Quando si include un contesto di crittografia in una richiesta di crittografia dei dati, associa AWS KMS crittograficamente il contesto di crittografia ai dati crittografati. Per decrittografare i dati, è necessario passare lo stesso contesto di crittografia.

Amazon Verified Permissions utilizza lo stesso contesto di crittografia in tutte le operazioni AWS KMS crittografiche e può essere verificato all'interno dei AWS CloudTrail log quando Verified Permissions effettua AWS KMS chiamate per tuo conto per i processi. encryption/decryption Per impostazione predefinita, Verified Permissions utilizza le seguenti coppie chiave-valore del contesto di crittografia per crittografare le risorse:

```
{
    "aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
}
```

Amazon Verified Permissions ti consente anche di aggiungere un contesto di crittografia personalizzato come parte dei metadati aggiuntivi che desideri includere durante i processi. encryption/decryption Ciò significa che la tua politica chiave può essere più dettagliata nella concessione delle autorizzazioni, come nell'esempio seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable this account full access to this key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable AVP to retrieve this key's metadata",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
```

```
            "Action": "kms:DescribeKey",
            "Resource": "*"
        },
        {
            "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:Encrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/*",
                    "kms:EncryptionContext:policy_owner": "Tim"
                }
            }
        }
    ]
}
```

Questa politica chiave consente a Verified Permissions di effettuare AWS KMS chiamate per conto dell'utente, se la mappa del contesto di crittografia contiene una chiave `aws:verifiedpermissions:policy-store-arn` il cui valore segue il formato `arn:aws:verifiedpermissions::111122223333:policy-store/*` e contiene anche una coppia chiave-valore. `"policy_owner": "Tim"` Scopri come impostare the section called "Creazione di un archivio di policy crittografato" un contesto di crittografia personalizzato.

> **Note**
>
> È consigliabile che le politiche chiave con condizioni basate sul contesto di crittografia si riferiscano a un sottoinsieme della mappa del contesto di crittografia, anziché verificare ogni coppia chiave-valore. Il servizio e le sue dipendenze a monte possono aggiungere ulteriori coppie chiave-valore che non sono visibili all'utente e possono influire sull'accesso alle chiavi

di Verified Permissions se la politica delle chiavi lo consente in base all'aspetto esatto della mappa del contesto di crittografia.

Comprensione kms: ViaService

La chiave `kms:ViaService` condizionale limita l'uso di una AWS KMS chiave alle richieste provenienti da AWS servizi specifici. Questa chiave condizionale si applica solo alle sessioni di accesso diretto (FAS). Per ulteriori informazioni su`kms:ViaService`, consulta kms: ViaService nella AWS KMS Developer Guide.

Ad esempio, la seguente dichiarazione politica chiave utilizza la chiave `kms:ViaService` condition per consentire l'utilizzo di chiave gestita dal cliente per le azioni specificate solo quando la richiesta proviene da Amazon Verified Permissions nella regione Stati Uniti orientali (Virginia settentrionale) per conto di. `BrentRole`

```
{
    "Sid": "Enable AVP to encrypt/decrypt resources using credentials of BrentRole",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/BrentRole"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Encrypt",
        "kms:ReEncryptFrom",
        "kms:ReEncryptTo",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": [
                "verifiedpermissions.us-east-1.amazonaws.com"
            ]
        }
    }
}
```

Ciò è necessario affinché Verified Permissions sia in grado di trasmettere la tua identità, le tue autorizzazioni e gli attributi di sessione quando Verified Permissions invia una richiesta di crittografia/decrittografia per tuo AWS KMS conto. Per ulteriori informazioni sulle richieste FAS, consulta Forward Access Sessions nella Guida per l'utente.IAM

Politica AWS KMS chiave completa

Sulla base dei concetti delle sezioni precedenti, questo è un esempio di policy chiave che consentirà ad Amazon Verified Permissions di utilizzare una CMK per la crittografia/decrittografia:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable this account full access to this key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable AVP to retrieve this key's metadata",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
                }
            }
        },
        {
            "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:Encrypt",
                "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/*",
                    "kms:EncryptionContext:policy_owner": "Tim",
                    "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "Enable AVP to encrypt/decrypt resources using credentials of
 BrentRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/BrentRole"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:Encrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "verifiedpermissions.us-east-1.amazonaws.com"
```

```
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/*",
                    "kms:EncryptionContext:policy_owner": "Tim"
                }
            }
        }
    ]
}
```

> **Warning**
>
> Fai attenzione quando modifichi le politiche AWS KMS chiave per le chiavi già utilizzate da Amazon Verified Permissions. Sebbene Verified Permissions convalidi le autorizzazioni di crittografia e decrittografia quando configuri inizialmente una AWS KMS chiave durante la creazione di risorse di primo livello, non può verificare le successive modifiche alle politiche su richiesta. La rimozione inavvertitamente delle autorizzazioni necessarie potrebbe interrompere le decisioni di autorizzazione e i regolari flussi del servizio Verified Permissions. Per indicazioni sulla risoluzione degli errori comuni relativi a chiave gestita dal cliente s in Amazon Verified Permissions, consulta. the section called "Risolvi i problemi relativi alle chiavi gestite dai clienti in Amazon Verified Permissions"

IAM Politiche necessarie per le risorse crittografate

I clienti che richiamano le autorizzazioni verificate tramite un IAM ruolo all'interno del proprio account dovranno assicurarsi che la IAM politica corrispondente disponga delle autorizzazioni appropriate per utilizzarle chiave gestita dal cliente per la crittografia e la decrittografia delle risorse.

Per creare archivi di policy crittografati da a chiave gestita dal cliente, la seguente IAM politica illustra le azioni minime necessarie e con Autorizzazioni verificate a tale scopo: AWS KMS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "verifiedpermissions:CreatePolicyStore",
```

```
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "kms:Decrypt",
                "kms:Encrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:DescribeKey",
                "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

> **Note**
>
> Per il recupero (operazioni Get* e List*) e l'eliminazione degli archivi di policy crittografati da a, non sono necessarie autorizzazioni aggiuntive. chiave gestita dal cliente

Per aggiornare un policy store crittografato da a chiave gestita dal cliente, recuperare (operazioni Get* e List*), aggiornare ed eliminare le risorse secondarie di un policy store crittografato da a, il seguente criterio illustra le azioni minime necessarie e con Autorizzazioni chiave gestita dal cliente verificate a IAM tal fine: AWS KMS

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

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

Come singola IAM policy, i clienti possono semplicemente aggiungere quanto segue alla propria policy relativa ai ruoli: IAM

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "verifiedpermissions:*",
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "kms:Decrypt",
                "kms:Encrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:DescribeKey",
                "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Gestione degli archivi di policy crittografati

Gli archivi delle politiche sono il contenitore base che conterrà tutte le risorse relative alle politiche. Per ulteriori informazioni sugli archivi delle politiche e sulla gerarchia delle risorse per bambini, consulta gli archivi di policy di Amazon Verified Permissions nella Amazon Verified Permissions User Guide.

Quando crei un archivio di policy in Verified Permissions, puoi abilitare la crittografia inattiva utilizzando le chiavi. AWS KMS Ciò garantisce che:

- Tutte le operazioni di lettura, aggiornamento ed eliminazione sugli archivi di policy e sulle relative risorse secondarie utilizzeranno i processi chiave gestita dal cliente di decrittografia forniti
- Qualsiasi richiesta di decisione di autorizzazione (ad esempio IsAuthorized BatchIsAuthorized IsAuthorizedWithToken, ecc.) utilizzerà i processi di decrittografia forniti per la decrittografia chiave gestita dal cliente

Creazione di un archivio di policy crittografato

Prima di creare un archivio di policy crittografato, assicurati che su quello che chiave gestita dal cliente stai utilizzando siano impostate le istruzioni chiave appropriate per Amazon Verified Permissions per utilizzare la chiave per la crittografia/decrittografia. Scopri per quali autorizzazioni sono necessarie. the section called "Autorizzazione all'uso della tua AWS KMS chiave per Amazon Verified Permissions"

Utilizzando AWS CLI:

```
aws verifiedpermissions create-policy-store --region us-east-1 --encryption-settings
 file://encrypted.json --validation-settings "{\"mode\": \"OFF\"}"
```

Dove `encrypted.json` assomiglia a:

```
{
    "kmsEncryptionSettings": {
        "key": "arn:aws:kms:us-east-1:111122223333:key/12345678-90ab-cdef-ghij-
klmnopqrstuv",
        "encryptionContext": {
            "<ENCRYPTION_CONTEXT_KEY_1>": "<ENCRYPTION_CONTEXT_VALUE_1>",
            "<ENCRYPTION_CONTEXT_KEY_2>": "<ENCRYPTION_CONTEXT_VALUE_2>",
            ...
        }
    }
}
```

Assicurati di sostituire `key` con il tuo chiave gestita dal cliente ARN e di sostituire `<ENCRYPTION_CONTEXT_KEY>` e `<ENCRYPTION_CONTEXT_VALUE>` abbinare le coppie `encryptionContext` chiave-valore desiderate. `encryptionContext`può essere omesso completamente se non si desidera aggiungere coppie chiave-valore.

> **Important**
>
> Non includere la coppia `aws:verifiedpermissions:policy-store-arn` chiave-valore nel contesto di crittografia personalizzato. Questa viene aggiunta automaticamente e causerà errori di convalida se fa parte delle coppie chiave-valore del contesto di crittografia personalizzato passate.

Per ulteriori informazioni sulla disponibilità di risorse secondarie APIs di un policy store, consulta Actions nella Amazon Verified Permissions API Reference Guide.

> **Note**
>
> Se le risorse Amazon Verified Permissions AWS KMS chiave gestita dal cliente in uso vengono eliminate, disabilitate o inaccessibili a causa di una politica di AWS KMS chiave errata, la decrittografia delle risorse avrà esito negativo e quindi le decisioni di autorizzazione non saranno più valide. La perdita di accesso può essere temporanea (una politica chiave può essere corretta) o permanente (una chiave eliminata non può essere ripristinata) a seconda delle circostanze. Ti consigliamo di limitare l'accesso a operazioni critiche, come l'eliminazione o la disabilitazione della chiave. AWS KMS Inoltre, consigliamo alla tua organizzazione di impostare procedure di accesso AWS trasparenti per garantire che gli utenti privilegiati possano accedere AWS nell'improbabile eventualità che Amazon Verified Permissions non sia accessibile.

Monitoraggio dell'interazione delle autorizzazioni verificate da Amazon con AWS KMS

Puoi monitorare l'utilizzo del tuo chiave gestita dal cliente account da parte di Amazon Verified Permissions. AWS CloudTrail Ogni richiesta AWS KMS tramite Verified Permissions include il contesto di crittografia e l'ARN della chiave utilizzata (chiave gestita dal cliente il tuo) nei parametri della richiesta:

Esempio di AWS CloudTrail registrazione per: `GenerateDataKeyWithoutPlaintext`

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "verifiedpermissions.amazonaws.com"
```

```
    },
    "eventTime": "2025-09-28T16:51:04Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "verifiedpermissions.amazonaws.com",
    "userAgent": "verifiedpermissions.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwx",
        "encryptionContext": {
            "aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012",
            "policy_store_editor": "Janus"
        },
        ...
    },
    ...
}
```

Esempio di voce di AWS CloudTrail registro per`Decrypt`:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "verifiedpermissions.amazonaws.com"
    },
    "eventTime": "2025-09-28T16:53:21Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "verifiedpermissions.amazonaws.com",
    "userAgent": "verifiedpermissions.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwx",
        "encryptionContext": {
            "aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012",
            "policy_store_owner": "Elias"
        }
```

```
    },
    ...
}
```

Esempio di voce di AWS CloudTrail registro per`ReEncrypt`:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "verifiedpermissions.amazonaws.com"
    },
    "eventTime": "2025-09-28T16:51:04Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReEncrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "verifiedpermissions.amazonaws.com",
    "userAgent": "verifiedpermissions.amazonaws.com",
    "requestParameters": {
        "sourceKeyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwx",
        "destinationEncryptionContext": {
            "aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
        },
        "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "destinationKeyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-
ijkl-4567-mnopqrstuvwx",
        "sourceEncryptionContext": {
            "aws:verifiedpermissions:policy_store_arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
        },
        "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        ...
    },
    ...
}
```

Tieni presente che le voci di registro includono il `invokedBy` riferimento al principale di Amazon Verified Permissions e `encryptionContext/sourceEncryptionContext/ destinationEncryptionContext` l'inclusione nella mappa. `requestParameters`

Esempio di immissione di AWS CloudTrail log per: `DescribeKey`

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "verifiedpermissions.amazonaws.com"
    },
    "eventTime": "2025-09-28T16:51:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "verifiedpermissions.amazonaws.com",
    "userAgent": "verifiedpermissions.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwx"
    },
    ...
}
```

Tieni presente che la voce di registro include il `invokedBy` riferimento al principale di Amazon Verified Permissions.

Per ulteriori informazioni sulle voci di AWS CloudTrail registro, consulta Understanding AWS CloudTrail events nella Guida per l'AWS CloudTrail utente.

Limitazioni

Questo argomento descrive le attuali limitazioni delle autorizzazioni verificate e l'utilizzo di chiave gestita dal cliente s per la crittografia delle risorse.

- Una volta abilitata, non è possibile disabilitare la crittografia per un archivio di politiche
- Dopo aver creato un policy store senza crittografia, non è possibile aggiornare il policy store in modo che venga crittografato da un chiave gestita dal cliente
- Dopo aver revocato l'accesso tramite Autorizzazioni verificate a un archivio chiave gestita dal cliente di criteri crittografato esistente, è possibile che vengano prese decisioni di autorizzazione obsolete
- Dopo aver creato un policy store con a chiave gestita dal cliente, non è possibile modificare i valori del contesto di crittografia personalizzato; si tratta di valori statici impostati durante la creazione dell'archivio delle politiche crittografato

Risolvi i problemi relativi alle chiavi gestite dai clienti in Amazon Verified Permissions

Questo argomento descrive gli errori chiave gestita dal cliente correlati più comuni che potresti riscontrare durante l'utilizzo di Amazon Verified Permissions e fornisce le procedure di risoluzione dei problemi per risolverli.

Accesso negato: problema di AWS KMS autorizzazione

Errore: «Il servizio o il chiamante non è autorizzato a utilizzare la AWS KMS chiave fornita, poiché la risorsa non esiste in questa regione, nessuna politica basata sulle risorse consente l'accesso o una politica basata sulle risorse nega esplicitamente l'accesso»

Ciò potrebbe significare che il servizio o il chiamante non dispongono delle autorizzazioni di `kms:*` azione richieste nella policy o nella IAM policy chiave oppure che la AWS KMS chiave a cui si fa riferimento non esiste o non esiste più.

Risoluzione dei problemi con: AWS CloudTrail

- Cerca `kms.amazonaws.com` eventi in AWS CloudTrail
- Cerca il nome dell'evento dell' AWS KMS operazione che è stata identificata come non consentita (ad esempio `Decrypt` `ReEncryptGenerateDataKeyWithoutPlaintext`,`DescribeKey`,, ecc.)
- Controlla i `errorMessage` campi `errorCode` e
- Controlla `userIdentity` per confermare quale preside ha tentato l'operazione

Per risolvere il problema, concedi all'utente o al IAM principale i permessi di accesso appropriati per l' AWS KMS operazione nella IAM policy e nella policy AWS KMS chiave. Per ulteriori informazioni, consulta the section called “Politica AWS KMS chiave completa”.

Eccezione di convalida: configurazione chiave AWS KMS

Errore: «La AWS KMS chiave configurata non ha una configurazione valida»

Ciò significa che la chiave a cui si fa riferimento non può essere utilizzata dal servizio per la chiave gestita dal cliente crittografia a causa della sua configurazione corrente. I motivi possono includere che la chiave è disabilitata, la chiave ha un tipo non supportato EncryptionAlgorithm o la chiave ha un tipo non KeyUsage supportato.

Eccezione di limitazione: limiti di velocità AWS KMS

Errore: «Hai superato la velocità alla quale puoi chiamare» AWS KMS

Questo errore indica che hai superato il AWS KMS limite per le operazioni di crittografia per la tua chiave: -per-second.html. https://docs.aws.amazon.com/kms/ latest/developerguide/requests

Informazioni correlate

- Gestione degli archivi basati sulla politica delle autorizzazioni verificate
- AWS KMS Best practice
- AWS KMS Contesto di crittografia
- AWS CloudTrail Integration
- AWS CloudTrail Esempi di immissione nel registro

Gestione delle identità e degli accessi per Amazon Verified Permissions

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. IAM gli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse di Autorizzazioni verificate. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso tramite policy
- Come funziona Amazon Verified Permissions con IAM
- IAM politiche per le autorizzazioni verificate
- Esempi di policy basate sull'identità per Amazon Verified Permissions
- AWS politiche gestite per Amazon Verified Permissions
- Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Verified Permissions

Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Verified Permissions)
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta Come funziona Amazon Verified Permissions con IAM)
- IAM amministratore: scrive politiche per gestire l'accesso (vediEsempi di policy basate sull'identità per Amazon Verified Permissions)

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo. Utente root dell'account AWS IAM

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/ Facebook Per ulteriori informazioni sull'accesso, consulta Come accedere all' Account AWS nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l'utente.IAM

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono credenziali utente root, consulta Attività che richiedono credenziali utente root nella Guida per l'IAM utente.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta Che cos'è il Centro identità IAM? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida per l'utente.IAM

Un IAM gruppo specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per IAM gli utenti nella Guida per l'IAM utente.

IAM ruoli

Un IAM ruolo è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. È possibile assumere temporaneamente un IAM ruolo in Console di gestione AWS cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere Utilizzo IAM dei ruoli nella Guida per l'IAM utente.

IAM i ruoli con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato - Per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per informazioni sui ruoli per la federazione, consulta Creare un ruolo per un provider di identità di terze parti (federazione) nella Guida per l'IAM utente. Se si utilizza IAM Identity Center, configurare un set di autorizzazioni. Centro identità IAM mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare le risorse alle quali le identità possono accedere dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

- Autorizzazioni utente IAM temporanee: un utente o un ruolo IAM può assumere un IAM ruolo per acquisire temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (principale attendibile) di un account diverso di accedere alle risorse nel tuo account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta In che modo IAM i ruoli differiscono dalle politiche basate sulle risorse nella Guida per l'utente.IAM
- Applicazioni in esecuzione Amazon EC2: puoi utilizzare un IAM ruolo per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che effettuano richieste API. AWS CLI AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, devi creare un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un IAM ruolo per concedere le autorizzazioni alle applicazioni in esecuzione su Amazon EC2 istanze nella Guida per l'utente.IAM

Per sapere se utilizzare IAM i ruoli o gli utenti IAM, consulta Quando creare un IAM ruolo (anziché un utente) nella Guida per l'IAM utente.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sui documenti relativi alle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'IAM utente.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un IAM amministratore crea IAM le politiche e le aggiunge ai ruoli, che gli utenti possono quindi assumere. IAM le politiche definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguire l'operazione.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una politica basata sull'identità, consulta Definire le IAM autorizzazioni personalizzate con le politiche gestite dal cliente nella Guida per l'utente.IAM

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra politiche gestite e politiche in linea, consulta Scegliere tra politiche gestite e politiche in linea nella Guida per l'utente.IAM

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di IAM role trust e le Amazon S3 bucket policy. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio specificare un'entità principale.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite da IAM una policy basata sulle risorse.

Elenchi di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 AWS WAF, e Amazon VPC sono esempi di servizi che supportano. ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta le autorizzazioni massime che una politica basata sull'identità può concedere a un'entità. IAM Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità nella Guida per l' IAMutente.IAM
- Politiche di controllo del servizio (SCPs): specifica le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta Politiche di controllo delle risorse (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di IAM .

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per informazioni su come AWS determinare se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle politiche nella Guida per l'IAM utente.

Come funziona Amazon Verified Permissions con IAM

Prima di utilizzare IAM per gestire l'accesso alle autorizzazioni verificate, scopri quali IAM funzionalità sono disponibili per l'uso con le autorizzazioni verificate.

IAM funzionalità che puoi utilizzare con Amazon Verified Permissions

IAM funzionalità	Supporto per le autorizzazioni verificate
Policy basate sull'identità	Sì
Policy basate su risorse	No
Operazioni di policy	Sì
Risorse relative alle policy	Sì

IAM funzionalità	Supporto per le autorizzazioni verificate
Chiavi di condizione delle policy	No
ACLs	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	No
Ruoli collegati al servizio	No

Per avere una panoramica generale del funzionamento delle autorizzazioni verificate e degli altri AWS servizi con la maggior parte delle IAM funzionalità, consulta AWS i servizi con cui funzionano IAM nella Guida per l'IAM utente.

Politiche basate sull'identità per le autorizzazioni verificate

Supporta le policy basate su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta Definire le IAM autorizzazioni personalizzate con le politiche gestite dal cliente nella Guida per l'utente.IAM

Con le politiche IAM basate sull'identità, puoi specificare azioni e risorse consentite o negate, nonché le condizioni in base alle quali le azioni sono consentite o negate. Per informazioni su tutti gli elementi che è possibile utilizzare in una policy JSON, consulta il riferimento agli elementi della policy IAM JSON nella Guida per l'utente.IAM

Esempi di policy basate sull'identità per le autorizzazioni verificate

Per visualizzare esempi di politiche basate sull'identità delle autorizzazioni verificate, consulta. Esempi di policy basate sull'identità per Amazon Verified Permissions

Politiche basate sulle risorse all'interno delle autorizzazioni verificate

Supporta le policy basate su risorse	No

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di politiche basate sulle risorse sono le politiche di fiducia dei ruoli e le IAM politiche dei bucket. Amazon S3 Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio specificare un'entità principale. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per abilitare l'accesso tra più account, puoi specificare un intero account o IAM entità in un altro account come principale in una politica basata sulle risorse. Per ulteriori informazioni, consulta la sezione Cross Account Resource Access IAM nella Guida per l'utente.IAM

Azioni politiche per le autorizzazioni verificate

Supporta le operazioni di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di azioni relative alle autorizzazioni verificate, consulta Azioni definite da Amazon Verified Permissions nel Service Authorization Reference.

Le azioni politiche in Verified Permissions utilizzano il seguente prefisso prima dell'azione:

```
verifiedpermissions
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

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

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Get`, includi la seguente azione:

```
"Action": "verifiedpermissions:Get*"
```

Per visualizzare esempi di politiche basate sull'identità delle autorizzazioni verificate, consulta. Esempi di policy basate sull'identità per Amazon Verified Permissions

Risorse politiche per le autorizzazioni verificate

Supporta le risorse di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di risorse Verified Permissions e relativi ARNs, consulta Tipi di risorse definiti da Amazon Verified Permissions nel Service Authorization Reference. Per sapere con quali azioni puoi specificare l'ARN di ogni risorsa, consulta Azioni definite da Amazon Verified Permissions.

Chiavi relative alle condizioni della policy per le autorizzazioni verificate

Supporta le chiavi di condizione delle policy specifiche del servizio	No

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'IAM utente.

ACLs in Autorizzazioni verificate

Supporti ACLs	No

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con autorizzazioni verificate

Supporta ABAC (tag nelle policy)	Sì

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. Puoi allegare tag a IAM entità e AWS risorse, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Definire le autorizzazioni con l'autorizzazione ABAC nella Guida per l'utente.IAM Per visualizzare un tutorial con i passaggi per configurare ABAC, consulta Utilizzare il controllo degli accessi basato sugli attributi (ABAC) nella Guida per l'utente.IAM

Utilizzo di credenziali temporanee con autorizzazioni verificate

Supporta le credenziali temporanee	Sì

Le credenziali temporanee forniscono l'accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM e Servizi AWS utilizzabili IAM nella Guida per l'IAM utente.

Autorizzazioni principali multiservizio per le autorizzazioni verificate

Supporta le autorizzazioni delle entità principali	Sì

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per le autorizzazioni verificate

Supporta i ruoli di servizio	No

Un ruolo di servizio è un IAM ruolo che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'interno IAM. Per ulteriori informazioni, consulta Creare un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAM utente.

Ruoli collegati ai servizi per le autorizzazioni verificate

Supporta i ruoli collegati ai servizi	No

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono

visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.

Per informazioni dettagliate sulla creazione o la gestione di ruoli collegati ai servizi, consulta AWS Servizi compatibili con. IAM Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

IAM politiche per le autorizzazioni verificate

Verified Permissions gestisce le autorizzazioni degli utenti all'interno dell'applicazione. Affinché l'applicazione possa richiamare le autorizzazioni verificate APIs o Console di gestione AWS consentire agli utenti di gestire le politiche Cedar in un archivio di policy per le autorizzazioni verificate, è necessario aggiungere le autorizzazioni necessarie. IAM

Le politiche basate sull'identità sono documenti di policy sulle autorizzazioni JSON che è possibile allegare a un'identità, ad esempio un utente, un gruppo di utenti o un IAM ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta Creazione di politiche nella Guida per l'utente. IAM IAM

Con le politiche IAM basate sull'identità, puoi specificare azioni e risorse consentite o negate, nonché le condizioni in base alle quali le azioni sono consentite o negate (elencate di seguito). Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi che è possibile utilizzare in una policy JSON, consulta il riferimento agli elementi della policy IAM JSON nella Guida per l'utente. IAM

Azione	Descrizione
CreateIdentitySource	Azione per creare una nuova fonte di identità.
CreatePolicy	Azione per creare una policy Cedar in un archivio di policy. È possibile creare una politica statica o una politica collegata a un modello di politica.
CreatePolicyStore	Azione per creare un nuovo archivio delle politiche.

Azione	Descrizione
CreatePolicyTemplate	Azione per creare un nuovo modello di policy.
DeleteIdentitySource	Azione per eliminare una fonte di identità.
DeletePolicy	Azione per eliminare una policy da un policy store.
DeletePolicyStore	Azione per eliminare un policy store.
DeletePolicyTemplate	Azione per eliminare un modello di policy.
GetIdentitySource	Azione per ottenere una fonte di identità.
GetPolicy	Azione per recuperare informazioni su una politica specificata.
GetPolicyStore	Azione per recuperare informazioni su un archivio di politiche specificato.
GetPolicyTemplate	Azione per ottenere un modello di policy.
GetSchema	Azione per ottenere uno schema.
IsAuthorized	Azione per ottenere una risposta di autorizzazione basata sui parametri descritti nella richiesta di autorizzazione.
IsAuthorizedWithToken	Azione per ottenere una risposta di autorizzazione basata sui parametri descritti nella richiesta di autorizzazione in cui il principale proviene da un token di identità.
ListIdentitySources	Azione per elencare tutte le fonti di identità in Account AWS.
ListPolicies	Azione per elencare tutte le politiche in un archivio delle politiche.

Azione	Descrizione
ListPolicyStores	Azione per elencare tutti gli archivi delle politiche in Account AWS.
ListPolicyTemplates	Azione per elencare tutti i modelli di policy in Account AWS.
ListTagsForResource	Azione per elencare tutti i tag di una risorsa.
PutSchema	Azione per aggiungere uno schema a un archivio di politiche.
TagResource	Azione per aggiungere un tag a una risorsa.
UpdateIdentitySource	Azione per aggiornare una fonte di identità.
UpdatePolicy	Azione per aggiornare una policy in un policy store.
UpdatePolicyStore	Azione per aggiornare un archivio delle politiche.
UpdatePolicyTemplate	Azione per aggiornare un modello di policy.
UntagResource	Azione per rimuovere un tag da una risorsa.

Esempio IAM di politica per l'autorizzazione all' CreatePolicy azione:

JSON

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

```
        }
    ]
}
```

Esempi di policy basate sull'identità per Amazon Verified Permissions

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse con autorizzazioni verificate. Inoltre, non possono eseguire attività utilizzando Console di gestione AWS, AWS Command Line Interface (AWS CLI) o l' AWS API. Un IAM amministratore deve creare IAM politiche che concedano a utenti e ruoli l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno. L'amministratore deve quindi collegare queste policy agli utenti che ne hanno bisogno.

Per informazioni su come creare una policy IAM basata sull'identità utilizzando questi esempi di documenti di policy JSON, consulta Creating IAM policies in the User Guide.IAM

Per dettagli sulle azioni e sui tipi di risorse definiti da Verified Permissions, incluso il formato di ARNs per ogni tipo di risorsa, consulta Azioni, risorse e chiavi di condizione per Amazon Verified Permissions nel Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console delle autorizzazioni verificate
- Consentire agli utenti di visualizzare le loro autorizzazioni

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse Verified Permissions nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti

specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta le politiche AWS gestite o le politiche AWS gestite per le funzioni lavorative nella Guida per l'IAM utente.

- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le IAM politiche, concedi solo le autorizzazioni necessarie per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per ulteriori informazioni sull'utilizzo per applicare le autorizzazioni, consulta Politiche e autorizzazioni nella Guida IAM per l'utente. IAMIAM
- Utilizza le condizioni nelle IAM politiche per limitare ulteriormente l'accesso: puoi aggiungere una condizione alle tue politiche per limitare l'accesso ad azioni e risorse. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. È inoltre possibile utilizzare le condizioni per concedere l'accesso alle azioni di servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM : condizione nella Guida per l'utente di IAM .
- Utilizza IAM Access Analyzer per convalidare IAM le tue policy e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio delle IAM policy (JSON) e alle best practice IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy con IAM Access Analyzer nella Guida per l'utente.IAM
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per ulteriori informazioni, consulta Accesso sicuro alle API con MFA nella Guida per l'IAM utente.

Per ulteriori informazioni sulle best practice in IAM, consulta la sezione Procedure consigliate in materia di sicurezza IAM nella Guida per l'IAM utente.

Utilizzo della console delle autorizzazioni verificate

Per accedere alla console Amazon Verified Permissions, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse di Autorizzazioni verificate presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, è opportuno concedere l'accesso solo alle azioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la console Autorizzazioni verificate, allega anche le Autorizzazioni verificate *ConsoleAccess* o la politica *ReadOnly* AWS gestita alle entità. Per ulteriori informazioni, consulta Aggiungere autorizzazioni a un utente nella Guida per l'utente.IAM

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. 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": "*"
        }
    ]
}
```

AWS politiche gestite per Amazon Verified Permissions

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le politiche AWS gestite piuttosto che scrivere le politiche autonomamente. Ci vogliono tempo ed esperienza per creare politiche gestite dai IAM clienti che forniscano al team solo le autorizzazioni di cui ha bisogno. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste policy coprono i casi d'uso comuni e sono disponibili nell'account Account AWS. Per ulteriori informazioni sulle politiche AWS gestite, consulta le politiche AWS gestite nella Guida IAM per l'utente.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy ReadOnlyAccess AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per un elenco e le descrizioni delle politiche relative alle funzioni lavorative, consulta le politiche AWS gestite per le funzioni lavorative nella Guida per l'utente.IAM

AWS politica gestita: AmazonVerifiedPermissionsFullAccess

La politica `AmazonVerifiedPermissionsFullAccess` gestita garantisce l'accesso completo alle autorizzazioni verificate. Per utilizzare fonti di identità Amazon Cognito basate sull'identità, dovrai allegare una policy separata, come la AmazonCognitoReadOnlypolicy.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AccountLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:CreatePolicyStore",
        "verifiedpermissions:ListPolicyStores"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PolicyStoreLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:*"
      ],
      "Resource": [
        "arn:aws:verifiedpermissions::*:policy-store/*"
      ]
    }
  ]
}
```

AWS politica gestita: AmazonVerifiedPermissionsReadOnlyAccess

La politica `AmazonVerifiedPermissionsReadOnlyAccess` gestita concede l'accesso in sola lettura alle autorizzazioni verificate.

Questa politica garantisce l'accesso a tutte le operazioni di lettura di Amazon Verified Permissions, inclusa la richiesta APIs `IsAuthorized` di autorizzazione e. `IsAuthorizedWithToken`

> **Note**
>
> L'accesso a `BatchIsAuthorized` e `BatchIsAuthorizedWithToken` viene concesso automaticamente quando l'accesso viene concesso rispettivamente a `IsAuthorized` e`IsAuthorizedWithToken`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AccountLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:ListPolicyStores"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PolicyStoreLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:GetIdentitySource",
        "verifiedpermissions:GetPolicy",
        "verifiedpermissions:GetPolicyStore",
        "verifiedpermissions:GetPolicyTemplate",
        "verifiedpermissions:GetSchema",
        "verifiedpermissions:IsAuthorized",
        "verifiedpermissions:IsAuthorizedWithToken",
        "verifiedpermissions:ListIdentitySources",
        "verifiedpermissions:ListPolicies",
        "verifiedpermissions:ListPolicyTemplates"
      ],
      "Resource": [
        "arn:aws:verifiedpermissions::*:policy-store/*"
      ]
    }
  ]
}
```

Autorizzazioni verificate: aggiornamenti alle politiche AWS gestite.

Visualizza i dettagli sugli aggiornamenti alle politiche AWS gestite per le autorizzazioni verificate da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici

sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina di cronologia dei documenti sulle autorizzazioni verificate.

Modifica	Descrizione	Data
AmazonVerifiedPermissionsFullAccess: nuova policy	Autorizzazioni verificate ha aggiunto una nuova politica per consentire l'accesso completo alle autorizzazioni verificate.	11 ottobre 2024
AmazonVerifiedPermissionsReadOnlyAccess: nuova policy	Verified Permissions ha aggiunto una nuova politica per consentire l'accesso a tutte le operazioni di lettura di Amazon Verified Permissions, inclusa la richiesta APIs `IsAuthorized` di autorizzazione e. `IsAuthorizedWithToken`	11 ottobre 2024
Verified Permissions ha iniziato a tenere traccia delle modifiche	Verified Permissions ha iniziato a tenere traccia delle modifiche alle sue politiche AWS gestite.	11 ottobre 2024

Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Verified Permissions

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con Autorizzazioni verificate e. IAM

Argomenti

- Non sono autorizzato a eseguire un'azione in Autorizzazioni verificate
- Non sono autorizzato a eseguire iam: PassRole

- Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse relative alle autorizzazioni verificate

Non sono autorizzato a eseguire un'azione in Autorizzazioni verificate

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `verifiedpermissions:`*`GetWidget`* fittizie.

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

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `verifiedpermissions:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di assegnare un ruolo a Autorizzazioni verificate.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio anziché creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in Autorizzazioni verificate. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

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

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse relative alle autorizzazioni verificate

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se Verified Permissions supporta queste funzionalità, consulta. Come funziona Amazon Verified Permissions con IAM
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM di un altro Account AWS utente di tua proprietà nella Guida per l'IAM utente.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a persone Account AWS di proprietà di terzi nella Guida per l'IAM utente.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso agli utenti autenticati esternamente (federazione delle identità) nella Guida per l'IAM utente.
- Per conoscere la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle risorse tra account nella Guida per l'utente. IAMIAM

Convalida della conformità per Amazon Verified Permissions

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta AWS la documentazione sulla sicurezza.

Resilienza nelle autorizzazioni verificate da Amazon

L'infrastruttura AWS globale è costruita attorno a Regioni AWS zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Quando si crea un archivio di criteri per le autorizzazioni verificate, questo viene creato all'interno di un singolo Regione AWS utente e viene replicato automaticamente nei data center che costituiscono le zone di disponibilità di quella regione. Al momento, Verified Permissions non supporta alcuna replica tra regioni.

Per ulteriori informazioni sulle Regioni AWS zone di disponibilità, consulta AWS Global Infrastructure.

Monitoraggio delle chiamate API Amazon Verified Permissions

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon Verified Permissions e delle altre AWS soluzioni. AWS fornisce i seguenti strumenti per monitorare le autorizzazioni verificate, segnalare quando qualcosa non va e intraprendere azioni automatiche se necessario:

- AWS CloudTrailacquisisce le chiamate API e gli eventi correlati effettuati da o per conto dell' AWS account e invia i file di registro a un Amazon S3 bucket specificato dall'utente. È possibile identificare quali utenti e account hanno effettuato le chiamate AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la Guida per l'utente AWS CloudTrail.

Per ulteriori informazioni sul monitoraggio delle autorizzazioni verificate con CloudTrail, consultaRegistrazione delle chiamate API Amazon Verified Permissions utilizzando AWS CloudTrail.

Registrazione delle chiamate API Amazon Verified Permissions utilizzando AWS CloudTrail

Amazon Verified Permissions è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in Autorizzazioni verificate. CloudTrail acquisisce tutte le chiamate API per le autorizzazioni verificate come eventi. Le chiamate acquisite includono chiamate dalla console Verified Permissions e chiamate in codice alle operazioni dell'API Verified Permissions. Se crei un percorso, puoi abilitare l'invio continuo di CloudTrail eventi a un Amazon S3 bucket, inclusi gli eventi per le autorizzazioni verificate. Se non configuri un trail, puoi comunque visualizzare gli eventi di gestione più recenti nella CloudTrail console nella cronologia degli eventi, ma non gli eventi per le chiamate API come. `isAuthorized` Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta effettuata a Verified Permissions, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e altri dettagli.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

Informazioni sulle autorizzazioni verificate in CloudTrail

CloudTrail è abilitato sul tuo Account AWS quando crei l'account. Quando si verifica un'attività in Autorizzazioni verificate, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nell' Account AWS. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi del tuo Account AWS, compresi gli eventi per le autorizzazioni verificate, crea un percorso. Un trail consente di CloudTrail inviare file di registro a un Amazon S3 bucket. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il trail registra gli eventi da tutte le regioni della AWS partizione e consegna i file di registro nel Amazon S3 bucket specificato. Inoltre, è possibile configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le azioni di Verified Permissions vengono registrate CloudTrail e documentate nella Amazon Verified Permissions API Reference Guide. Ad esempio, le chiamate a `CreateIdentitySourceDeletePolicy`, e le `ListPolicyStores` azioni generano voci nei file di registro. CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Per impostazione predefinita, IsAuthorizedgli eventi relativi ai dati, ad esempio, non IsAuthorizedWithTokenvengono registrati quando si crea un trail o un data store di eventi. Per registrare gli eventi CloudTrail relativi ai dati, è necessario aggiungere in modo esplicito le risorse o i tipi di risorse supportati per i quali si desidera raccogliere attività. Per ulteriori informazioni, consulta Eventi di dati nella Guida per l'utente AWS CloudTrail .

Informazioni sulle voci del file di registro delle autorizzazioni verificate

Un trail è una configurazione che consente la consegna di eventi come file di registro in un Amazon S3 bucket specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

Per le chiamate API di autorizzazione, gli elementi di risposta, come la decisione, sono inclusi in `additionalEventData` anzichéresponseElements.

Argomenti

- IsAuthorized
- BatchIsAuthorized
- CreatePolicyStore
- ListPolicyStores
- DeletePolicyStore
- PutSchema
- GetSchema
- CreatePolicyTemplate
- DeletePolicyTemplate
- CreatePolicy
- GetPolicy
- CreateIdentitySource
- GetIdentitySource
- ListIdentitySources
- DeleteIdentitySource

> **Note**
>
> Alcuni campi degli esempi sono stati oscurati per la privacy dei dati.

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"
}
```

Creazione di risorse Amazon Verified Permissions con AWS CloudFormation

Amazon Verified Permissions è integrato con AWS CloudFormation, un servizio che ti aiuta a modellare e configurare AWS le tue risorse in modo da poter dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura. Crei un modello che descrive tutte le AWS risorse che desideri (come gli archivi delle politiche) e fornisce e CloudFormation configura tali risorse per te.

Quando lo utilizzi CloudFormation, puoi riutilizzare il modello per configurare le risorse relative alle autorizzazioni verificate in modo coerente e ripetuto. Descrivi le tue risorse una sola volta, quindi fornisci le stesse risorse più e più volte in più aree geografiche Account AWS .

> **Important**
>
> Amazon Cognito Identity non è affatto disponibile come Regioni AWS Amazon Verified Permissions. Se ricevi un errore relativo ad Amazon Cognito Identity, ad esempio, ti consigliamo di creare il pool di Amazon Cognito utenti e il client nel luogo geograficamente più vicino in cui è disponibile Amazon Regione AWS Cognito Identity. CloudFormation `Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient` Usa questo pool di utenti appena creato per creare la fonte di identità Verified Permissions.

Autorizzazioni e modelli verificati CloudFormation

Per fornire e configurare le risorse per le autorizzazioni verificate e i servizi correlati, è necessario conoscere CloudFormation i modelli. I modelli sono file di testo formattati in JSON o YAML. Questi modelli descrivono le risorse che desideri fornire nei tuoi CloudFormation stack. Se non conosci JSON o YAML, puoi usare CloudFormation Designer per iniziare a usare i modelli. CloudFormation Per ulteriori informazioni, consulta Cos'è Designer? CloudFormation nella Guida AWS CloudFormation per l'utente.

Verified Permissions supporta la creazione di fonti di identità, policy, policy store, modelli di policy e alias di policy store in. CloudFormationPer ulteriori informazioni, inclusi esempi di modelli JSON e YAML per le risorse Verified Permissions, consulta il riferimento al tipo di risorsa Amazon Verified Permissions nella Guida per l'utente.AWS CloudFormation

AWS Costrutti CDK

AWS Cloud Development Kit (AWS CDK) È un framework di sviluppo software open source per definire l'infrastruttura cloud in codice e fornirla tramite. CloudFormation I costrutti, o componenti cloud riutilizzabili, possono essere utilizzati per creare modelli. CloudFormation Questi modelli possono quindi essere utilizzati per implementare l'infrastruttura cloud.

Per saperne di più e scaricare AWS CDK, consulta AWS Cloud Development Kit.

Di seguito sono riportati i collegamenti alla documentazione relativa alle AWS CDK risorse relative alle autorizzazioni verificate, ad esempio i costrutti.

- Autorizzazioni verificate Amazon L2 CDK Construct

Scopri di più su CloudFormation

Per ulteriori informazioni CloudFormation, consulta le seguenti risorse:

- AWS CloudFormation
- AWS CloudFormation Guida per l'utente
- CloudFormation Documentazione di riferimento delle API
- AWS CloudFormation Guida per l'utente dell'interfaccia a riga di comando

Accedi alle autorizzazioni verificate di Amazon utilizzando AWS PrivateLink

Puoi utilizzarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e Amazon Verified Permissions. Puoi accedere alle autorizzazioni verificate come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. Direct Connect Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per accedere alle autorizzazioni verificate.

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creata un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Si tratta di interfacce di rete gestite dai richiedenti che fungono da punto di ingresso per il traffico destinato alle autorizzazioni verificate.

Per ulteriori informazioni, consulta la sezione Accesso a Servizi AWS tramite AWS PrivateLink nella Guida di AWS PrivateLink .

Considerazioni relative alle autorizzazioni verificate

Prima di configurare un endpoint di interfaccia per le autorizzazioni verificate, consulta le considerazioni nella Guida.AWS PrivateLink

Verified Permissions supporta l'esecuzione di chiamate a tutte le sue azioni API tramite l'endpoint dell'interfaccia.

Le policy degli endpoint VPC non sono supportate per le autorizzazioni verificate. Per impostazione predefinita, l'accesso completo alle autorizzazioni verificate è consentito tramite l'endpoint dell'interfaccia. In alternativa, è possibile associare un gruppo di sicurezza alle interfacce di rete degli endpoint per controllare il traffico verso le autorizzazioni verificate attraverso l'endpoint dell'interfaccia.

Crea un endpoint di interfaccia per le autorizzazioni verificate

Puoi creare un endpoint di interfaccia per le autorizzazioni verificate utilizzando la console Amazon VPC o (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta la sezione Creazione di un endpoint di interfaccia nella Guida per l'utente di AWS PrivateLink .

Crea un endpoint di interfaccia per le autorizzazioni verificate utilizzando il seguente nome di servizio:

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

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi effettuare richieste API a Verified Permissions utilizzando il nome DNS regionale predefinito. Ad esempio, `verifiedpermissions.us-east-1.amazonaws.com`.

Creazione di una policy dell' endpoint per l'endpoint dell'interfaccia

Una policy dell'endpoint è una risorsa IAM che è possibile allegare all'endpoint dell'interfaccia. La policy predefinita per gli endpoint consente l'accesso completo alle autorizzazioni verificate tramite l'endpoint dell'interfaccia. Per controllare l'accesso consentito alle autorizzazioni verificate dal tuo VPC, collega una policy personalizzata per gli endpoint all'endpoint dell'interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (Account AWS, utenti IAM e ruoli IAM).
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.

Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai servizi con policy di endpoint nella Guida di AWS PrivateLink .

Esempio: policy degli endpoint VPC per le azioni relative alle autorizzazioni verificate

Di seguito è riportato l'esempio di una policy dell'endpoint personalizzata. Quando alleghi questa policy all'endpoint dell'interfaccia, concede l'accesso alle azioni di Autorizzazioni verificate elencate per tutti i principali utenti su tutte le risorse.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "verifiedpermissions:IsAuthorized",
            "verifiedpermissions:IsAuthorizedWithToken",
            "verifiedpermissions:GetPolicy"
```

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

Quote per le autorizzazioni verificate da Amazon

Your Account AWS ha delle quote predefinite, precedentemente denominate limiti, per ogni servizio. AWS Salvo diversa indicazione, ogni quota si applica a una Regione specifica. Se per alcune quote è possibile richiedere aumenti, altre quote non possono essere modificate.

Per visualizzare le quote per le autorizzazioni verificate, apri la console Service Quotas. Nel riquadro di navigazione, scegli AWS servizi e seleziona Autorizzazioni verificate.

Per richiedere un aumento delle quote, consultare Richiesta di aumento delle quote nella Guida per l'utente di Service Quotas. Se la quota non è ancora disponibile in Service Quotas, utilizza il modulo di incremento dei limiti.

Hai Account AWS le seguenti quote relative alle autorizzazioni verificate.

Argomenti

Quote per le risorse

Name	Predefinita	Adattabile	Description
Le policy vengono archiviate per regione e per account	Ogni regione supportata: 30.000	Sì	Il numero massimo di archivi di polizze.
Modelli di policy per archivio di policy	Ogni regione supportata: 40	Sì	Il numero massimo di modelli di policy in un archivio di policy.
Fonti di identità per archivio di policy	1	No	Il numero massimo di fonti di identità che è possibile definire per un archivio di politiche.

Name	Predefinita	Adattabile	Description
Alias del Policy Store per Policy Store	10	Sì	Il numero massimo di alias del policy store che è possibile associare a un singolo policy store.
Dimensione della richiesta di autorizzazione[1]	1 MB	No	La dimensione massima di una richiesta di autorizzazione.
Dimensione della politica	10,000 byte	Sì	La dimensione massima di una singola politica.
Dimensioni dello schema	100.000 byte	Sì	La dimensione massima dello schema di un archivio di politiche.
Dimensione della policy per risorsa	200.000 byte[2]	Sì	La dimensione massima di tutte le politiche che fanno riferimento a una risorsa specifica.

[1] La quota per una richiesta di autorizzazione è la stessa per entrambi IsAuthorizedeIsAuthorizedWithToken.

[2] Il limite predefinito per la dimensione totale di tutte le politiche previste per una singola risorsa è di 200.000 byte. Analogamente, la dimensione totale di tutte le politiche, in cui l'ambito lascia la risorsa indefinita, applicandosi quindi a tutte le risorse, è limitata per impostazione predefinita a 200.000 byte. Si noti che per le policy collegate al modello, la dimensione del modello di policy viene conteggiata una sola volta, più la dimensione di ogni set di parametri utilizzato per creare un'istanza di ogni policy collegata al modello. Questo limite può essere aumentato, a condizione che la progettazione della policy soddisfi determinati vincoli. Se hai bisogno di esplorare questa opzione, contatta. Supporto

Esempio di dimensione della policy collegato a un modello

È possibile determinare in che modo le politiche collegate ai modelli contribuiscono alla dimensione della politica per quota di risorse calcolando la somma della lunghezza del principale e della risorsa. Se il principale o la risorsa non sono specificati, la lunghezza di quel pezzo è 0. Se una risorsa non è specificata, la sua dimensione viene conteggiata ai fini della quota di `"unspecified"` risorse. La dimensione del corpo del modello in sé non ha alcun impatto sulla dimensione della politica.

Diamo un'occhiata al seguente modello:

```
@id("template1")
permit (
  principal in ?principal,
  action in [Action::"view", Action::"comment"],
  resource in ?resource
)
unless {
  resource.tag =="private"
};
```

Creiamo le seguenti politiche da quel modello:

```
TemplateLinkedPolicy {
  policyId: "policy1",
  templateId: "template1",
  principal: User::"alice",
  resource: Photo::"car.jpg"
}

TemplateLinkedPolicy {
  policyId: "policy2",
  templateId: "template1",
  principal: User::"bob",
  resource: Photo::"boat.jpg"
}

TemplateLinkedPolicy {
  policyId: "policy3",
  templateId: "template1",
  principal: User::"jane",
  resource: Photo::"car.jpg"
```

```
TemplateLinkedPolicy {
  policyId: "policy4",
  templateId: "template1",
  principal: User::"jane",
  resource
}
```

Ora calcoliamo la dimensione di queste politiche contando i caratteri presenti nella `principal` e `resource` per ognuna di esse. Ogni carattere conta come 1 byte.

La dimensione di `policy1` sarebbe la lunghezza del principale `User::"alice"` (13) più la lunghezza della risorsa `Photo::"car.jpg"` (16). Sommandoli abbiamo 13 + 16 = 29 byte.

La dimensione di `policy2` sarebbe la lunghezza del principale `User::"bob"` (11) più la lunghezza della risorsa `Photo::"boat.jpg"` (17). Sommandoli abbiamo 11+ 17 = 28 byte.

La dimensione di `policy3` sarebbe la lunghezza del principale `User::"jane"` (12) più la lunghezza della risorsa `Photo::"car.jpg"` (16). Sommandoli abbiamo 12 + 16 = 28 byte.

La dimensione di `policy4` sarebbe la lunghezza del principale `User::"jane"` (12) più la lunghezza della risorsa (0). Sommandoli abbiamo 12 + 0 = 12 byte.

Poiché `policy2` è l'unica politica che fa riferimento alla risorsa`Photo::"boat.jpg"`, la dimensione totale della risorsa è di 28 byte.

Poiché `policy1` `policy3` entrambi fanno riferimento alla risorsa`Photo::"car.jpg"`, la dimensione totale della risorsa è 29 + 28 = 57 byte.

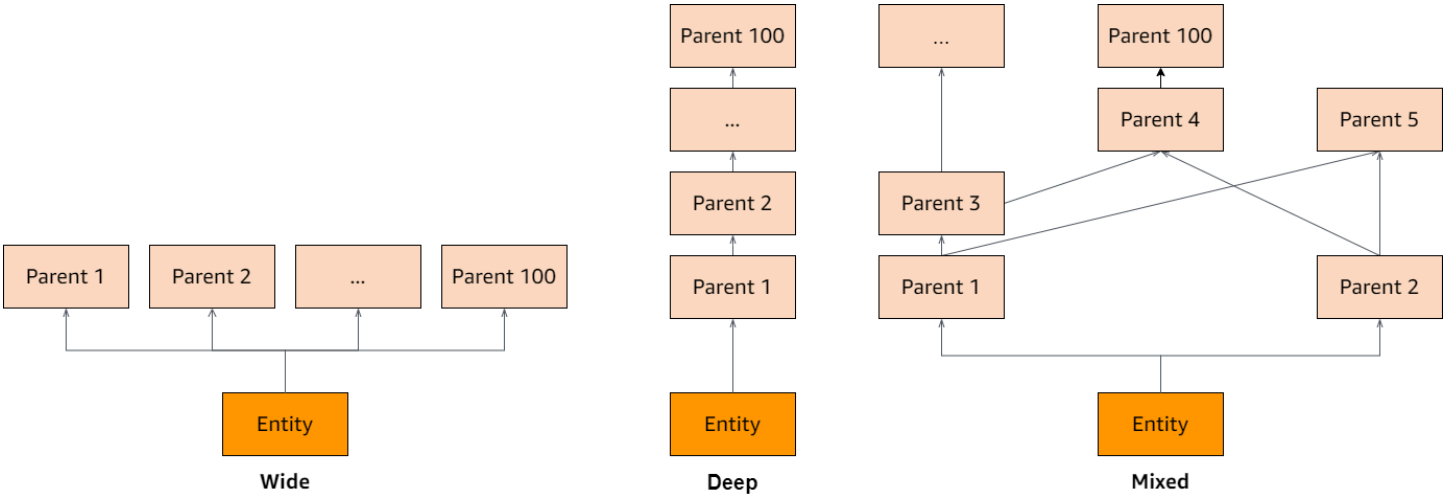
Poiché `policy4` è l'unica politica che fa riferimento alla `"unspecified"` risorsa, la dimensione totale della risorsa è di 12 byte.

Quote per le gerarchie

> **Note**
>
> Le seguenti quote sono aggregate, ovvero sommate. Il numero massimo di genitori transitivi per il gruppo è quello elencato. Ad esempio, se il limite di genitori transitivi per preside è 100, significa che potrebbero esserci 100 genitori dei dirigenti e 0 genitori sia per le azioni che per le risorse, o qualsiasi combinazione di genitori che sommi a 100 genitori totali.

Name	Predefinita	Adatta e	Description
Genitori transitivi per preside	100	No	Il numero massimo di genitori transitivi per ogni principale.
Genitori transitivi per azione	100	No	Il numero massimo di genitori transitivi per ogni azione.
Genitori transitivi per risorsa	100	No	Il numero massimo di genitori transitivi per ogni risorsa.

Il diagramma seguente illustra come è possibile definire i genitori transitivi per un'entità (principale, azione o risorsa).



Quote per operazioni al secondo

Verified Permissions limita le richieste agli endpoint di servizio Regione AWS quando le richieste delle applicazioni superano la quota per un'operazione API. Verified Permissions potrebbe restituire un'eccezione quando si supera la quota di richieste al secondo o si tentano operazioni di scrittura simultanee. È possibile visualizzare le quote RPS correnti in Service Quotas. Per evitare che le applicazioni superino la quota per un'operazione, è necessario ottimizzarle per i nuovi tentativi e il

backoff esponenziale. Per ulteriori informazioni, consulta Riprova con schema di backoff e Gestione e monitoraggio della limitazione delle API nei carichi di lavoro.

Name	Predefinita	Adattabile	Description
BatchGetPolicy richieste al secondo per regione per archivio di politiche	Ogni regione supportata: 10	Sì	Il numero massimo di BatchGetPolicy richieste al secondo per policy store.
BatchIsAuthorized richieste al secondo per regione per policy store	Ogni regione supportata: 30	Sì	Il numero massimo di BatchIsAuthorized richieste al secondo per policy store.
BatchIsAuthorizedWithToken richieste al secondo per regione per policy store	Ogni regione supportata: 30	Sì	Il numero massimo di BatchIsAuthorizedWithToken richieste al secondo per policy store.
CreateIdentitySource richieste al secondo per regione per policy store	Ogni regione supportata: 1	Sì	Il numero massimo di CreateIdentitySource richieste al secondo per policy store.
CreatePolicy richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di CreatePolicy richieste al secondo per policy store.
CreatePolicyStore richieste al secondo per regione per account	Ogni regione supportata: 1	No	Il numero massimo di CreatePolicyStore richieste al secondo.
CreatePolicyTemplate richieste al secondo per regione per archivio di politiche	Ogni regione supportata: 10	Sì	Il numero massimo di CreatePolicyTemplate richieste al secondo per policy store.

Name	Predefinita	Adattabile	Description
DeleteIdentitySource richieste al secondo per regione per policy store	Ogni regione supportata: 1	Sì	Il numero massimo di DeleteIdentitySource richieste al secondo per policy store.
DeletePolicy richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di DeletePolicy richieste al secondo per policy store.
DeletePolicyStore richieste al secondo per regione per account	Ogni regione supportata: 1	No	Il numero massimo di DeletePolicyStore richieste al secondo.
DeletePolicyTemplate richieste al secondo per regione per archivio di politiche	Ogni regione supportata: 10	Sì	Il numero massimo di DeletePolicyTemplate richieste al secondo per policy store.
GetIdentitySource richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di GetIdentitySource richieste al secondo per policy store.
GetPolicy richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di GetPolicy richieste al secondo per policy store.
GetPolicyStore richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di GetPolicyStore richieste al secondo.
GetPolicyTemplate richieste al secondo per regione per archivio di politiche	Ogni regione supportata: 10	Sì	Il numero massimo di GetPolicyTemplate richieste al secondo per policy store.

Name	Predefinita	Adattabile	Description
GetSchema richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di GetSchema richieste al secondo per policy store.
IsAuthorized richieste al secondo per regione per policy store	Ogni Regione supportata: 200	Sì	Il numero massimo di IsAuthorized richieste al secondo per policy store.
IsAuthorizedWithToken richieste al secondo per regione per policy store	Ogni Regione supportata: 200	Sì	Il numero massimo di IsAuthorizedWithToken richieste al secondo per policy store.
ListIdentitySources richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di ListIdentitySources richieste al secondo per policy store.
ListPolicies richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di ListPolicies richieste al secondo per policy store.
ListPolicyStores richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di ListPolicyStores richieste al secondo.
ListPolicyTemplates richieste al secondo per regione per archivio di politiche	Ogni regione supportata: 10	Sì	Il numero massimo di ListPolicyTemplates richieste al secondo per policy store.
PutSchema richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di PutSchema richieste al secondo per policy store.

Name	Predefinita	Adattabile	Description
UpdateIdentitySource richieste al secondo per regione per policy store	Ogni regione supportata: 1	Sì	Il numero massimo di UpdateIdentitySource richieste al secondo per policy store.
UpdatePolicy richieste al secondo per regione per policy store	Ogni regione supportata: 10	Sì	Il numero massimo di UpdatePolicy richieste al secondo per policy store.
UpdatePolicyStore richieste al secondo per regione per account	Ogni regione supportata: 10	No	Il numero massimo di UpdatePolicyStore richieste al secondo.
UpdatePolicyTemplate richieste al secondo per regione per archivio di politiche	Ogni regione supportata: 10	Sì	Il numero massimo di UpdatePolicyTemplate richieste al secondo per policy store.

Termini e concetti relativi al linguaggio delle politiche Amazon Verified Permissions e Cedar

È necessario comprendere i seguenti concetti per utilizzare Amazon Verified Permissions.

Concetti relativi alle autorizzazioni verificate

- Modello di autorizzazione
- Richiesta di autorizzazione
- Risposta di autorizzazione
- Politiche considerate
- Dati contestuali
- Definizione delle politiche
- Dati dell'entità
- Autorizzazioni, autorizzazioni e principi
- Applicazione delle politiche
- Archivio delle politiche
- Alias dell'archivio delle politiche
- Nome policy
- Nome del modello di policy
- Politiche soddisfatte
- Differenze tra Amazon Verified Permissions e il linguaggio delle policy Cedar

Concetti del linguaggio Cedar Policy

- Autorizzazione
- Entità
- Gruppi e gerarchie
- Spazi dei nomi
- Policy
- Modello di politica
- Schema

Modello di autorizzazione

Il modello di autorizzazione descrive l'ambito delle richieste di autorizzazione effettuate dall'applicazione e la base per la valutazione di tali richieste. È definito in termini di diversi tipi di risorse, azioni intraprese su tali risorse e tipi principali che eseguono tali azioni. Considera inoltre il contesto in cui vengono intraprese tali azioni.

Il controllo degli accessi basato sul ruolo (RBAC) è una base di valutazione in cui i ruoli sono definiti e associati a una serie di autorizzazioni. Questi ruoli possono quindi essere assegnati a una o più identità. L'identità assegnata acquisisce le autorizzazioni associate al ruolo. Se le autorizzazioni associate al ruolo vengono modificate, la modifica influirà automaticamente su qualsiasi identità a cui è stato assegnato il ruolo. Cedar può supportare le decisioni dell'RBAC attraverso l'uso di gruppi principali.

Il controllo degli accessi basato sugli attributi (ABAC) è una base di valutazione in cui le autorizzazioni associate a un'identità sono determinate dagli attributi di tale identità. Cedar può supportare le decisioni ABAC attraverso l'uso di condizioni politiche che fanno riferimento agli attributi del principale.

Il linguaggio di policy Cedar consente la combinazione di RBAC e ABAC in un'unica politica, consentendo di definire le autorizzazioni per un gruppo di utenti, che dispongono di condizioni basate sugli attributi.

Richiesta di autorizzazione

Una richiesta di autorizzazione è una richiesta di autorizzazioni verificate da parte di un'applicazione per valutare una serie di politiche al fine di determinare se un responsabile può eseguire un'azione su una risorsa per un determinato contesto.

Risposta di autorizzazione

La risposta di autorizzazione è la risposta alla richiesta di autorizzazione. Include una decisione di autorizzazione o rifiuto, oltre a informazioni aggiuntive, come le IDs politiche determinanti.

Politiche considerate

Le politiche considerate sono l'insieme completo di politiche che vengono selezionate da Verified Permissions per l'inclusione durante la valutazione di una richiesta di autorizzazione.

Dati contestuali

I dati contestuali sono valori di attributo che forniscono informazioni aggiuntive da valutare.

Definizione delle politiche

Le politiche determinanti sono le politiche che determinano la risposta di autorizzazione. Ad esempio, se esistono due politiche soddisfatte, in cui una è una negazione e l'altra è una politica di autorizzazione, la politica di rifiuto sarà la politica determinante. Se esistono più politiche di autorizzazione soddisfatte e nessuna politica di divieto soddisfatto, esistono più politiche di determinazione. Nel caso in cui nessuna politica corrisponda e la risposta sia negata, non esistono politiche determinanti.

Dati dell'entità

I dati dell'entità sono dati relativi al principale, all'azione e alla risorsa. I dati delle entità rilevanti per la valutazione delle politiche sono l'appartenenza al gruppo fino alla gerarchia delle entità e i valori degli attributi del principale e della risorsa.

Autorizzazioni, autorizzazioni e principi

Verified Permissions gestisce autorizzazioni e autorizzazioni dettagliate all'interno delle applicazioni personalizzate create dall'utente.

Un principale è l'utente di un'applicazione, umano o automatico, che ha un'identità legata a un identificatore come un nome utente o un ID macchina. Il processo di autenticazione determina se il principale è realmente l'identità che dichiara di essere.

A tale identità è associato un insieme di autorizzazioni dell'applicazione che determinano le operazioni che tale preside è autorizzato a fare all'interno dell'applicazione. L'autorizzazione è il processo di valutazione di tali autorizzazioni per determinare se una persona principale è autorizzata a eseguire una particolare azione nell'applicazione. Queste autorizzazioni possono essere espresse come politiche.

Applicazione delle politiche

L'applicazione delle politiche è il processo di applicazione della decisione di valutazione all'interno dell'applicazione al di fuori delle autorizzazioni verificate. Se la valutazione delle autorizzazioni

verificate restituisce un rifiuto, l'applicazione assicurerà che al principale sia impedito l'accesso alla risorsa.

Archivio delle politiche

Un policy store è un contenitore per policy e modelli. Ogni negozio contiene uno schema utilizzato per convalidare le politiche aggiunte all'archivio. Per impostazione predefinita, ogni applicazione dispone del proprio archivio delle politiche, ma più applicazioni possono condividere un unico archivio delle politiche. Quando un'applicazione effettua una richiesta di autorizzazione, identifica l'archivio delle politiche utilizzato per valutare tale richiesta. Gli archivi di policy forniscono un modo per isolare un set di policy e possono quindi essere utilizzati in un'applicazione multi-tenant per contenere gli schemi e le politiche per ogni tenant. Una singola applicazione può avere archivi di policy separati per ogni tenant.

Nel valutare una richiesta di autorizzazione, Verified Permissions considera solo il sottoinsieme delle politiche del policy store pertinenti alla richiesta. La pertinenza viene determinata in base all'ambito della politica. L'ambito identifica il principale e la risorsa specifici a cui si applica la politica e le azioni che il principale può eseguire sulla risorsa. La definizione dell'ambito aiuta a migliorare le prestazioni restringendo l'insieme delle politiche prese in considerazione.

Alias dell'archivio delle politiche

Un alias di policy store è un nome descrittivo per un policy store. È possibile utilizzare un alias del policy store per identificare un policy store in qualsiasi operazione di Verified Permissions che accetta un parametro. `policyStoreId` Gli alias del Policy Store sono AWS risorse indipendenti e proprie. ARNs Ogni alias è associato a un policy store alla volta ed è possibile associare più alias allo stesso policy store. Per ulteriori informazioni, consulta Alias dell'archivio delle policy di Amazon Verified Permissions.

Nome policy

Un nome di policy è un nome descrittivo facoltativo per un criterio. I nomi delle politiche devono essere univoci per tutte le politiche all'interno dell'archivio delle politiche e devono essere preceduti da`name/`. È possibile utilizzare il nome di una policy al posto dell'ID della policy nelle operazioni del piano di controllo che accettano un `policyId` parametro. I nomi possono essere impostati durante la creazione o l'aggiornamento di una policy. Solo `GetPolicy` e `ListPolicies` restituisce il nome nell'output.

Nome del modello di policy

Il nome di un modello di policy è un nome descrittivo facoltativo per un modello di policy. I nomi dei modelli di policy devono essere univoci per tutti i modelli di policy presenti nell'archivio delle policy e devono essere preceduti da`name/`. È possibile utilizzare il nome di un modello di policy al posto dell'ID del modello di policy nelle operazioni del piano di controllo che accettano un `policyTemplateId` parametro. I nomi possono essere impostati durante la creazione o l'aggiornamento di un modello di policy. Solo `GetPolicyTemplate` e `ListPolicyTemplates` restituisce il nome nell'output.

Politiche soddisfatte

Le politiche soddisfatte sono le politiche che corrispondono ai parametri della richiesta di autorizzazione.

Differenze tra Amazon Verified Permissions e il linguaggio delle policy Cedar

Amazon Verified Permissions utilizza il motore linguistico Cedar Policy per eseguire le proprie attività di autorizzazione. Tuttavia, ci sono alcune differenze tra l'implementazione nativa di Cedar e l'implementazione di Cedar che si trovano in Verified Permissions. Questo argomento identifica queste differenze.

Definizione dello spazio dei nomi

L'implementazione Verified Permissions di Cedar presenta le seguenti differenze rispetto all'implementazione nativa di Cedar:

- Verified Permissions supporta solo uno spazio dei nomi in uno schema definito in un policy store.
- Le autorizzazioni verificate non consentono di creare uno spazio dei nomi che sia una stringa vuota o che includa i seguenti valori:, o. `aws` `amazon` `cedar`

Supporto per modelli di policy

Sia Verified Permissions che Cedar consentono di inserire i segnaposto nell'ambito solo per il termine e. `principal` `resource` Tuttavia, le autorizzazioni verificate richiedono anche che nessuna delle due e sia priva di vincoli. `principal` `resource`

La seguente politica è valida in Cedar ma viene rifiutata da Verified Permissions perché non è vincolata. `principal`

```
permit(principal, action == Action::"view", resource == ?resource);
```

Entrambi gli esempi seguenti sono validi sia in Cedar che in Verified Permissions perché entrambi hanno dei vincoli. `principal` `resource`

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

```
permit(principal == ?principal, action == Action::"a", resource in ?resource);
```

Supporto dello schema

Verified Permissions richiede che tutti i nomi delle chiavi JSON dello schema siano stringhe non vuote. Cedar consente stringhe vuote in alcuni casi, ad esempio per proprietà o namespace.

Definizione dei gruppi di azione

I metodi di autorizzazione Cedar richiedono un elenco delle entità da prendere in considerazione quando si valuta una richiesta di autorizzazione rispetto alle politiche.

È possibile definire le azioni e i gruppi di azioni utilizzati dall'applicazione nello schema. Tuttavia, Cedar non include lo schema come parte di una richiesta di valutazione. Invece, Cedar utilizza lo schema solo per convalidare le politiche e i modelli di policy inviati. Poiché Cedar non fa riferimento allo schema durante le richieste di valutazione, anche se nello schema sono stati definiti gruppi di azioni, è necessario includere anche l'elenco di tutti i gruppi di azioni come parte dell'elenco delle entità da passare alle operazioni dell'API di autorizzazione.

Verified Permissions lo fa per te. Tutti i gruppi di azioni definiti nello schema vengono aggiunti automaticamente all'elenco di entità a cui si passa come parametro alle operazioni `IsAuthorized` or`IsAuthorizedWithToken`.

Formattazione delle entità

La formattazione JSON delle entità in Verified Permissions che utilizza il `entityList` parametro differisce da Cedar nei seguenti modi:

- In Verified Permissions, un oggetto JSON deve avere tutte le sue coppie chiave-valore racchiavette in un oggetto JSON con il nome di. `Record`
- Un elenco JSON in Verified Permissions deve essere racchiuso in una coppia chiave-valore JSON in cui il nome della chiave è e il valore è l'elenco JSON originale di Cedar. `Set`
- Per i nomi di `Boolean` tipo e tipo `StringLong`, ogni coppia chiave-valore di Cedar viene sostituita da un oggetto JSON in Verified Permissions. Il nome dell'oggetto è il nome della chiave originale. All'interno dell'oggetto JSON, esiste una coppia chiave-valore in cui il nome della chiave è il nome del tipo del valore scalare (`StringLong`, o`Boolean`) e il valore è il valore dell'entità Cedar.
- La formattazione della sintassi delle entità Cedar e delle entità Verified Permissions differisce nei seguenti modi:

Formato Cedar	Formato di autorizzazioni verificate
`uid`	`Identifier`
`type`	`EntityType`
`id`	`EntityId`
`attrs`	`Attributes`
`parents`	`Parents`

Example- Elenchi

Gli esempi seguenti mostrano come un elenco di entità viene espresso rispettivamente in Cedar e Verified Permissions.

Cedar

```
[
  {
    "number": 1
  },
  {
    "sentence": "Here is an example sentence"
  },
  {
    "Question": false
```

```
  }
]
```

Verified Permissions

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

Example- Valutazione delle politiche

Gli esempi seguenti mostrano come le entità sono formattate per la valutazione di una politica in una richiesta di autorizzazione in Cedar e Verified Permissions, rispettivamente.

Cedar

```
[
    {
        "uid": {
            "type": "PhotoApp::User",
```

```
            "id": "alice"
        },
        "attrs": {
            "age": 25,
            "name": "alice",
            "userId": "123456789012"
        },
        "parents": [
            {
                "type": "PhotoApp::UserGroup",
                "id": "alice_friends"
            },
            {
                "type": "PhotoApp::UserGroup",
                "id": "AVTeam"
            }
        ]
    },
    {
        "uid": {
            "type": "PhotoApp::Photo",
            "id": "vacationPhoto.jpg"
        },
        "attrs": {
            "private": false,
            "account": {
                "__entity": {
                    "type": "PhotoApp::Account",
                    "id": "ahmad"
                }
            }
        },
        "parents": []
    },
    {
        "uid": {
            "type": "PhotoApp::UserGroup",
            "id": "alice_friends"
        },
        "attrs": {},
        "parents": []
    },
    {
        "uid": {
```

```
            "type": "PhotoApp::UserGroup",
            "id": "AVTeam"
        },
        "attrs": {},
        "parents": []
    }
]
```

Verified Permissions

```
[
    {
        "Identifier": {
            "EntityType": "PhotoApp::User",
            "EntityId": "alice"
        },
        "Attributes": {
            "age": {
                "Long": 25
            },
            "name": {
                "String": "alice"
            },
            "userId": {
                "String": "123456789012"
            }
        },
        "Parents": [
            {
                "EntityType": "PhotoApp::UserGroup",
                "EntityId": "alice_friends"
            },
            {
                "EntityType": "PhotoApp::UserGroup",
                "EntityId": "AVTeam"
            }
        ]
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::Photo",
            "EntityId": "vacationPhoto.jpg"
        },
```

```
        "Attributes": {
            "private": {
                "Boolean": false
            },
            "account": {
                "EntityIdentifier": {
                    "EntityType": "PhotoApp::Account",
                    "EntityId": "ahmad"
                }
            }
        },
        "Parents": []
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::UserGroup",
            "EntityId": "alice_friends"
        },
        "Parents": []
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::UserGroup",
            "EntityId": "AVTeam"
        },
        "Parents": []
    }
]
```

Limiti di lunghezza e dimensione

Verified Permissions supporta l'archiviazione sotto forma di archivi di policy per archiviare schemi, policy e modelli di policy. Tale archiviazione fa sì che le autorizzazioni verificate impongano alcuni limiti di lunghezza e dimensione che non sono rilevanti per Cedar.

Oggetto	Limite di autorizzazioni verificate (in byte)	Limite Cedar
Dimensione della polizza[1]	10.000	Nessuno

Oggetto	Limite di autorizzazioni verificate (in byte)	Limite Cedar
Descrizione della politica in linea	150	Non applicabile a Cedar
Dimensioni del modello di policy	10.000	Nessuno
Dimensioni dello schema	100.000	Nessuno
Tipo di entità	200	Nessuno
ID Policy	64	Nessuno
ID del modello di policy	64	Nessuno
ID entità	200	Nessuno
ID dell'archivio delle politiche	64	Non applicabile a Cedar

[1] Esiste un limite di policy per policy store in Verified Permissions in base alla dimensione combinata dei principali, delle azioni e delle risorse dei criteri creati nell'archivio delle politiche. La dimensione totale di tutte le policy relative a una singola risorsa non può superare i 200.000 byte. Per le policy collegate al modello, la dimensione del modello di policy viene conteggiata una sola volta, più la dimensione di ogni set di parametri utilizzato per creare un'istanza di ogni policy collegata al modello.

Domande frequenti sull'aggiornamento delle autorizzazioni verificate di Amazon a Cedar 4

Amazon Verified Permissions sta aggiornando la versione di Cedar che utilizza dalla versione 2 alla versione 4. Cedar è il linguaggio open source che usi per scrivere politiche, modelli di policy e schemi nei tuoi archivi di policy. Con il supporto di Cedar 4 in Verified Permissions, puoi utilizzare nuove funzionalità come i tag `is` dell'operatore e dell'entità per scrivere politiche più espressive.

Amazon Verified Permissions aggiorna automaticamente i policy store a Cedar 4. Tuttavia, alcune politiche, schemi e richieste di autorizzazione scritte per Cedar 2 sono incompatibili con Cedar 4. Se questo è il caso del tuo policy store, non lo aggiorneremo automaticamente. Potrebbe essere necessario apportare modifiche alle politiche, ai modelli di policy, agli schemi o al codice dell'applicazione prima di poter effettuare l'aggiornamento a Cedar 4.

Argomenti

Perché alcune politiche, modelli di policy e schemi non sono compatibili con Cedar 4?

A partire da Cedar 2, il team di Cedar ha apportato diverse modifiche non compatibili con le versioni precedenti, per correggere i bug e semplificare il linguaggio. Queste modifiche includono:

- modifiche alla sintassi per politiche, modelli di policy e schemi
- un validatore di policy più preciso, che rileva più errori
- modifiche al comportamento delle funzioni integrate come `isInRange`

Per un elenco completo delle modifiche incompatibili con le versioni precedenti, cerca gli elementi contrassegnati con (*) nel changelog di Cedar.

Come faccio a sapere se il mio policy store utilizza Cedar 2 o Cedar 4?

Puoi controllare la versione di Cedar utilizzata dal tuo policy store utilizzando la console Amazon Verified Permissions o utilizzando l'GetPolicyStoreoperazione.

> **Note**
>
> Tutti gli archivi di policy della stessa Account AWS regione utilizzano la stessa versione di Cedar.

Console

Per controllare la versione Cedar di un policy store (console)

1. Accedi Console di gestione AWS e apri la console Amazon Verified Permissions all'indirizzo https://console.aws.amazon.com/verifiedpermissions/.
2. Dal pannello di navigazione, scegli Policy stores, quindi scegli il Policy Store che desideri controllare.
3. Scegliere Settings (Impostazioni) nel riquadro di navigazione.
4. Nella casella Dettagli, individua il campo della versione Cedar.

Il campo indica `CEDAR_2` se il tuo policy store utilizza Cedar 2 e `CEDAR_4` se utilizza Cedar 4.

CLI

Per verificare la versione Cedar di un policy store ()AWS CLI

1. Installa e configura AWS Command Line Interface (AWS CLI), se non l'hai già fatto. Per informazioni, consulta la pagina Installazione o aggiornamento della versione più recente di AWS CLI.
2. Utilizza il comando `get-policy-store`. Nell'esempio seguente, sostituiscilo *`policy-store-id`* con l'identificatore del tuo policy store:

```
aws verifiedpermissions get-policy-store \
  --policy-store-id policy-store-id
```

Il `cedarVersion` campo nell'output mostra la versione di Cedar utilizzata dall'archivio delle politiche. Esempio:

```
{
    "policyStoreId": "ABCDEFG12345678abcdefg",
    "arn": "arn:aws:verifiedpermissions::111122223333:policy-store/
ABCDEFG12345678abcdefg",
    "validationSettings": {
        "mode": "STRICT"
    },
    "createdDate": "2025-06-03T13:09:47.752255+00:00",
    "lastUpdatedDate": "2025-06-03T13:09:47.752255+00:00",
    "deletionProtection": "ENABLED",
    "cedarVersion": "CEDAR_2"
}
```

Il campo indica `CEDAR_2` se il tuo policy store utilizza Cedar 2 e `CEDAR_4` se utilizza Cedar 4.

Come posso effettuare l'aggiornamento a Cedar 4?

Amazon Verified Permissions ha già aggiornato la maggior parte dei clienti a Cedar 4. Se non hai mai creato un archivio di politiche, tutti i nuovi archivi di policy che creerai utilizzeranno Cedar 4. Se sei un cliente esistente, probabilmente ti abbiamo già aggiornato a Cedar 4. Consulta Come faccio a sapere se il mio policy store utilizza Cedar 2 o Cedar 4? per verificare quale versione di Cedar viene utilizzata dai tuoi archivi di polizze.

Se non sei stato aggiornato, Verified Permissions ha rilevato una policy, un modello di policy, uno schema o una richiesta di autorizzazione in uno dei tuoi archivi di policy che non è compatibile con Cedar 4. Ti invieremo una notifica via e-mail che descrive quali risorse sono incompatibili entro il 2025. Per effettuare l'upgrade prima, apri una custodia con. Supporto

> **Important**
>
> Tutti gli archivi di policy nello stesso archivio Account AWS utilizzano la stessa versione di Cedar. Se un archivio di polizze del tuo account è incompatibile con Cedar 4, non puoi utilizzare Cedar 4 in nessun archivio di polizze di quell'account.

Posso effettuare il downgrade del mio negozio di polizze da Cedar 4 a Cedar 2?

No. Se riscontri problemi dopo l'aggiornamento del tuo Policy Store a Cedar 4, apri una custodia con. Supporto

Perché ricevo un messaggio di errore che dice che il mio policy store è configurato per Cedar 2?

Alcune funzionalità di Amazon Verified Permissions si basano sulle nuove funzionalità di Cedar 4. Se il tuo policy store non utilizza Cedar 4, non puoi utilizzare i seguenti campi API:

- Nelle BatchIsAuthorizedWithToken operazioni IsAuthorizedBatchIsAuthorized, IsAuthorizedWithToken e:
  - `datetime`, `decimal` o `duration` valori nei `context` campi `attributes` o

Non puoi utilizzare la sintassi o i tipi di dati nelle politiche, nei modelli di policy o negli schemi introdotti dopo Cedar 2 finché il tuo policy store non viene aggiornato.

Come posso rendere il mio schema compatibile con Cedar 4?

La console Verified Permissions può risolvere automaticamente alcuni problemi di compatibilità nello schema. Se lo schema non può essere corretto automaticamente, la console mostrerà un elenco di errori da correggere manualmente.

> **Important**
>
> L'editor di codice nella console Amazon Verified Permissions mostra sempre gli errori e gli avvisi di Cedar 4, anche se il tuo negozio di polizze utilizza Cedar 2. Puoi continuare ad

apportare aggiornamenti dello schema non compatibili con Cedar 4 utilizzando il pulsante Salva modifiche o l'API Verified Permissions.

Per correggere uno schema utilizzando la console

1. Accedi Console di gestione AWS e apri la console Amazon Verified Permissions all'indirizzo verifiedpermissions.
2. Dal pannello di navigazione, scegli Policy stores, quindi scegli il Policy Store che desideri controllare.
3. Scegli Schema nel riquadro di navigazione.
4. Se lo schema può essere corretto automaticamente, vedrai un banner con la scritta «Fai clic su «Correggi» per visualizzare l'anteprima di una versione compatibile». Seleziona Correggi.
5. Controlla le modifiche apportate allo schema e fai clic su Anteprima schema aggiornato.
6. Rivedi lo schema aggiornato e fai clic su Salva modifiche.

Se lo schema non può essere corretto automaticamente, nella console puoi visualizzare un elenco di errori da correggere autonomamente.

1. Apri la pagina Modifica schema come descritto sopra.
2. Seleziona la modalità JSON.
3. Passa il mouse sull'icona di errore rossa nella grondaia sul lato sinistro dell'editor di codice. Il messaggio di errore viene visualizzato in un tooltip.

Ecco alcuni errori comuni che potresti riscontrare e come risolverli:

impossibile analizzare lo schema da JSON: `` ***field-name***

Con Cedar 2, puoi includere campi arbitrari in parti di schemi come le definizioni dei tipi, anche se non hanno alcun significato come parte di uno schema Cedar. In Cedar 4, questo non è più consentito. Per risolvere questo errore, rimuovi il campo chiamato *field-name* dallo schema JSON. Per un elenco di campi dello schema validi, consulta la documentazione di Cedar.

tipo di estensione sconosciuto `` ***extension-name***

In Cedar 2, quando dichiari un attributo il cui `type` è`Extension`, puoi specificare qualsiasi valore per il `name` campo, indipendentemente dal fatto che il valore sia un nome di tipo di estensione

valido o meno. Questo è ora un errore con Cedar 4. Per risolverlo, sostituiscilo *extension-name* con un nome di tipo di estensione valido. È possibile trovare un elenco di nomi di tipi di estensione validi nella documentazione di Cedar.

Se non sei ancora sicuro di come risolvere gli errori nel tuo schema, contatta Supporto

Come posso rendere le mie politiche e i miei modelli compatibili con Cedar 4?

La console Verified Permissions mostra eventuali errori nella politica o nel modello che la rendono incompatibile con Cedar 4.

Per visualizzare gli errori di una politica o di un modello nella console

1. Accedi Console di gestione AWS e apri la console Amazon Verified Permissions all'indirizzo verifiedpermissions.
2. Dal pannello di navigazione, scegli Policy stores, quindi scegli il Policy Store che desideri controllare.
3. Scegli Policy o Policy templates nel riquadro di navigazione, a seconda dei casi.
4. Seleziona la politica o il modello incompatibile.
5. Seleziona Modifica
6. Passa il mouse sull'icona di errore rossa nella grondaia sul lato sinistro dell'editor di codice. Il messaggio di errore viene visualizzato in un tooltip.

Ecco alcuni errori comuni che potresti riscontrare e come risolverli:

i set letterali vuoti sono proibiti nelle politiche

In Cedar 2, puoi usare la sintassi `mySet == []` per verificare se un set è vuoto. Con Cedar 4, le politiche che utilizzano questa sintassi non vengono più convalidate rispetto a uno schema. Sostituisci `mySet == []` nella tua politica con. `mySet.isEmpty()`

Cronologia dei documenti per la Amazon Verified Permissions User Guide

La tabella seguente descrive le versioni della documentazione per le autorizzazioni verificate.

Modifica	Descrizione	Data
Nomi di policy e nomi di modelli di policy	È ora possibile assegnare nomi alle politiche e ai modelli di policy per farvi riferimento con nomi descrittivi.	4 marzo 2025
Alias dell'archivio delle politiche	È ora possibile creare alias degli archivi delle politiche per fare riferimento ai propri archivi delle politiche con nomi descrittivi.	26 febbraio 2025
Nuove politiche AWS gestite	Ora puoi utilizzare `AmazonVerifiedPermissionsFullAccess` e `AmazonVerifiedPermissionsReadOnlyAccess` IAM gestire le politiche con autorizzazioni verificate.	11 ottobre 2024
Fonti di identità OIDC	Ora puoi autorizzare gli utenti dai provider di identità OpenID Connect (OIDC).	8 giugno 2024
Autorizzazione in batch con token di origine dell'identità	Ora puoi autorizzare gli utenti da un pool di Amazon Cognito utenti in un'unica richiesta `BatchIsAuthorizedWithToken` API.	5 aprile 2024

Creazione di un archivio di politiche con API Gateway	È ora possibile creare un policy store da un'API e da un pool di Amazon Cognito utenti esistenti.	1 aprile 2024
Concetti ed esempi relativi al contesto	Sono state aggiunte informazioni sul contesto nelle richieste di autorizzazione con autorizzazioni verificate.	1 febbraio 2024
Concetti ed esempi di autorizzazione	Sono state aggiunte informazioni sulle richieste di autorizzazione con autorizzazioni verificate.	1 febbraio 2024
AWS CloudFormation integration	Verified Permissions supporta la creazione di fonti di identità, policy, policy store e modelli di policy in. CloudFormation	30 giugno 2023
Versione iniziale	Versione iniziale della Amazon Verified Permissions User Guide	13 giugno 2023

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.