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 gli utenti delle applicazioni.

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 in un'applicazione personalizzata. Verified Permissions presuppone che il principale sia stato precedentemente identificato e autenticato con altri mezzi, ad esempio utilizzando protocolli come OpenID Connect, un provider ospitato come Amazon Cognito o un'altra soluzione di autenticazione. Verified Permissions non dipende da dove viene gestito l'utente e dal modo in cui l'utente è stato autenticato.

Verified Permissions è un servizio che consente ai clienti di creare, mantenere e testare le politiche in. AWS Management Console Le autorizzazioni sono espresse utilizzando il linguaggio di policy Cedar. L'applicazione client richiama le API di autorizzazione per valutare le politiche Cedar archiviate con il servizio e decidere in merito all'accesso se un'azione è consentita.

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 si crea un'applicazione, è necessario

assicurarsi che solo gli utenti autorizzati possano accedere all'applicazione e possano fare solo ciò a cui ciascun utente è autorizzato 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 2.4 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.

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 una directory utente, un server di autenticazione e un servizio di autorizzazione per i token di accesso OAuth 2.0 e le credenziali AWS. 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 API su qualsiasi scala. Per ulteriori informazioni, consulta la Guida per gli sviluppatori di API Gateway.

- 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 AWS IAM Identity CenterGuida per l'utente.

Accesso alle autorizzazioni verificate

Puoi utilizzare Amazon Verified Permissions in uno dei seguenti modi.

AWS Management Console

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 a riga di comando

Puoi utilizzare gli strumenti della AWS riga di comando per impartire comandi dalla riga di comando del tuo 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 offre due gruppi di strumenti a riga di comando: la AWS Command Line Interface (AWS CLI) e AWS Tools for Windows PowerShell. Per informazioni sull'installazione e la configurazione di AWS CLI, consulta la Guida per l'utente di AWS Command Line Interface. Per informazioni sull'installazione e l'utilizzo degli strumenti per Windows PowerShell, consulta la Guida per l'AWS Tools for Windows PowerShellutente.

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

AWS SDK

AWS fornisce gli SDK (kit di sviluppo software) costituiti da librerie e codice di esempio per vari linguaggi di programmazione e piattaforme (Java, Python, Ruby, .NET, iOS, Android, ecc.). Gli SDK offrono un modo conveniente per creare un accesso programmatico alle autorizzazioni verificate e. AWS Ad esempio, gli SDK si occupano di attività quali la firma crittografica delle richieste, la gestione degli errori e la ripetizione automatica delle richieste.

Per saperne di più e scaricare gli AWS SDK, consulta Strumenti per. Amazon Web Services

Di seguito sono riportati i collegamenti alla documentazione relativa alle risorse relative alle autorizzazioni verificate in vari AWS SDK.

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

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 AWS Billing and Cost Management. La fattura contiene collegamenti per passare ai report di utilizzo, che consentono di visualizzare i dettagli della fattura. Per ulteriori informazioni sulla fatturazione dell'Account AWS, consulta la Guida per l'utente di AWS Billing.

Per domande relative a fatturazione, account ed eventi AWS, contatta AWS Support.

Termini e concetti relativi alle autorizzazioni verificate di Amazon

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
- Politiche soddisfatte
- Differenze tra autorizzazioni verificate e 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 gli ID delle 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.

Politiche soddisfatte

Le politiche soddisfatte sono le politiche che corrispondono ai parametri della richiesta di autorizzazione.

Differenze tra autorizzazioni verificate e 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 ne supporta solo unonamespace in uno schemadefinito in un archivio di politiche.
- La non consente di creare unnamespacecon i seguenti valori:`aws`,`amazon`, oppure`cedar`.

Supporto per modelli di policy

Sia Verified Permissions che Cedar consentono di inserire i segnaposto nell'ambito solo per`principal`e`resource`. Tuttavia, Verified Permissions richiede anche che né`principal`e`resource`non sono vincolati.

La seguente politica è valida in Cedar ma viene rifiutata da Verified Permissions perché`principal`non è vincolato.

```
permit(principal, action == Action::"view", resource == ?resource);
```

Entrambi gli esempi seguenti sono validi sia in Cedar che in Verified Permissions perché entrambi`principal`e`resource`hanno dei vincoli.

```
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 le proprietà.

Supporto per tipi di estensione

La non consente di modificare Cedartipi di estensionenelle politiche, ma attualmente non supporta la loro inclusione nella definizione di uno schema o come parte di`entities`parametro del`IsAuthorized`e`IsAuthorizedWithToken`operazioni.

I tipi di estensione includono il punto fisso (`decimal`) e indirizzo IP (`ipaddr`) tipi di dati.

Formato Cedar JSON per entità

Al momento, Verified Permissions richiede che l'utente trasmetta l'elenco delle entità da prendere in considerazione in una richiesta di autorizzazione utilizzando la struttura definita perEntitiesDefinition,

che è una serie diEntityItemelementi. Le autorizzazioni verificate attualmente non supportano il passaggio dell'elenco di entità da considerare in una richiesta di autorizzazione inFormato Cedar JSON. Per requisiti specifici di formattazione delle entità da utilizzare nelle autorizzazioni verificate, consultaFormattazione delle entità in Amazon Verified Permissions.

Definizione di 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 automaticamente aggiunti all'elenco di entità a cui si passa come parametro al`IsAuthorizedoIsAuthorizedWithToken`operazioni.

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 Verified Autorizzations imponga alcuni limiti di lunghezza e dimensione che non sono rilevanti per Cedar.

Oggetto	Limite di autorizzazioni verificate (in byte)	Limite Cedar
La dimensione della polizza[1]	10.000	Nessuno
Descrizione della politica in linea	150	La non si applica al cedro
La dimensione del modello di policy	10.000	Nessuno
Dimensione dello schema	10.000	Nessuno

Oggetto	Limite di autorizzazioni verificate (in byte)	Limite Cedar
Tipo di entità	200	Nessuno
ID Policy	64	Nessuno
ID del modello di policy	64	Nessuno
ID entità	200	Nessuno
ID del Policy Store	64	La non si applica al cedro

[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 nel policy store. 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.

Guida introduttiva alle autorizzazioni verificate

Usa questo tutorial per iniziare a usare Amazon Verified Permissions.

Argomenti

Registrarsi per creare un Account AWS

Se non disponi di un Account AWS, completa la procedura seguente per crearne uno.

Per registrarsi a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Durante la registrazione di un Account AWS, viene creato un Utente root dell'account AWS. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWSnell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente amministrativo e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

Al termine del processo di registrazione, riceverai un'e-mail di conferma da AWS. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Creazione di un utente amministratore

Dopo aver effettuato la registrazione di un Account AWS, proteggi Utente root dell'account AWS, abilita AWS IAM Identity Center e crea un utente amministratore in modo da non utilizzare l'utente root per le attività quotidiane.

Protezione dell'Utente root dell'account AWS

1. Accedi alla AWS Management Console come proprietario dell'account scegliendo Utente root e immettendo l'indirizzo email del Account AWS. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Accesso come utente root della 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'IAMutente.

Creazione di un utente amministratore

1. Abilita IAM Identity Center

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

   Per un tutorial sull'utilizzo di IAM Identity Center directory come origine di identità, consulta Configure user access with the default IAM Identity Center directory nella Guida per l'utente di AWS IAM Identity Center.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

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

IAMpolitiche per le autorizzazioni verificate

Verified Permissions gestisce le autorizzazioni degli utenti all'interno dell'applicazione. Affinché l'applicazione possa chiamare le API Verified Permissions o AWS Management Console 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 policy basate su identità sono documenti di policy di autorizzazione JSON che puoi collegare a un'identità come un 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 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

Action	Descrizione
CreatePolicyStore	Azione per creare un nuovo archivio di politiche .
DeletePolicyStore	Azione per eliminare un archivio delle politiche.
ListPolicyStores	Azione per elencare tutti gli archivi delle politiche inAccount AWS.
CreatePolicy	Azione per creare una policy Cedar in un policy store. È possibile creare una politica statica o una politica collegata a un modello di politica.
DeletePolicy	Azione per eliminare una policy da un policy store.
GetPolicy	Azione per recuperare informazioni su una politica specificata.
ListPolicies	Azione per elencare tutte le politiche in un archivio di politiche.
IsAuthorized	Azione per ottenere una risposta di autorizzazione basata sui parametri descritti nella richiesta di autorizzazione.

Esempio IAM di politica per l'autorizzazione all' CreatePolicy azione:

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

Crea il tuo primo archivio di norme sulle autorizzazioni verificate

Quando accedi alla console Verified Permissions per la prima volta, puoi scegliere come creare il tuo primo policy store e la tua prima policy Cedar. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In. Nella home page della console, seleziona il servizio Amazon Verified Permissions. Scegliere Iniziare.

Creazione di un archivio di policy di esempio

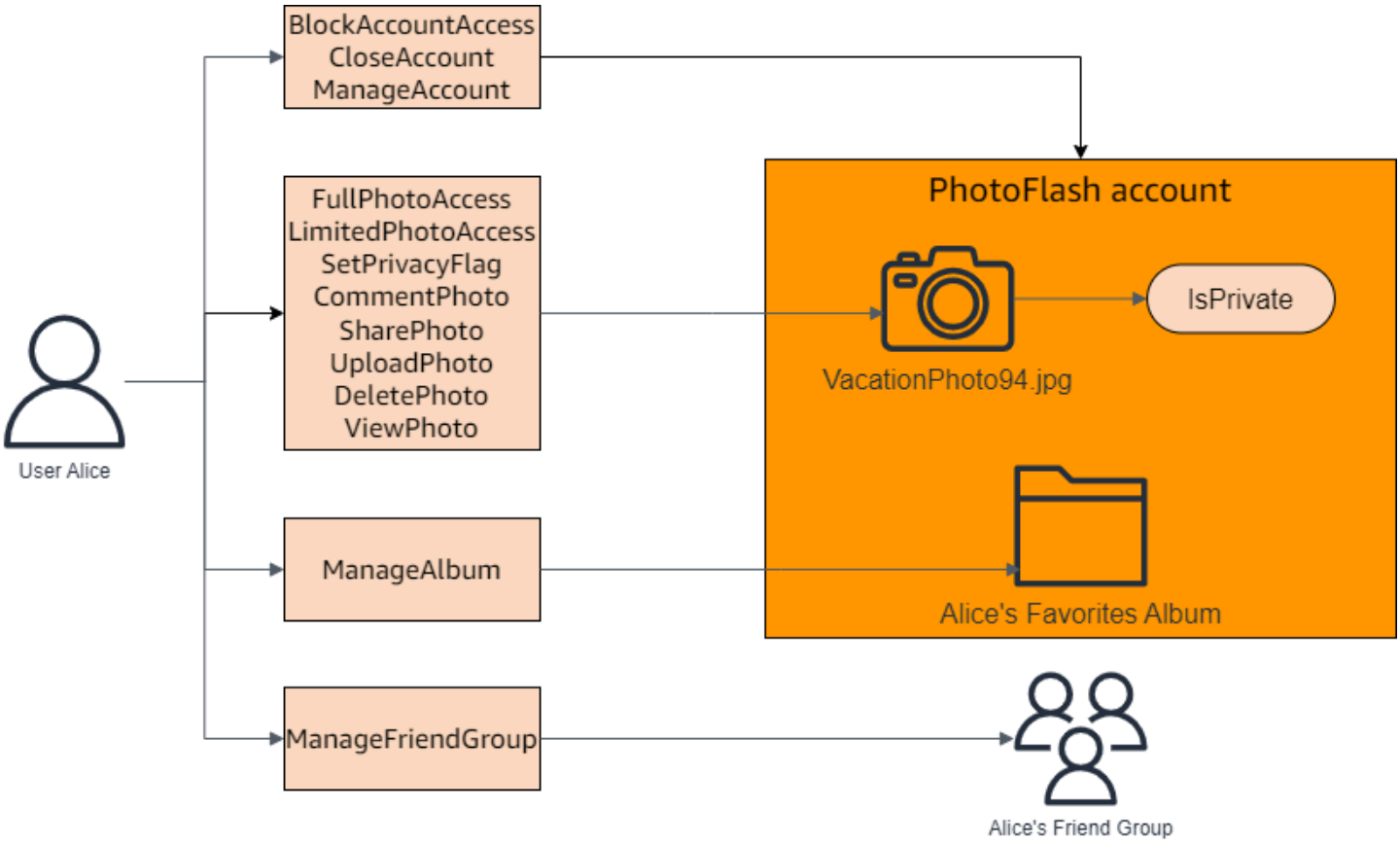
Se è la prima volta che utilizzi le autorizzazioni verificate, ti consigliamo di utilizzare uno degli archivi di policy di esempio per acquisire familiarità con il funzionamento delle autorizzazioni verificate. Gli archivi di policy di esempio forniscono policy e uno schema predefiniti.

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

1. Nella sezione Metodo di configurazione, selezionare Sample policy store.
2. Nella sezione Progetto di esempio, scegli il tipo di applicazione di esempio per le autorizzazioni verificate da utilizzare. Per questo tutorial, scegli il PhotoFlashpolicy store.
3. Un namespace per lo schema del tuo policy store di esempio viene generato automaticamente in base al progetto di esempio che hai scelto.
4. Scegli Crea archivio di politiche.

   Il tuo policy store viene creato con policy, modelli di policy e uno schema per il policy store di esempio.

Il diagramma seguente illustra le relazioni tra le azioni di PhotoFlash esempio del Policy Store e i tipi di risorse a cui si applicano.



Creazione di politiche collegate a modelli per un archivio di policy di esempio

L'archivio PhotoFlash di policy di esempio include policy, modelli di policy e uno schema. È possibile creare policy collegate ai modelli in base ai modelli di policy inclusi nel policy store di esempio.

Per creare policy collegate a modelli per il policy store di esempio

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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 con la descrizione Concedi l'accesso completo alle foto condivise non private, quindi scegli Avanti.

5. Per Principal, inserisci`PhotoFlash::User::"Alice"`. Per Risorsa, immetter`PhotoFlash::Album::"Bob-Vacation-Album"`.
6. Scegli Crea politica collegata al modello.

   La nuova politica collegata al modello viene visualizzata in Politiche.
7. Crea un'altra policy collegata al modello per l'archivio di policy di esempio. PhotoFlash Scegli Crea policy, quindi scegli Crea policy collegata al modello.
8. Scegli il pulsante di opzione accanto al modello di policy con la descrizione Concedi accesso limitato alle foto condivise non private, quindi scegli Avanti.
9. Per Principal, inserisci`PhotoFlash::FriendGroup::"MySchoolFriends"`. Per Risorsa, immetter`PhotoFlash::Album::"Alice's favorite album"`.
10. Scegli Crea politica collegata al modello.

    La nuova politica collegata al modello viene visualizzata in Politiche.

Testeremo le nuove politiche collegate ai modelli nella prossima sezione del tutorial. Per ulteriori esempi di valori per cui è possibile creare una politica collegata a un modello, consulta. PhotoFlash PhotoFlashesempi di policy collegati a modelli

Test di un esempio di policy store

Dopo aver creato il tuo archivio di politiche di esempio e le politiche collegate al modello, puoi testare le politiche statiche di esempio Verified Permissions e le tue nuove politiche collegate al modello eseguendo una richiesta di autorizzazione simulata utilizzando il banco di prova Verified Permissions.

A seconda di quando è stato creato il policy store di esempio, i modelli di policy potrebbero differire dai riferimenti in questa procedura. Prima di iniziare questa parte del tutorial, verificate di avere tutti i modelli di policy che seguono nel vostro policy store di PhotoFlash esempio. Se la tua politica non è in linea con queste politiche, modifica le politiche esistenti o crea un nuovo archivio di politiche dall'opzione PhotoFlashSample project.

Concedi l'accesso completo alle foto condivise non private

```
permit (
    principal in ?principal,
    action in PhotoFlash::Action::"FullPhotoAccess",
    resource in ?resource
)
```

```
when { resource.IsPrivate == false };
```

Concedi un accesso limitato alle foto condivise non private

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

Per testare alcuni esempi di policy store

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Test bench.
3. Scegli la modalità Visual.
4. Nella sezione Principal, scegli PhotoFlash: :User tra i tipi principali del tuo schema. Digitate un identificatore per l'utente nella casella di testo. Ad esempio, `Alice`.
5. Non scegliete Aggiungi un genitore come principale.
6. Per l'attributo Account: Entity, assicurati che l'entità PhotoFlash: :Account sia selezionata. Digita un identificatore per l'account. Ad esempio, `Alice-account`.
7. Nella sezione Risorse, scegli il tipo di risorsa PhotoFlash: :Photo. Digitate un identificatore per la foto nella casella di testo. Ad esempio, `photo.jpeg`.
8. Scegli Aggiungi un genitore e scegli PhotoFlash: :Account per il tipo di entità. Digita lo stesso identificatore per l'account principale per la foto che hai specificato nel campo Account: Entità per l'utente. Ad esempio, `Alice-account`.
9. Nella sezione Azione, scegli PhotoFlash: :Action:» ViewPhoto "dall'elenco delle azioni valide.
10. Nella sezione Entità aggiuntive, scegli Aggiungi questa entità per aggiungere l'entità dell'account suggerita.
11. Scegli Esegui richiesta di autorizzazione nella parte superiore della pagina per simulare la richiesta di autorizzazione per le politiche Cedar nell'archivio delle politiche di esempio. Il banco di prova dovrebbe mostrare la decisione di consentire la richiesta.

La tabella seguente fornisce valori aggiuntivi per il principale, la risorsa e l'azione che è possibile testare con il banco di prova Verified Permissions. La tabella include la decisione relativa alla

richiesta di autorizzazione basata sulle politiche statiche incluse nel policy store di PhotoFlash esempio e sulle politiche collegate al modello create nella sezione precedente.

Valore principale	Conto principale: valore dell'entità	Valore della risorsa	Valore principale della risorsa	Action	Decisione di autorizzazione
PhotoFlash: :Utente | Alice	PhotoFlash: :Conto | Conto Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Conto | Conto BOB	PhotoFlash: :Azione::» "ViewPhoto	Rifiuta
PhotoFlash: :Utente | Alice	PhotoFlash: :Conto | Conto Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Conto | Conto Alice	PhotoFlash: :Azione::» "ViewPhoto	Consenso
PhotoFlash: :Utente | Alice	PhotoFlash: :Conto | Conto Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash: :Album | Bob-Vacation-Album	PhotoFlash: :Azione::» "ViewPhoto	Consenso
PhotoFlash: :Utente | Alice	PhotoFlash: :Conto | Conto Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash: :Album | Bob-Vacation-Album	PhotoFlash: :Azione::» "DeletePhoto	Rifiuta
PhotoFlash: :Utente | Alice	PhotoFlash: :Conto | Conto Alice	PhotoFlash: :Photo | Bob-photo .jpeg,: Boolean | true IsPrivate	PhotoFlash: :Album | Album Bob-Vacation	PhotoFlash: :Azione::» "ViewPhoto	Rifiuta
PhotoFlash: :Utente | Jane, PhotoFlash:: | FriendGroup	PhotoFlash: :Conto | Jane - Conto	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | L'album preferito di Alice	PhotoFlash: :Azione::» "ViewPhoto	Consenso

Valore principale	Conto principale: valore dell'entità	Valore della risorsa	Valore principale della risorsa	Action	Decisione di autorizzazione
MySchoolFriends					
PhotoFlash: :Utente | Jane, PhotoFlash:: | FriendGroup MySchoolFriends	PhotoFlash: :Conto | Jane - Conto	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | L'album preferito di Alice	PhotoFlash: :Azione::» "DeletePhoto	Rifiuta

Archivi di policy di Amazon Verified Permissions

Un archivio di policy è un contenitore per policy e modelli di policy. Ogni policy store contiene uno schema utilizzato per convalidare i criteri aggiunti al policy store. Si consiglia 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.

Si consiglia di utilizzare namespace per le entità Cedar nei propri 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. **::** Verified Permissions supporta uno spazio dei nomi per archivio di politiche. Per ulteriori informazioni, consulta Namespaces nella Cedar Policy Language Reference Guide.

Creazione di archivi di policy per le autorizzazioni verificate

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

- Configurazione guidata: definirai un tipo di risorsa con azioni valide e un tipo principale prima di creare la prima politica.
- Esempio di archivio delle politiche: scegli un esempio di archivio delle politiche di progetto predefinito. Ti consigliamo questa opzione se stai imparando a conoscere le autorizzazioni verificate e desideri visualizzare e testare politiche di esempio.
- 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 sezione Metodo di configurazione, scegli Configurazione guidata.

2. Inserire 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 Weather updates.
3. Nella sezione Dettagli, digitate un Namespace per lo schema.
4. Seleziona Avanti.
5. Nella finestra Tipo di risorsa, digita un nome per il tipo di risorsa.
6. (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. Scegliete se ogni attributo è obbligatorio. 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 di risorsa, scegli Rimuovi accanto all'attributo.
7. 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. Per rimuovere un'azione che è stata aggiunta per il tipo di risorsa, scegli Rimuovi accanto all'azione.
8. Nel campo Nome del tipo principale, digita il nome di un tipo di principale che utilizzerà le azioni specificate per il tipo di risorsa.
9. Seleziona Avanti.
10. 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. Digita il nome dell'attributo e scegli un tipo di attributo per ogni attributo del principale. Verified Permissions 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 utenti di Amazon Cognito a cui connetterti. Scegli Connetti. Per ulteriori informazioni, consulta Authorization with Amazon Verified Permissions nella Amazon Cognito Developer Guide.
11. Seleziona Avanti.
12. Nella sezione Dettagli della politica, digita una descrizione facoltativa della politica per la tua prima politica Cedar.

13. 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.
    - Scegli Tutti i mandanti per applicare la politica a tutti i mandanti del tuo archivio polizze.
14. 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.
    - Scegli Tutte le risorse per applicare la politica a tutte le risorse del tuo archivio delle politiche.
15. 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.
16. Consulta la politica nella sezione Anteprima della politica. Scegli Crea archivio di politiche.

Sample policy store

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

1. Nella sezione Metodo di configurazione, selezionare 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. Esempi di politiche collegate a modelli per Autorizzazioni verificate, archivi di policy di esempio

Empty policy store

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

1. Nella sezione Metodo di configurazione, selezionare 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 per le autorizzazioni verificate di Amazon eCreazione di policy collegate a modelli.

AWS CLI

Per creare un archivio delle politiche vuoto utilizzandoAWS 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 per le autorizzazioni verificate di Amazon.
> - Per aggiungere modelli di policy, consultaCreazione di modelli di policy.

```
$ 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.

Cambio degli archivi di policy per le autorizzazioni verificate

AWS Management Console

Per passare da un archivio di norme all'altro o creare archivi di norme aggiuntivi

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli l'interruttore accanto a Current policy store.
3. È possibile passare da un archivio di policy esistente all'altro o creare archivi di policy aggiuntivi.
   - Per cambiare policy store, scegli l'ID del policy store a cui passare.
   - Per creare un nuovo policy store, scegli Crea nuovo policy store. Segui le istruzioni in Creazione di archivi di policy per le autorizzazioni verificate.

AWS CLI

Per cambiare archivio delle politiche o creare archivi di policy aggiuntivi

AWS CLINon mantiene un archivio di politiche «predefinito». Invece, la maggior parte dei AWS CLI comandi utilizza il `--policy-store-id` per specificare quale archivio di politiche utilizzare per ogni comando.

Per creare un nuovo archivio delle politiche, utilizzare il create-policy-storecomando.

Eliminazione degli archivi di norme sulle autorizzazioni verificate

AWS Management Console

Per eliminare un archivio delle politiche

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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.

AWS CLI

Per eliminare un archivio delle politiche

È possibile eliminare un policy store utilizzando l'`delete-policy-store`operazione.

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

Questo comando non produce alcun output in caso di successo.

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 ulteriori informazioni, vedere il formato dello schema Cedar nella Guida di riferimento del linguaggio di policy 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.

AWS Management Console

Per creare uno schema

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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 archivio delle politiche eseguendo un AWS CLI comando simile all'esempio seguente.

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 riportato è rifinito in righe per garantire la 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 in modalità visiva

Quando selezioni Schema nella console Autorizzazioni verificate, la modalità visiva mostra i tipi di entità e le azioni che compongono lo schema. In questa visualizzazione di primo livello o dall'interno dei dettagli di qualsiasi entità, puoi scegliere Modifica schema per iniziare ad aggiornare lo schema. La modalità visiva non è disponibile con alcuni formati di schema come i record annidati.

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 delle relazioni tra entità dello schema.

Diagramma delle azioni

La visualizzazione del diagramma delle azioni elenca i tipi di Principal configurati nell'archivio delle politiche, 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 all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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.

Modifica degli schemi in modalità JSON

Per modificare uno schema di autorizzazioni verificate in modalità JSON

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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. Seleziona Salvataggio delle modifiche.

Eliminazione di uno schema

AWS Management Console

Per eliminare uno schema di autorizzazioni verificate

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Schema.
3. Scegli Elimina schema.

AWS CLI

Per eliminare uno schema di autorizzazioni verificate

Non esiste un comando di eliminazione dello schema. È possibile eliminare lo schema in un archivio di politiche utilizzando il `put-schema` comando con uno schema vuoto nel `cedarJson` campo. Uno schema vuoto è rappresentato da un paio di parentesi graffe arricciate '{}'.

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

Modalità di convalida della politica 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 nel 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 la richiesta se la convalida fallisce.

AWS Management Console

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

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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. Seleziona Salvataggio delle 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 qualsiasi 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.

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:

- Utilizza valori come gli identificatori univoci universali (UUID) 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 == User: :"A1B2C3D4-E5F6-A1B2-C3D4-example11111",//alice
- Non includete informazioni di identificazione personale, riservate o sensibili come parte dell'identificatore univoco dei vostri mandanti o delle vostre risorse. Questi identificatori sono inclusi nelle voci di registro condivise nei percorsi. AWS CloudTrail

Formattazione delle entità in Amazon Verified Permissions

Amazon Verified Permissions utilizza il linguaggio delle policy Cedar per creare policy. La sintassi delle policy e i tipi di dati supportati corrispondono alla sintassi e ai tipi di dati descritti negli argomenti Basic policy building in Cedar e Data types supported by Cedar nella Cedar Policy Language

Reference Guide. Tuttavia, esistono differenze tra Verified Permissions e Cedar nella formattazione delle entità quando si effettua una richiesta di autorizzazione.

La formattazione JSON delle entità in Verified Permissions 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`

L'esempio seguente mostra come le entità in un elenco vengono formattate utilizzando Cedar.

```
[
  {
    "number": 1
  },
  {
    "sentence": "Here is an example sentence"
  },
  {
    "Question": false
```

```
  }
]
```

L'esempio seguente mostra come le stesse entità del precedente esempio di elenco Cedar sono formattate in Verified Permissions.

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

L'esempio seguente mostra come le entità Cedar sono formattate per la valutazione di una politica in una richiesta di autorizzazione.

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

L'esempio seguente mostra come le stesse entità dell'esempio precedente di Cedar sono formattate in 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": []
    }
]
```

Creazione di politiche statiche per le autorizzazioni verificate di Amazon

È possibile creare una politica statica Cedar per consentire o impedire ai principali di eseguire azioni specifiche su risorse specifiche per l'applicazione.

AWS Management Console

Per creare una politica statica

1. Apri la console delle autorizzazioni verificate all'[indirizzo https://console.aws.amazon.com/verifiedpermissions/](https://console.aws.amazon.com/verifiedpermissions/). 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.

4. Nella sezione Effetto della politica, scegli se la politica consentirà o proibirà quando una richiesta corrisponde alla politica.
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.
   - Scegliete 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 politica a tutte le azioni nel tuo archivio delle polizze.
8. Seleziona 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"
}
```

Modifica delle politiche statiche di Amazon Verified Permissions

Puoi modificare una politica statica Cedar esistente nel tuo policy store. È possibile aggiornare direttamente solo le politiche statiche. È possibile modificare solo alcuni elementi di una politica statica:

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

Non è possibile modificare questi elementi di una politica statica:

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

Per modificare una politica collegata al modello, è necessario invece aggiornare il modello. Per ulteriori informazioni, consulta Modifica dei modelli di policy.

AWS Management Console

Per modificare una politica statica

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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"
    }
}
```

Visualizzazione delle politiche

AWS Management Console

Per visualizzare le politiche relative alle autorizzazioni verificate

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy). Vengono visualizzate tutte le politiche che hai creato.
3. Scegli la casella di testo Cerca per filtrare le politiche per Principal o Resource.
4. Scegli il pulsante di opzione accanto a una politica per visualizzare i dettagli sulla politica, ad esempio quando la politica è stata creata, aggiornata e il contenuto della politica.
5. È possibile eliminare una politica selezionando il pulsante di opzione accanto a una politica e quindi scegliendo Elimina. Scegli Elimina politica per confermare l'eliminazione della politica.

AWS CLI

Per elencare tutte le politiche disponibili in un archivio di politiche

È possibile visualizzare l'elenco delle politiche utilizzando l'GetPolicyoperazione. L'esempio seguente recupera un elenco che include una politica statica e una politica collegata al modello.

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

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

Per visualizzare i dettagli di una singola politica

È possibile recuperare i dettagli di una politica utilizzando l'GetPolicyoperazione. L'esempio seguente recupera i dettagli di una policy collegata a un modello.

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

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

```
        "entityType": "Photo"
    }
}
```

Esempi di politiche di Amazon Verified Permissions

I seguenti esempi di policy relative alle autorizzazioni verificate si basano sullo schema definito per l'ipotetica applicazione richiamata, PhotoFlash descritta nella sezione Example schema della Cedar Policy Language Reference Guide. Per ulteriori informazioni sulla sintassi delle politiche Cedar, vedere Costruzione delle politiche di base in Cedar nella Cedar Policy Language Reference Guide.

Esempi di policy

- 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

Consente l'accesso a singole entità

Questo esempio mostra come è possibile creare una politica che `alice` consenta all'utente di visualizzare la foto`VacationPhoto94.jpg`.

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

Consente l'accesso a gruppi di entità

Questo esempio mostra come è possibile creare una politica che consenta a tutti i membri del gruppo `alice_friends` di visualizzare la foto`VacationPhoto94.jpg`.

```
permit(
  principal in Group::"alice_friends",
  action == Action::"view",
```

```
  resource == Photo::"VacationPhoto94.jpg"
);
```

Questo esempio mostra come è possibile creare una politica che `alice` consenta all'utente di visualizzare qualsiasi foto dell'album`alice_vacation`.

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

Questo esempio mostra come è possibile creare una politica che `alice` consenta all'utente di visualizzare, modificare o eliminare qualsiasi foto nell'album`alice_vacation`.

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

Questo esempio mostra come è possibile creare una politica che consenta le autorizzazioni per l'utente `alice` nell'album`alice_vacation`, dove si `admin` trova un gruppo definito nella gerarchia dello schema che contiene le autorizzazioni per visualizzare, modificare ed eliminare una foto.

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

Questo esempio mostra come è possibile creare una politica che consenta le autorizzazioni per l'utente `alice` nell'album`alice_vacation`, dove si `viewer` trova un gruppo definito nella gerarchia dello schema che contiene l'autorizzazione a visualizzare e commentare una foto. L'`edit`autorizzazione `alice` viene inoltre concessa all'utente mediante la seconda azione elencata nella politica.

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

```
  resource in Album::"alice_vacation"
)
```

Consente l'accesso a qualsiasi entità

Questo esempio mostra come è possibile creare una politica che consenta a qualsiasi principale autenticato di visualizzare l'album`alice_vacation`.

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

Questo esempio mostra come è possibile creare una politica che consenta all'utente di `alice` elencare tutti gli album dell'`jane`account, elencare le foto in ogni album e visualizzare le foto nell'account.

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

Questo esempio mostra come è possibile creare una politica che `alice` consenta all'utente di eseguire qualsiasi azione sulle risorse dell'album`jane_vaction`.

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

Consente l'accesso agli attributi di un'entità (ABAC)

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. Le autorizzazioni verificate consentono di allegare attributi a principali, azioni e risorse. È quindi possibile fare riferimento a questi attributi all'interno delle `unless` clausole `when` e delle politiche che valutano gli attributi dei principali, delle azioni e delle risorse che costituiscono il contesto della richiesta.

Gli esempi seguenti utilizzano gli attributi definiti nell'applicazione ipotetica denominata PhotoFlash descritta nella sezione Example schema della Cedar Policy Language Reference Guide.

Questo esempio mostra come è possibile creare una politica che consenta a qualsiasi dirigente del `HardwareEngineering` dipartimento con un livello di mansione maggiore o uguale a 5 di visualizzare ed elencare le foto nell'album. `device_prototypes`

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

Questo esempio mostra come è possibile creare una politica che `alice` consenta all'utente di visualizzare qualsiasi risorsa di tipo di file`JPEG`.

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

Le azioni hanno attributi di contesto. È necessario passare questi attributi in una richiesta `context` di autorizzazione. Questo esempio mostra come è possibile creare una politica che `alice` consenta all'utente di eseguire qualsiasi `readOnly` azione. È inoltre possibile impostare una `appliesTo` proprietà per le azioni nello schema. Ciò specifica le azioni valide per una risorsa quando si desidera garantire che, ad esempio, gli utenti possano tentare di `ViewPhoto` autorizzare solo una risorsa di tipo. `PhotoFlash::Photo`

```
permit(
  principal == PhotoFlash::User::"alice",
  action,
  resource
```

```
) when {
    context has readOnly &&
    context.readOnly == true
};
```

Un modo migliore per impostare le proprietà delle azioni nello schema, tuttavia, consiste nel disporle in gruppi di azioni funzionali. Ad esempio, è possibile creare un'azione denominata `ReadOnlyPhotoAccess` e `PhotoFlash::Action::"ViewPhoto"` impostata come membro `ReadOnlyPhotoAccess` come gruppo di azioni. Questo esempio mostra come è possibile creare una politica che conceda ad Alice l'accesso alle azioni di sola lettura in quel gruppo.

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

Questo esempio mostra come è possibile creare una politica che consenta a tutti i responsabili di eseguire qualsiasi azione sulle risorse per le quali dispongono di attributi. `owner`

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

Questo esempio mostra come è possibile creare una politica che consenta a qualsiasi principale di visualizzare qualsiasi risorsa se l'`department`attributo del principale corrisponde all'`department`attributo della risorsa.

> **Note**
>
> Se un'entità non ha un attributo menzionato in una condizione politica, la politica verrà ignorata quando si prende una decisione di autorizzazione e la valutazione di tale politica avrà esito negativo per quell'entità. Ad esempio, a qualsiasi principale che non dispone di

un `department` attributo non può essere concesso l'accesso a nessuna risorsa in base a questa politica.

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

Questo esempio mostra come è possibile creare una politica che consenta a qualsiasi principale di eseguire qualsiasi azione su una risorsa se il principale è il responsabile `owner` della risorsa OPPURE se il principale fa parte del `admins` gruppo della risorsa.

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

Nega l'accesso

Se una politica `forbid` prevede l'effetto della politica, limita le autorizzazioni anziché concedere le autorizzazioni.

Important

Durante l'autorizzazione, se vengono applicate sia una policy che una `permit` `forbid` policy, questa ha la precedenza. `forbid`

Gli esempi seguenti utilizzano gli attributi definiti nell'applicazione ipotetica denominata PhotoFlash descritta nella sezione Example schema della Cedar Policy Language Reference Guide.

Questo esempio mostra come è possibile creare una politica che `alice` impedisca all'utente di eseguire tutte le azioni tranne `readOnly` che su qualsiasi risorsa.

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

Questo esempio mostra come è possibile creare una politica che neghi l'accesso a tutte le risorse che hanno un `private` attributo a meno che il principale non disponga dell'`owner`attributo per la risorsa.

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

Modelli di policy per Autorizzazioni verificate da Amazon

Puoi creare modelli di policy Cedar in Verified Permissions per definire una regola di controllo degli accessi per il tuo sistema. I modelli di policy sono policy Cedar con segnaposti per`principal`,`resource`o entrambi. I modelli di policy consentono di definire una policy una sola volta e quindi di associarla a più principi e risorse. Gli aggiornamenti al modello di policy si riflettono su tutti i principali e le risorse che utilizzano il modello. Per ulteriori informazioni, consulta la paginaModelli di policynella Guida di riferimento al linguaggio delle politiche Cedar.

Ti consigliamo di utilizzare modelli di policy per creare policy che possono essere condivise in tutta l'applicazione. Ad esempio, è possibile creare un modello di policy per un editor che fornisca autorizzazioni di lettura, modifica e commento per il principale e la risorsa che utilizzano il modello di policy.

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

Quando un principale viene designato come editor di una risorsa, l'applicazione può creare un'istanza di una policy utilizzando il modello per fornire le autorizzazioni al principale per eseguire le azioni di lettura, modifica e commento sulla risorsa.

Creazione di modelli di policy

AWS Management Console

Per creare un modello di policy

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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

Per 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"
}
```

Creazione di policy collegate a modelli

È possibile creare policy collegate a un modello per collegarsi a un modello di policy. 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.

AWS Management Console

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

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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 == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
```

```
);
```

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

```
{
    "templateLinked": {
        "policyTemplateId": "pt-4651be67-c128-4d22-8e67-9b068980c631",
        "principal": {
            "entityType": "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": "User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "Photo"
    }
}
```

Modifica dei modelli di policy

AWS Management Console

Per modificare i modelli di policy

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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 scegliendo 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 aggiornare 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"
}
```

Esempi di politiche collegate a modelli per Autorizzazioni verificate, archivi di policy di esempio

Quando si crea un policy store in Verified Permissions utilizzando il metodo Sample policy store, quest'ultimo viene creato con policy, modelli di policy e uno schema predefiniti per il progetto di esempio 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.

PhotoFlashesempi di policy collegati a modelli

Questo esempio 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"
 );
```

Questo esempio 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"
 );
```

Questo esempio 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"
 );
```

Questo esempio 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"
 );
```

Questo esempio mostra come è possibile 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"
 );
```

Questo esempio mostra come è possibile creare una politica collegata al modello che utilizza il modello di politica Blocca utente da un account.

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

DigitalPetStore

Il policy store DigitalPetStore 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 di policy collegati a modelli

Questo esempio mostra come è possibile creare una policy collegata a un 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"
);
```

Questo esempio 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"
);
```

Utilizzo di Amazon Verified Permissions con provider di identità

Puoi utilizzare i provider di identità OpenID Connect (OIDC) (IdPs) con autorizzazioni verificate per trasmettere gli attributi utente da utilizzare come principali nelle politiche di autorizzazione verificata.

Utilizzo di Amazon Cognito e fonti di identità

Una fonte di identità è una raccolta di informazioni sugli utenti a cui fa riferimento un provider di identità per semplificare le richieste di autorizzazione ai tuoi archivi di polizze. È possibile creare una fonte di identità per fornire informazioni sui principali delle applicazioni di autorizzazione verificata. Puoi specificare l'ID del Regione AWS pool di utenti di Amazon Cognito e il tipo principale delle tue fonti di identità. 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.

Le dichiarazioni JSON Web Token (JWT) del pool di utenti contengono attributi utente. Puoi aggiungere attestazioni personalizzate per informare le decisioni di autorizzazione prese da Verified Permissions. Le rivendicazioni relative ai token di identità includono `cognito:username` `ecognito:groups`. Per ulteriori informazioni, consulta Usare i token con i pool di utenti nella Amazon Cognito Developer Guide.

> **Important**
>
> Sebbene sia possibile revocare i token Amazon Cognito prima della scadenza, i JWT sono considerati risorse stateless autonome con 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.

Puoi anche aggiungere attributi personalizzati al tuo pool di utenti. Gli attributi personalizzati richiedono il prefisso `custom:` per distinguerli dagli attributi standard. Ad esempio, è possibile

aggiungere un `custom:joblevel` attributo a un pool di utenti. Per ulteriori informazioni, consulta Attributi personalizzati nella Amazon Cognito Developer Guide.

Quando scrivi policy Cedar in Verified Permissions utilizzando claim e attributi del pool di utenti di Amazon Cognito che contengono `:` un carattere, devi farvi riferimento nelle policy Cedar con un punto `.` () anziché due punti `:` () per rispettare la sintassi della policy Cedar. Ad esempio, è necessario modificare rispettivamente e in e. `cognito:username` `cognito:groups` `cognito.username` `cognito.groups`

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

Questo esempio mostra come creare una policy che faccia riferimento ai pool di utenti di Amazon Cognito, attestazioni personalizzate corrispondenti a un principale.

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

Per ulteriori informazioni, consulta Authorization with Amazon Verified Permissions nella Amazon Cognito Developer Guide.

Collaborazione con altri provider di identità

Puoi estrarre gli attributi delle tue entità da un JSON Web Token (JWT) di qualsiasi provider OpenID Connect (OIDC) (IdP) e analizzarli in Autorizzazioni verificate.

Questo esempio mostra come è possibile chiamare le autorizzazioni verificate da un IdC conforme a OIDC[1].

```
async function authorizeUsingJwtToken(jwtToken) {

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

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

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

    return authResult;

}

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

```
        return;
    }
    if (Array.isArray(value)) {
      var attibuteItem = [];
      value.forEach((item) => {
        attibuteItem.push({
          string: item,
        });
      });
      attributes[key] = {
        set: attibuteItem,
      };
    } else if (typeof value === 'string') {
      attributes[key] = {
        string: value,
      }
    } else if (typeof value === 'bigint' || typeof value ==='number') {
        attributes[key] = {
            long: value,
          }
    } else if (typeof value === 'bigint' || typeof value ==='number') {
        attributes[key] = {
            long: value,
          }
    } else if (typeof value === 'boolean') {
        attributes[key] = {
            boolean: value,
       }
    }

  });

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

[1] Questo esempio di codice utilizza la libreria per verificare i JWT firmati da OIDC Compatible aws-jwt-verify. IdPs

Creazione di fonti di identità Amazon Verified Permissions

La procedura seguente aggiunge una fonte di identità Amazon Cognito al tuo policy store.

> **Note**
>
> Le fonti di identità non sono disponibili nel riquadro di navigazione a sinistra finché non hai creato un policy store. 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 del policy store. Per integrare le identità di Amazon Cognito con lo schema del tuo Policy Store, devi impostare il tipo di entità principale su un'entità Policy Store supportata.

AWS Management Console

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

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. Scegli il tuo negozio di polizze.
2. Nel riquadro di navigazione a sinistra, scegli Identity sources.
3. Scegli Crea fonte di identità.
4. Nella sezione Dettagli del pool di utenti di Cognito, seleziona Regione AWS e digita l'ID del pool di utenti per la tua origine di identità.
5. Nella sezione Dettagli del principale, scegli il tipo di Principal per l'origine dell'identità. Le identità dei pool di utenti Amazon Cognito connessi verranno mappate sul tipo principale selezionato.
6. Nella sezione Convalida dell'applicazione client, scegli se convalidare gli ID delle applicazioni client.

   - Per convalidare gli ID delle applicazioni client, scegli Accetta solo token con gli ID delle applicazioni client corrispondenti. 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 gli ID delle applicazioni client se non desiderate convalidare gli ID delle applicazioni client.

7. Scegli Crea origine di identità.
8. Prima di poter fare riferimento agli attributi che estrai dai token di identità o di accesso nelle tue politiche Cedar, devi aggiornare lo schema per rendere Cedar consapevole del tipo di principale creato dalla tua 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 Amazon Cognito agli attributi principali di Cedar, consulta. Mappatura dei token Amazon Cognito allo schema di autorizzazioni verificate

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 utenti di Amazon Cognito.

Il `config.txt` file seguente contiene i dettagli del pool di utenti di Amazon Cognito da utilizzare con il parametro --configuration nel comando. `create-identity-source`

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

Comando:

```
$ 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"
}
```

Prima di poter fare riferimento agli attributi che estrai dai token di identità o di accesso nelle tue policy Cedar, devi aggiornare lo schema per rendere Cedar consapevole del tipo di principale creato dalla tua 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 Amazon Cognito agli attributi principali di Cedar, consulta. Mappatura dei token Amazon Cognito allo schema di autorizzazioni verificate

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 identità di Amazon Verified Permissions

AWS Management Console

Per aggiornare la fonte di identità di un pool di utenti di Amazon Cognito

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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. Nella sezione 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. Nella sezione 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 sezione Convalida dell'applicazione client, scegli se convalidare gli ID delle applicazioni client.

    - Per convalidare gli ID delle applicazioni client, scegli Accetta solo token con gli ID delle applicazioni client corrispondenti. 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 gli ID delle applicazioni client se non desiderate convalidare gli ID delle applicazioni client.

8. Seleziona Salvataggio delle modifiche.
9. 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 la fonte di identità di un pool di utenti di Amazon Cognito

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

Il `config.txt` file seguente contiene i dettagli del pool di utenti di Amazon Cognito da utilizzare con il parametro --configuration nel comando. `create-identity-source`

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

Comando:

```
$ 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"
}
```

Se modifichi il tipo principale per la fonte di identità, devi aggiornare lo schema in modo che rifletta correttamente il tipo principale aggiornato.

Mappatura dei token Amazon Cognito allo schema di autorizzazioni verificate

I token di identità di Amazon Cognito hanno quattro tipi di attributi che influiscono sulla denominazione degli attributi nello schema di autorizzazioni verificate. Per utilizzare Amazon Cognito come fonte di identità nel tuo archivio di policy di Autorizzazioni verificate e scrivere policy utilizzando gli attributi del token Amazon Cognito che verranno verificati utilizzando l'isAuthorizedWithazione Token API, devi aggiungere gli attributi di Amazon Cognito al tuo schema.

- Affermazioni specifiche di Amazon Cognito con il prefisso dello spazio dei `cognito` nomi, ad esempio. `cognito:username`

> **Note**
>
> L'attestazione specifica di `cognito:groups` Amazon Cognito non è attualmente mappata negli schemi di autorizzazioni verificate. I gruppi Amazon Cognito possono utilizzare identificatori di cui non è garantito l'univocità e che possono essere riutilizzati. Includerli in una decisione di autorizzazione potrebbe introdurre rischi per la sicurezza dell'applicazione.

- Affermazioni personalizzate con il prefisso `custom` dello spazio dei nomi, ad esempio. `custom:employmentStoreCode`
- Affermazioni standard che non hanno un prefisso dello spazio dei nomi, ad esempio. `email`
- Affermazioni transitorie che vengono aggiunte durante la personalizzazione del token, ad esempio, e. `tenant` `department` `clearance`

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.

L'esempio seguente di token di identità ha ciascuno dei quattro tipi di attributi. Include l'attestazione specifica di Amazon Cognito`cognito:username`, l'attestazione personalizzata`custom:employmentStoreCode`, l'attestazione standard e l'attestazione `email` 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_wBEbEZKaX",
    "cognito:username": "alice",
    "custom:employmentStoreCode": "petstore-dallas",
    "origin_jti": "5b9f50a3-05da-454a-8b99-b79c2349de77",
    "aud": "52n97d5afhfiuXXX",
    "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 un archivio di identità e lo associ al tuo pool di utenti Amazon Cognito, specifichi il tipo di principale che Cedar genera quando un token viene passato a una `IsAuthorizedWithToken` richiesta di valutazione. Le tue politiche possono quindi testare gli attributi di quel principale come parte della valutazione della richiesta. È necessario definire quel tipo principale e gli attributi a cui si desidera poter fare riferimento dalle politiche Cedar.

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 in modalità JSON. Se la configurazione dell'origine dell'identità specifica il tipo principale`CognitoUser`, puoi includere qualcosa di simile al seguente esempio per rendere tali attributi disponibili a Cedar.

```
"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
            }
        }
    }
}
```

Dopo aver aggiornato lo schema in modo che rifletta gli attributi di Amazon Cognito, puoi creare policy che fanno riferimento agli attributi.

```
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"
};
```

Mappatura dei token di accesso di Amazon Cognito

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

- `client_id`rappresenta l'ID client dell'applicazione. Questo valore può essere utilizzato negli scenari di machine-to-machine autorizzazione per rappresentare l'identità della macchina.
- `scope`è l'ambito standard OAuth 2.0 che rappresenta gli ambiti autorizzati per il portatore del token.

Un token di accesso Amazon Cognito viene mappato su un oggetto di contesto quando viene passato a Autorizzazioni verificate. 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_wBEbEZKaX",
    "client_id": "52n97d5afhfiu1c4di1k5m8f60",
    "origin_jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "event_id": "bda909cb-3e29-4bb8-83e3-ce6808f49011",
    "token_use": "access",
    "scope": "aws.cognito.signin.user.admin",
    "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 in modalità JSON.

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

Dopo aver aggiornato lo schema in modo che rifletta gli attributi di Amazon Cognito, puoi creare policy che fanno riferimento agli attributi.

```
permit(principal, action in [MyApplication::Action::"Read",
 MyApplication::Action::"GetStoreInventory"], resource)
when {
    context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
    context.token.scope like "*aws.cognito.signin.user.admin*"
};
```

Progettazione di un modello di autorizzazione per la tua applicazione

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, acquisendo una chiara comprensione di ciò che gli utenti finali dovrebbero vedere quando gestiscono le autorizzazioni nell'interfaccia utente dell'applicazione. Quindi, lavorate 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? Hanno relazioni tra loro? Ad esempio, i file si trovano all'interno di una cartella?
- 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?
- Le autorizzazioni devono essere ereditate da più risorse, ad esempio i file che ereditano le autorizzazioni da una cartella principale?
- 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 influenzano ciò che gli utenti vedono 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.

Questa sezione fornisce indicazioni generali su come affrontare l'esercizio di progettazione, gli aspetti a cui prestare attenzione e una raccolta di best practice per utilizzare con successo le autorizzazioni verificate.

Oltre alle linee guida qui presentate, ricordatevi di prendere in considerazione le migliori pratiche contenute nella guida di riferimento linguistica Cedar Policy Language.

Argomenti

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 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.

Concentrati sulle tue risorse, non sulle operazioni delle API

Nella maggior parte delle applicazioni rivolte ai consumatori, le autorizzazioni sono modellate in base alle risorse supportate dall'applicazione. 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 `action`in Cedar chiamato`createThing`. Funziona in molte situazioni e semplifica la comprensione delle autorizzazioni. Per invocare il`createThing`operazione, è necessario`createThing`autorizzazione dell'azione. Sembra semplice, vero?

Tuttavia, questo approccio incentrato sulle API può essere tutt'altro che ottimale, perché le API sono semplicemente un proxy di ciò che i clienti stanno veramente cercando di proteggere: i dati e le risorse sottostanti. Se più API 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 atutti e dueoperazioni. 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 ignorare la semantica dell'API e concentrarti invece sui dati e sulle risorse sottostanti e sulle relative operazioni di associazione. L'applicazione di questo approccio all'esempio dell'appartenenza al gruppo porterebbe a un'autorizzazione astratta, ad esempio`viewGroupMembership`, che ciascuna delle tre operazioni API deve consultare.

Nome API	Autorizzazioni	
`listMembersOfGroup`	richiede`viewGroupMembership`	autorizzazione del gruppo
`listGroupMembershipsForUser`	richiede`viewGroupMembership`	autorizzazione dell'utente
`isUserInGroups`	richiede`viewGroupMembership`	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.

L'autorizzazione composta è normale

L'autorizzazione composta si verifica quando un'attività di un singolo utente, ad esempio fare clic su un pulsante nell'interfaccia dell'applicazione, richiede più query di autorizzazione individuali per determinare se tale attività è consentita. Ad esempio, lo spostamento di un file in una nuova directory in un file system potrebbe richiedere tre diverse autorizzazioni: la possibilità di eliminare un file dalla directory di origine, la possibilità di aggiungere un file alla directory di destinazione ed eventualmente la possibilità di toccare il file stesso (a seconda dell'applicazione).

Se non conosci la progettazione di un modello di autorizzazione, potresti pensare che ogni decisione di autorizzazione debba essere risolvibile in un'unica richiesta di autorizzazione. Ma ciò può portare a modelli eccessivamente complessi e a dichiarazioni politiche complicate. In pratica, l'utilizzo di autorizzazioni composte può essere utile per aiutarvi a produrre un modello di autorizzazione più semplice. Una misura di un modello di autorizzazione ben progettato è che, quando le singole azioni

sono sufficientemente scomposte, le operazioni composte, come lo spostamento di un file, possono essere rappresentate da un'aggregazione intuitiva di primitive.

Un'altra situazione in cui si verifica l'autorizzazione composta è quando più parti sono coinvolte nel processo di concessione di un'autorizzazione. Prendi in considerazione un elenco organizzativo in cui gli utenti possono essere membri di gruppi. Un approccio semplice consiste nel concedere al proprietario del gruppo il permesso di aggiungere chiunque. Tuttavia, cosa succede se desideri che i tuoi utenti acconsentano innanzitutto all'aggiunta? Ciò introduce un accordo di stretta di mano in cui sia l'utente che il gruppo devono acconsentire all'appartenenza. A tale scopo, è possibile introdurre un'altra autorizzazione associata all'utente e specificare se l'utente può essere aggiunto a qualsiasi gruppo o a un gruppo particolare. Quando un chiamante tenta successivamente di aggiungere membri a un gruppo, l'applicazione deve applicare entrambe le autorizzazioni: che il chiamante sia autorizzato ad aggiungere membri al gruppo specificato e che il singolo utente aggiunto disponga delle autorizzazioni necessarie. QuandoNLe strette di mano esistono, è comune osservarleNrichieste di autorizzazione complesse per far rispettare ogni parte dell'accordo.

Se vi trovate di fronte a un problema di progettazione in cui sono coinvolte più risorse e non vi è chiaro come modellare le autorizzazioni, può essere un segno che avete uno scenario di autorizzazione composto. In questo caso, è possibile trovare una soluzione scomponendo l'operazione in più controlli di autorizzazione individuali.

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.

1. Un archivio di politiche 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 crea un volume relativamente più elevato di richieste di autorizzazione che potrebbero avere un impatto sulla fattura. AWS Quindi, come dovresti progettare il tuo approccio? Le seguenti sono condizioni comuni che potrebbero contribuire alla vostra strategia di autorizzazione multi-tenant Verified Permissions.

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

I modelli di policy e uno schema di policy store aggiungono un livello di sovraccarico di progettazione e manutenzione in ogni archivio delle politiche.

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 pertinenti.

Considerazione	Livello di impegno in un archivio di policy condiviso	Livello di impegno negli archivi di policy per inquilino
Isolamento delle politiche degli inquilini	Medio. Must include tenant identifiers in policies and authorization requests.	Basso. Isolation is default behavior. Tenant-specific policies are inaccessible to other tenants.
Flusso di autorizzazione	Basso. All queries target one policy store.	Medio. Must maintain mappings between each tenant and their policy store ID.
Gestione dei modelli e degli schemi	Basso. Must make one schema work for all tenants.	Elevato. Schemas and templates might be less complex individually, but changes require more coordination and complexity.
Gestione delle politiche globali	Bassa. All policies are global and can be centrally updated.	Elevato. You must add global policies to each policy store in onboarding. Replicate global policy updates between many policy stores.
Disimbarco dell'inquilino	Medio. Must identify and delete only tenant-specific policies.	Basso. Delete the policy store.
Quote di risorse di servizio	Elevato. Tenants share resource quotas that affect policy stores like schema size, policy size per resource, and	Bassa. Each tenant has dedicated resource quotas.

identity sources per policy store.

Come scegliere

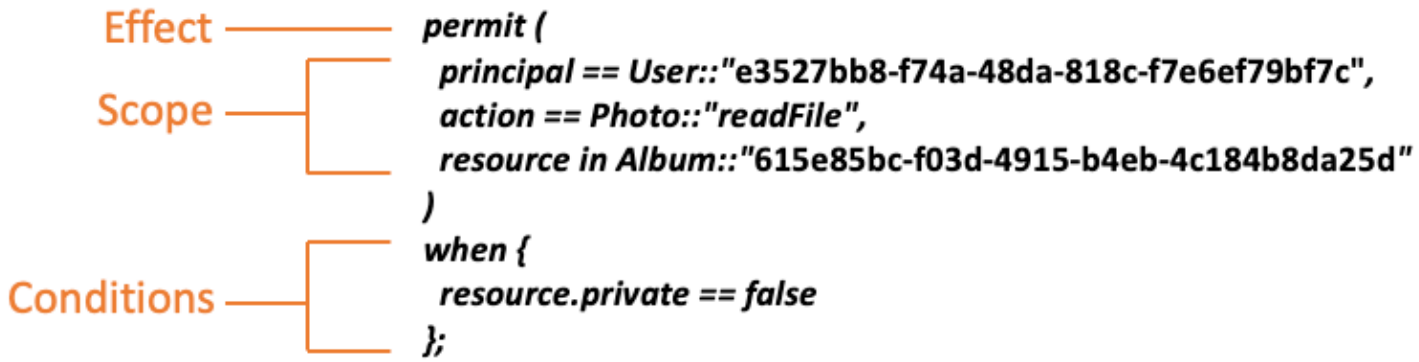
Ogni applicazione multi-tenant è diversa. Confrontate attentamente i due approcci e le relative considerazioni prima di prendere una decisione architettonica.

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.

Se possibile, compila l'ambito della policy

L'ambito della politica è la parte di una dichiarazione politica di Cedar dopo il`permit`o`forbid`parole chiave e tra le parentesi di apertura.



Si consiglia di inserire i valori per`principal`e`resource`quando possibile. Ciò consente a Verified Permissions di indicizzare le politiche per un recupero più efficiente e quindi migliora le prestazioni.

Se devi concedere le stesse autorizzazioni a molti principali o risorse diversi, ti consigliamo di utilizzare un modello di policy e di collegarlo a ciascuna coppia di principali/risorse.

Evita di creare un'unica politica di grandi dimensioni che contenga elenchi di principali e risorse in unwhenclausola. Ciò potrebbe comportare limiti di scalabilità o sfide operative. Ad esempio, per aggiungere o rimuovere un singolo utente da un elenco di grandi dimensioni all'interno di una policy, è necessario leggere l'intera policy, modificare l'elenco, scrivere la nuova policy per intero e gestire gli errori di concorrenza se un amministratore sovrascrive le modifiche di un altro. Al contrario, utilizzando molte autorizzazioni dettagliate, aggiungere o rimuovere un utente è semplice come aggiungere o rimuovere la singola politica che lo riguarda.

Ogni risorsa vive in un contenitore

Quando si progetta un modello di autorizzazione, ogni azione deve essere associata a una particolare risorsa. Con un'azione come`viewFile`, la risorsa a cui è possibile applicarlo è intuitiva: un singolo file o forse una raccolta di file all'interno di una cartella. Tuttavia, un'operazione come`createFile`è meno intuitiva. Quando si modella la capacità di creare un file, a quale risorsa si applica? Non può essere il file stesso, perché il file non esiste ancora.

Questo è un esempio del problema generalizzato della creazione di risorse. La creazione di risorse è un problema di avvio. Deve esserci un modo per consentire a qualcosa di avere il permesso di creare risorse anche quando non esistono ancora risorse. La soluzione è riconoscere che ogni risorsa deve esistere all'interno di un contenitore ed è il contenitore stesso a fungere da punto di ancoraggio per le autorizzazioni. Ad esempio, se nel sistema esiste già una cartella, la possibilità di creare un file può essere modellata come un'autorizzazione per quella cartella, poiché quella è la posizione in cui sono necessarie le autorizzazioni per creare un'istanza della nuova risorsa.

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

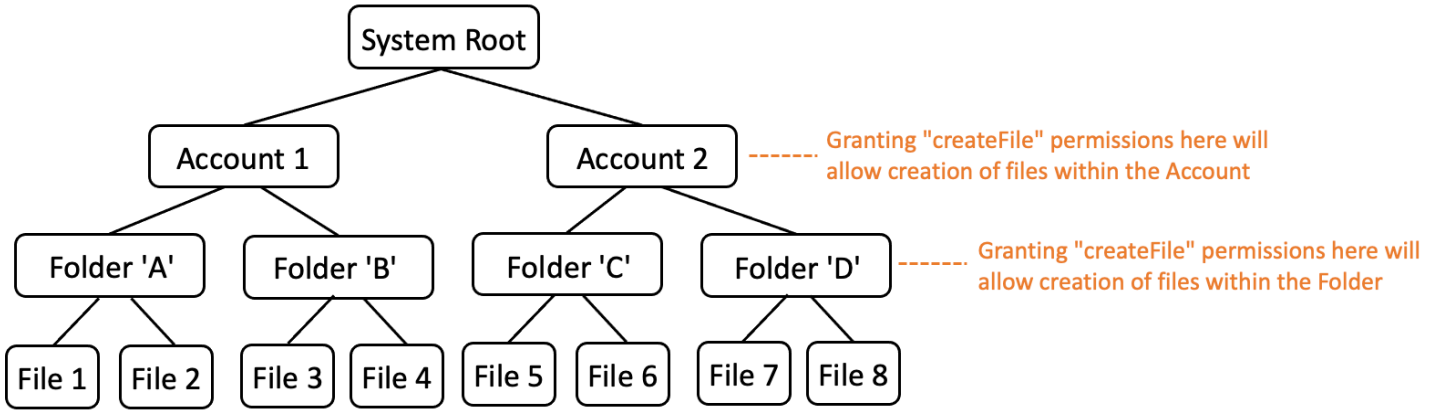
Ma cosa succede se non esiste alcuna cartella? Forse si tratta di un account cliente nuovo di zecca in un'applicazione in cui non esistono ancora risorse. In questa situazione, esiste ancora un contesto che può essere compreso in modo intuitivo chiedendo: dove può il cliente creare nuovi file? Non vuoi che siano in grado di creare file all'interno di un account cliente casuale. Piuttosto, esiste un contesto implicito: il confine dell'account del cliente. Pertanto, l'account stesso rappresenta il contenitore per la

creazione di risorse e questo può essere modellato in modo esplicito in una politica simile all'esempio seguente.

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

Tuttavia, cosa succede se non esistono nemmeno account? Potresti scegliere di progettare il flusso di lavoro di registrazione dei clienti in modo che crei nuovi account nel sistema. In tal caso, avrai bisogno di un contenitore che contenga il confine più esterno entro il quale il processo può creare gli account. Questo contenitore a livello di radice rappresenta il sistema nel suo insieme e potrebbe avere un nome simile a «root di sistema». Tuttavia, la decisione se è necessario e come chiamarlo spetta all'utente, proprietario dell'applicazione.

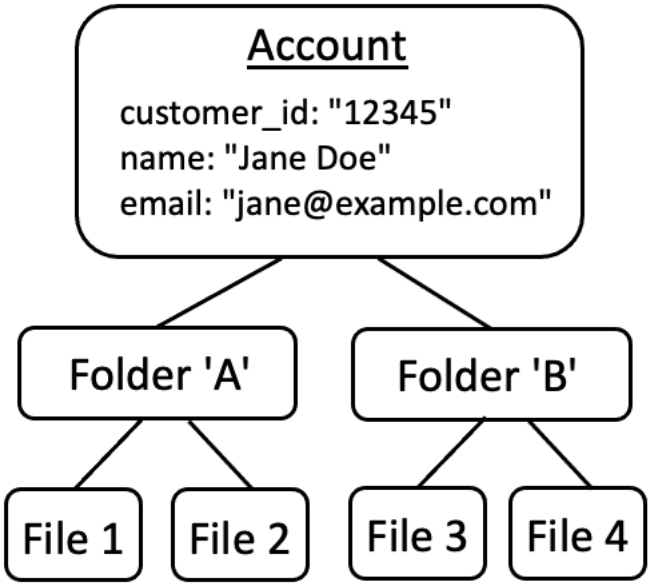
Per questa applicazione di esempio, la gerarchia dei contenitori risultante apparirebbe quindi come segue:



Questo è un esempio di gerarchia. Anche altre sono valide. La cosa da ricordare è che la creazione di risorse avviene sempre nel contesto di un contenitore di risorse. Questi contenitori possono essere impliciti, come i limiti di un account, e può essere facile trascurarli. Durante la progettazione del modello di autorizzazione, assicuratevi di prendere nota di questi presupposti impliciti in modo che possano essere documentati e rappresentati formalmente nel modello di autorizzazione.

Separare i principali dai contenitori di risorse

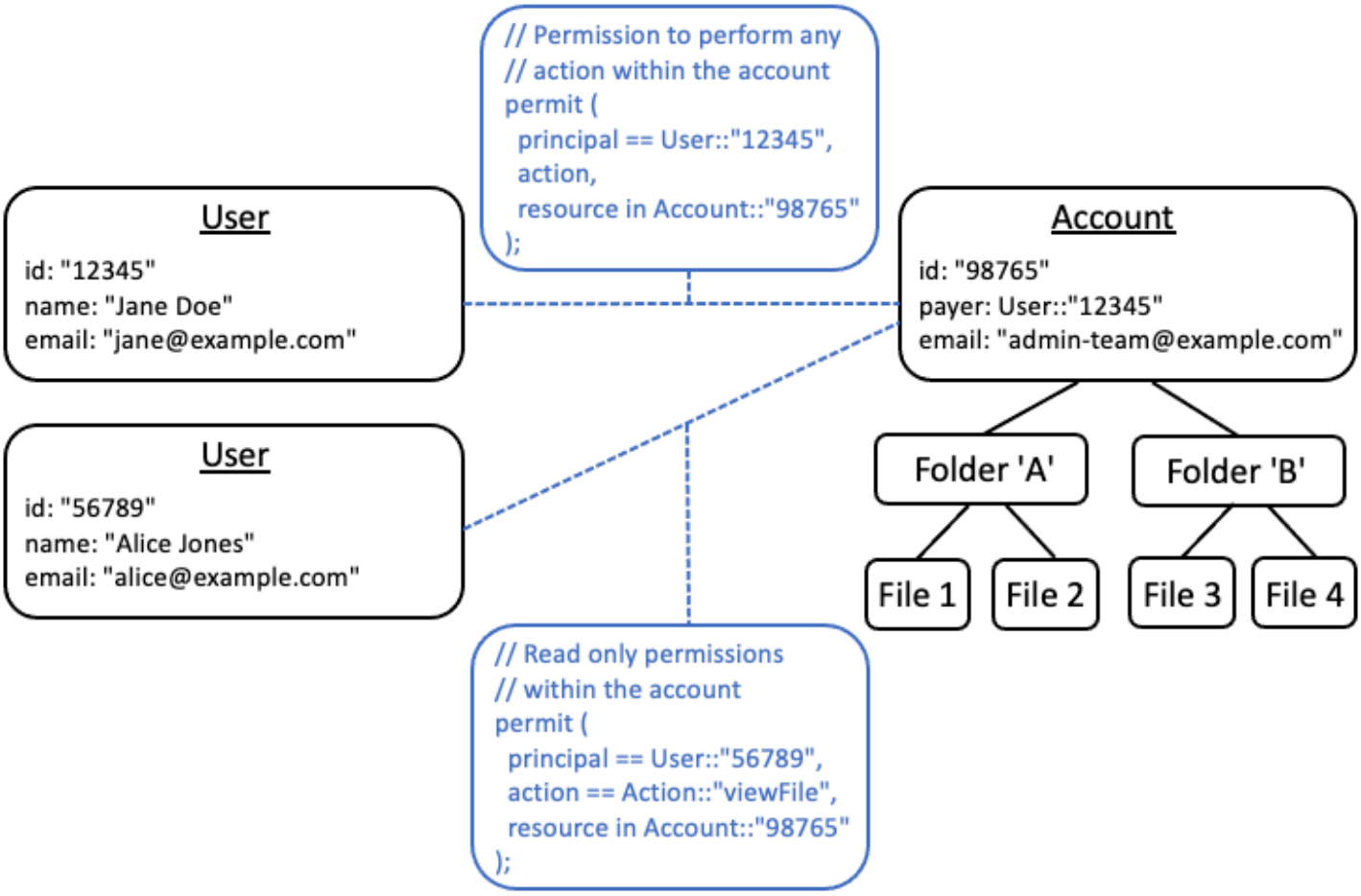
Quando si progetta una gerarchia di risorse, una delle inclinazioni più comuni, in particolare per le applicazioni rivolte ai consumatori, è quella di utilizzare l'identità utente del cliente come contenitore per le risorse all'interno di un account cliente.



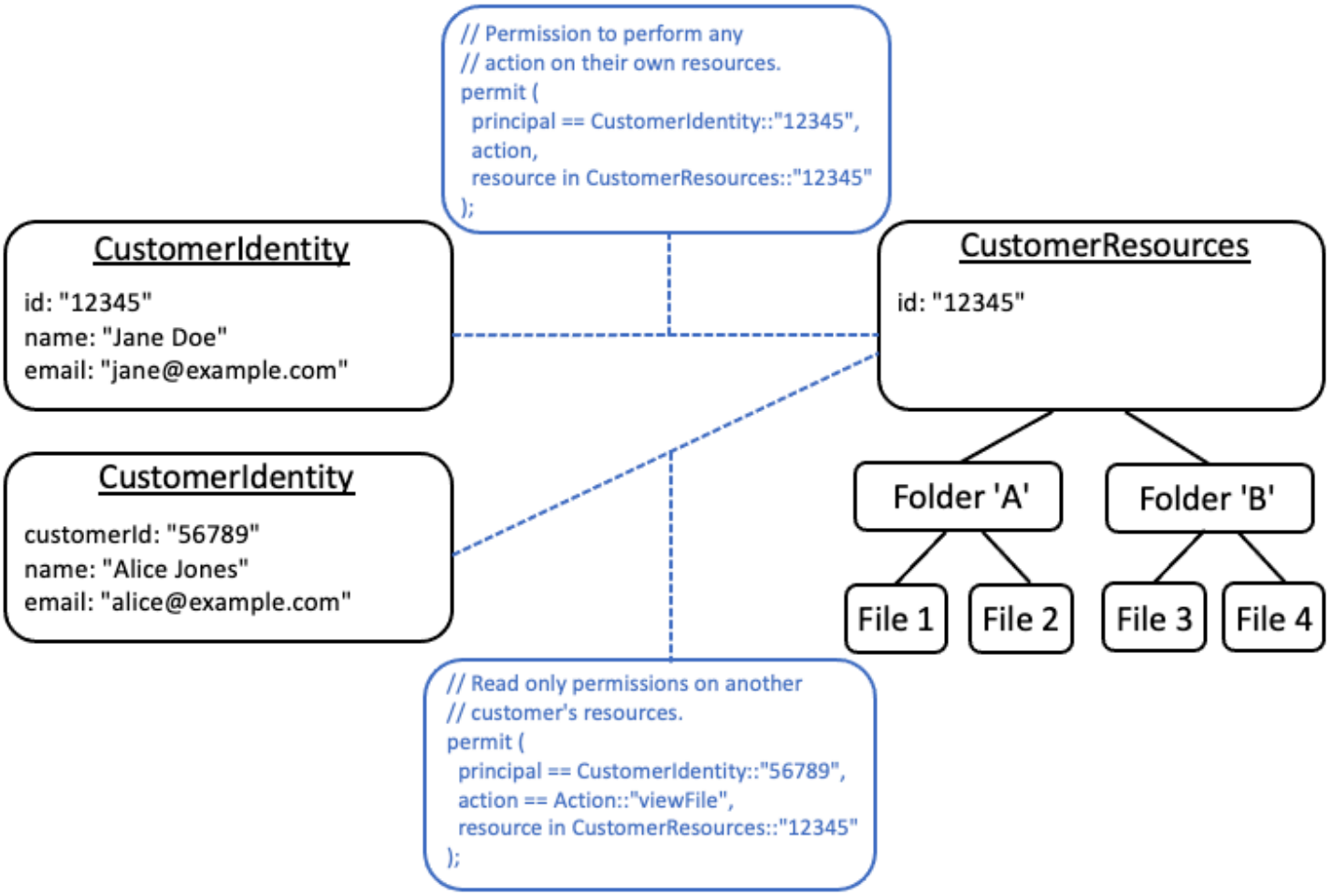
Ti consigliamo di considerare questa strategia come un anti-pattern. Questo perché c'è una tendenza naturale nelle applicazioni più avanzate a delegare l'accesso ad altri utenti. Ad esempio, potresti scegliere di introdurre account «familiari», in cui altri utenti possono condividere le risorse dell'account. Analogamente, i clienti aziendali a volte desiderano designare più membri della forza lavoro come operatori di parti dell'account. Potrebbe inoltre essere necessario trasferire la proprietà di un account a un altro utente o unire le risorse di più account.

Quando un'identità utente viene utilizzata come contenitore di risorse per un account, gli scenari precedenti diventano più difficili da realizzare. Ancora più allarmante, se ad altri viene concesso l'accesso al contenitore dell'account con questo approccio, potrebbe inavvertitamente essere autorizzato a modificare l'identità dell'utente stesso, ad esempio cambiando l'e-mail o le credenziali di accesso di Jane.

Pertanto, quando possibile, un approccio più resiliente consiste nel separare i principali dai contenitori di risorse e modellare la connessione tra di essi utilizzando concetti come «autorizzazioni di amministratore» o «proprietà».



Se disponi di un'applicazione esistente che non è in grado di perseguire questo modello disaccoppiato, ti consigliamo di imitarlo il più possibile durante la progettazione di un modello di autorizzazione. Ad esempio, un'applicazione che possiede un solo concetto denominato`Customer`che incapsula l'identità dell'utente, le credenziali di accesso e le risorse di sua proprietà, potrebbe mapparle a un modello di autorizzazione che contenga un'entità logica per`Customer Identity`(contenente nome, e-mail, ecc.) e un'entità logica separata per`Customer Resources`o`Customer Account`, fungendo da nodo principale per tutte le risorse che possiedono. Entrambe le entità possono condividere le stesse informazioni`Id`, ma con un diverso`Type`.



Non incorporare le autorizzazioni negli attributi

Gli attributi vengono utilizzati al meglio comeingressoalla decisione di autorizzazione. Non utilizzare gli attributi per rappresentare le autorizzazioni stesse, ad esempio dichiarando un attributo denominato «PermittedFolders» su un utente:

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

```
}
```

E, successivamente, utilizzando l'attributo all'interno di una politica:

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

Questo approccio trasforma quello che altrimenti sarebbe un semplice modello di autorizzazione, in cui uno specifico principale ha accesso a una cartella specifica, in un modello di controllo degli accessi basato sugli attributi (ABAC) con i relativi compromessi. Uno di questi compromessi è che diventa più difficile determinare rapidamente chi ha l'autorizzazione a utilizzare una risorsa. Nell'esempio precedente, per determinare chi ha accesso a una particolare cartella, è necessario esaminare ogni utente per verificare se quella cartella è elencata nei relativi attributi, tenendo conto in particolare del fatto che esiste una politica che concede l'accesso quando lo fa.

Un altro rischio legato a questo approccio sono i fattori di scalabilità quando le autorizzazioni sono raggruppate in un'unica soluzione`User`record. Se l'utente ha accesso a molte cose, la dimensione cumulativa delle`User`il record crescerà e forse si avvicinerà al limite massimo del sistema che memorizza i dati.

Si consiglia invece di rappresentare questo scenario utilizzando più politiche individuali, magari utilizzando modelli di policy per ridurre al minimo la ripetizione.

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

permit (
    principal == User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
    action == Action::"readFile",
    resource in Folder::"661817a9-d478-4096-943d-4ef1e082d19a"
```

```
);

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

Le autorizzazioni verificate possono gestire in modo efficiente molte politiche individuali e dettagliate durante la valutazione delle autorizzazioni. Modellare le cose in questo modo è più gestibile e verificabile nel tempo.

Preferisci le autorizzazioni granulari nel modello e le autorizzazioni aggregate nell'interfaccia utente

Una strategia di cui i progettisti spesso si pentono in seguito è la progettazione di un modello di autorizzazione con azioni molto ampie, come`ReadeWrite`, e rendendosi conto in seguito che sono necessarie azioni più dettagliate. L'esigenza di una maggiore granularità può essere determinata dal feedback dei clienti per controlli di accesso più granulari o dai revisori di conformità e sicurezza che incoraggiano le autorizzazioni con privilegi minimi.

Se le autorizzazioni granulari non sono definite in anticipo, può essere necessaria una conversione complicata per modificare il codice dell'applicazione e le istruzioni politiche in autorizzazioni più dettagliate per l'utente. Ad esempio, il codice dell'applicazione che in precedenza autorizzava un'azione granulare del corso dovrà essere modificato per utilizzare le azioni granulari. Inoltre, le politiche dovranno essere aggiornate per riflettere la migrazione:

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

Per evitare questa migrazione costosa, è meglio definire in anticipo autorizzazioni dettagliate. Tuttavia, ciò può comportare un compromesso se gli utenti finali sono successivamente costretti

a comprendere un numero maggiore di autorizzazioni granulari, soprattutto se la maggior parte dei clienti sarebbe soddisfatta di controlli granulari come`ReadeWrite`. Per ottenere il meglio da entrambi i mondi, puoi raggruppare le autorizzazioni granulari in raccolte predefinite come`ReadeWrite`utilizzando meccanismi come modelli di policy o gruppi di azioni. Utilizzando questo approccio, i clienti vedono solo le autorizzazioni dettagliate del corso. Ma dietro le quinte, hai reso la tua applicazione a prova di futuro modellando le autorizzazioni granulari del corso come una raccolta di azioni granulari. Quando i clienti o i revisori lo richiedono, le autorizzazioni granulari possono essere esposte.

Prendi in considerazione altri motivi per richiedere l'autorizzazione

Di solito associamo i controlli di autorizzazione alle richieste degli utenti. Il controllo è un modo per determinare se l'utente è autorizzato a eseguire tale richiesta. Tuttavia, è possibile utilizzare i dati di autorizzazione anche per influenzare la progettazione dell'interfaccia dell'applicazione. Ad esempio, potreste voler visualizzare una schermata iniziale che mostri un elenco delle sole risorse a cui l'utente finale può accedere. Quando si visualizzano i dettagli di una risorsa, è possibile che l'interfaccia mostri solo le operazioni che l'utente può eseguire su quella risorsa.

Queste situazioni possono introdurre dei compromessi nel modello di autorizzazione. Ad esempio, il forte affidamento sulle politiche di controllo degli accessi basate sugli attributi (ABAC) può rendere più difficile rispondere rapidamente alla domanda «chi ha accesso a cosa?» Questo perché per rispondere a questa domanda è necessario esaminare ogni regola relativa a ogni principale e risorsa per determinare se esiste una corrispondenza. Di conseguenza, un prodotto che deve essere ottimizzato per elencare solo le risorse accessibili dall'utente potrebbe scegliere di utilizzare un modello di controllo degli accessi basato sui ruoli (RBAC). Utilizzando RBAC, può essere più semplice iterare su tutte le politiche associate a un utente per determinare l'accesso alle risorse.

Banco di prova

Il banco di prova delle autorizzazioni verificate consente di 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.

Visual mode

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

Per testare le politiche in modalità Visual

1. Apri la console delle autorizzazioni verificate all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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 genitore 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 padre 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 all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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.

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 contro Amazon Verified Permissions. Per implementare l'autorizzazione Verified Permissions, devi combinare la configurazione delle policy 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 policy store.

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 raccolgono informazioni da API esterne. 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.

Operazioni API 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 convalida le entità contenute nella richiesta rispetto allo schema del policy store. Verified Permissions valuta quindi la richiesta rispetto a tutte le politiche nell'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 (JWT) di Amazon Cognito. Verified Permissions funziona direttamente con Amazon Cognito come fonte di identità nel tuo archivio di polizze. Verified Permissions inserisce gli attributi relativi al principale contenuto nella tua richiesta utilizzando le attestazioni contenute nell'ID degli utenti o nei token di accesso.

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 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. Verified Permissions restituisce una decisione per ogni richiesta.

Verifica del tuo modello di autorizzazione

Per comprendere l'effetto della decisione di autorizzazione di Verified Permissions quando distribuisci la tua applicazione, puoi valutare le tue politiche man mano che le sviluppi con Banco di prova e con le richieste dell'API REST HTTPS alle autorizzazioni verificate. Il test bench è uno strumento AWS Management Console per valutare le richieste e le risposte di autorizzazione nell'archivio delle politiche.

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 all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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 dà come risultato una `ALLOW` decisione. Nota la politica 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 all'indirizzo https://console.aws.amazon.com/verifiedpermissions/. 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 con app e SDK AWS

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 identità Amazon Cognito 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 negli SDK. AWS Per includere le autorizzazioni verificate in un'app, integra l'AWSSDK per la lingua scelta nel pacchetto dell'app.

Per saperne di più e scaricare gli AWS SDK, consulta Strumenti per. Amazon Web Services

Di seguito sono riportati i collegamenti alla documentazione relativa alle risorse relative alle autorizzazioni verificate in vari AWS SDK.

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

Il seguente AWS SDK for 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

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. 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": {
      "MfaAuthorized": {
         "boolean": true
      },
      "AccountCodes": {
         "set": [
            {
               "long": 111122223333
            },
            {
               "long": 444455556666
            },
            {
               "long": 123456789012
            }
         ]
      },
      "UserAgent": {
         "string": "My UserAgent 1.12"
      },
      "RequestedOrderCount": {
         "long": 4
      },
      "NetworkInfo": {
         "record": {
```

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

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 dipendente`Bob`.

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.

Schema

Applica i seguenti aggiornamenti allo schema del 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": {
            "UserAgent": {
               "required": true,
               "type": "String"
            },
```

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

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

> **Note**
>
> L'editor visivo dello schema delle autorizzazioni verificate attualmente non supporta `commonType` i costrutti. Quando li aggiungi allo schema, non puoi più visualizzarlo in modalità visiva.

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

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, deve trovarsi in una sessione crittografata e il suo indirizzo IP deve iniziare con`192.0.2.`.
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": {
      "MfaAuthorized": {
       "boolean": true
      },
      "UserAgent": {
       "string": "My UserAgent 1.12"
      },
      "RequestedOrderCount":{
       "long": 4
      },
```

```
      "AccountCodes": {
       "set": [
        {"long": 111122223333},
        {"long": 444455556666},
        {"long": 123456789012}
        ]
      },
      "NetworkInfo": {
        "record": {
          "IPAddress": {"string": "192.0.2.178"},
          "Country": {"string": "United States of America"},
          "SSL": {"boolean": true}
        }
      },
      "approvedBy": {
        "entityIdentifier": {
         "entityId": "Bob",
         "entityType": "DigitalPetStore::User"
        }
      }
     }
   },
   "entities": {
     "entityList": [
       {
         "identifier": {
            "entityType": "DigitalPetStore::User",
            "entityId": "Alice"
         },
         "attributes": {
            "memberId": {
              "string": "801b87f2-1a5c-40b3-b580-eacad506d4e6"
              }
         },
         "parents": [
            {
            "entityType": "DigitalPetStore::Role",
            "entityId": "Customer"
            }
         ]
       },
       {
         "identifier": {
            "entityType": "DigitalPetStore::User",
```

```
            "entityId": "Bob"
          },
          "attributes": {
             "memberId": {
               "string": "49d9b81e-735d-429c-989d-93bec0bcfd8b"
               }
          },
          "parents": [
             {
             "entityType": "DigitalPetStore::Role",
             "entityId": "Employee"
             }
          ]
       },
       {
          "identifier": {
             "entityType": "DigitalPetStore::Order",
             "entityId": "1234"
          },
          "attributes": {
             "owner": {
               "entityIdentifier": {
                 "entityType": "DigitalPetStore::User",
                 "entityId": "Alice"
               }
            }
          },
          "parents": []
       }
     ]
   },
   "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Sicurezza nelle autorizzazioni verificate da Amazon

Per AWS, la sicurezza del cloud ha la massima priorità. In quanto cliente AWS, puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e l'utente. Il modello di responsabilità condivisa fa riferimento ad una sicurezza del cloud e nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che esegue i servizi AWS in Cloud AWS. AWS fornisce inoltre i servizi che è possibile utilizzare in modo sicuro. I revisori di terze parti testano regolarmente e verificano l'efficacia della nostra sicurezza nell'ambito dei Programmi di conformità AWS. Per ulteriori informazioni sui programmi di conformità che si applicano alle autorizzazioni verificate di Amazon, consultaAWSServizi inclusi nel programma Scope by Compliance.
- Sicurezza nel cloud: la tua responsabilità è determinata dal servizio AWS che utilizzi. L'utente è anche responsabile per altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda, le leggi e le normative applicabili.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi le autorizzazioni verificate. Gli argomenti seguenti mostrano come configurare le autorizzazioni verificate per soddisfare i tuoi obiettivi di sicurezza e conformità. Imparerai anche a usarne altriAWSservizi che ti aiutano a monitorare e proteggere le tue risorse relative alle autorizzazioni verificate.

Argomenti

Protezione dei dati nelle autorizzazioni verificate di Amazon

LaAWS modello di responsabilità condivisasi applica alla protezione dei dati nelle autorizzazioni verificate di Amazon. Come descritto in questo modello, AWS è responsabile della protezione

dell'infrastruttura globale che esegue tutto l'Cloud AWS. L'utente è responsabile di mantenere il controllo sui contenuti ospitati su questa infrastruttura. Questi contenuti comprendono la configurazione della protezione e le attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi 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 proteggereAccount AWScredenziali e configurazione di singoli utenti conAWS IAM Identity CenteroAWS Identity and Access Management(IAM). In questo modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere il suo lavoro.
- Ti consigliamo di proteggere i tuoi dati nei seguenti modi:
  - Utilizza l'autenticazione a più fattori (MFA) con ogni account.
  - Utilizza SSL/TLS per comunicare con le risorse AWS. È richiesto TLS 1.2.
  - Configura la registrazione delle API e delle attività degli utenti con AWS CloudTrail.
  - Utilizza le soluzioni di crittografia AWS, insieme a tutti i controlli di sicurezza di default all'interno dei Servizi AWS.
  - Utilizza i servizi di sicurezza gestiti avanzati, ad esempio Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
  - Se si richiedono moduli crittografici convalidati FIPS 140-2 quando si accede ad AWS tramite una CLI o un'API, utilizzare 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 identificative sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio un campo Name (Nome). Ciò include quando lavori con Autorizzazioni verificate o altroServizi AWSutilizzando la console, l'API,AWS CLI, oppureAWSSDK. I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i log di fatturazione o di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.
- I nomi delle tue azioni non devono includere informazioni sensibili.
- Ti consigliamo inoltre vivamente di utilizzare sempre identificatori univoci, non modificabili e non riutilizzabili per le tue entità (risorse e committenti). In un ambiente di test, puoi scegliere di utilizzare semplici identificatori di entità, come`janeobob`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 possono essere riutilizzati. Ti consigliamo di utilizzare valori

come identificatori univoci universali (UUID). Ad esempio, considera l'utente`jane`chi lascia l'azienda. Successivamente, permetti a qualcun altro di usare il nome`jane`. Quel nuovo utente accede automaticamente a tutto ciò che è concesso dalle politiche che ancora fanno riferimento`User::"jane"`. Autorizzazioni verificate e Cedar non possono distinguere tra il nuovo utente e l'utente precedente.

Questa guida si applica sia agli identificatori principali che a quelli delle risorse. Utilizza sempre identificatori 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 definiscano`Long`e`Decimal`i valori rientrano nell'intervallo valido di ogni tipo. Inoltre, assicurati che l'utilizzo di operatori aritmetici 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 Proibita potrebbe inaspettatamente non riuscire a bloccare l'accesso.

Crittografia dei dati

Amazon Verified Permissions crittografa automaticamente tutti i dati dei clienti, come le politiche, con unChiave gestita da AWS, pertanto l'uso di una chiave gestita dal cliente non è né necessario né supportato.

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 risorse. AWS IAMgli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse di Autorizzazioni verificate. IAMè un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Come funziona Amazon Verified Permissions con IAM
- Esempi di policy basate sull'identità 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 a seconda del lavoro svolto in Autorizzazioni verificate.

Utente del servizio: se utilizzi il servizio Autorizzazioni verificate per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità di Autorizzazioni verificate per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità delle Autorizzazioni verificate, consulta. Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Verified Permissions

Amministratore del servizio: se sei responsabile delle risorse relative alle autorizzazioni verificate presso la tua azienda, probabilmente hai pieno accesso alle autorizzazioni verificate. È tuo compito determinare a quali funzionalità e risorse di Autorizzazioni verificate devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere le nozioni di base di IAM. Per ulteriori informazioni su come la tua azienda può utilizzare le autorizzazioni IAM verificate, consulta. Come funziona Amazon Verified Permissions con IAM

IAMamministratore: se sei un IAM amministratore, potresti voler conoscere i dettagli su come scrivere politiche per gestire l'accesso alle autorizzazioni verificate. Per visualizzare esempi di politiche basate sull'identità delle autorizzazioni verificate che puoi utilizzare in, consulta. IAM Esempi di policy basate sull'identità per Amazon Verified Permissions

Autenticazione con identità

L'autenticazione è la procedura di accesso ad AWS con le credenziali di identità. Devi autenticarti (accedere aAWS) come utente IAM o assumendo un ruolo. Utente root dell'account AWS IAM

Puoi accedere ad AWS come identità federata utilizzando le credenziali fornite attraverso un'origine di identità. Gli utenti AWS IAM Identity Center (Centro identità IAM), l'autenticazione Single Sign-On (SSO) dell'azienda e le credenziali di Google o Facebook sono esempi di identità federate. Se accedi

come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Se accedi ad AWS tramite la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere alla AWS Management Console o al portale di accesso AWS. Per ulteriori informazioni sull'accesso ad AWS, consulta la sezione Come accedere al tuo Account AWS nella Guida per l'utente di Accedi ad AWS.

Se accedi ad AWS in modo programmatico, AWS fornisce un Software Development Kit (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le richieste utilizzando le tue credenziali. Se non utilizzi gli strumenti AWS, devi firmare le richieste personalmente. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta Firmare le richieste AWS API nella Guida per l'IAMutente.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. AWS consiglia ad esempio di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza dell'account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'AWS IAM Identity Centerutente e Utilizzo dell'autenticazione a più fattori (MFA) AWS nella Guida per l'utente. IAM

Utente root di un Account AWS

Quando crei un Account AWS, inizi con una singola identità di accesso che ha accesso completo a tutti i Servizi AWS e le risorse nell'account. Tale identità è detta utente root Account AWS ed è possibile accedervi con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per l'elenco completo delle attività che richiedono l'accesso come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'IAMutente.

Identità federata

Come best practice, richiedi agli utenti umani, compresi quelli che richiedono l'accesso di amministratore, di utilizzare la federazione con un provider di identità per accedere a Servizi AWSutilizzando credenziali temporanee.

Un'identità federata è un utente della directory degli utenti aziendali, un provider di identità Web, AWS Directory Service, la directory Identity Center o qualsiasi utente che accede ai Servizi AWS utilizzando le credenziali fornite tramite un'origine di identità. Quando le identità federate accedono agli Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. È possibile creare utenti e gruppi in IAM Identity Center oppure connettersi e sincronizzarsi con un gruppo di utenti e gruppi nell'origine di identità per utilizzarli in tutte le applicazioni e gli Account AWS. Per ulteriori informazioni sul Centro identità IAM, consulta Cos'è Centro identità IAM? nella Guida per l'utente di AWS IAM Identity Center.

Utenti e gruppi IAM

Un utente IAM è una identità all'interno del tuo Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta Ruotare regolarmente le chiavi di accesso per i casi d'uso che richiedono credenziali a lungo termine nella Guida per l'IAMutente.

Un IAMgruppo è un'identità che specifica una raccolta di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, puoi avere un gruppo denominato Amministratori IAM e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per saperne di più, consulta Quando creare un utente IAM (anziché un ruolo) nella Guida per l'IAMutente.

Ruoli IAM

Un IAMruolo 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 ruolo IAM nella AWS Management Console mediante lo scambio di ruoli. È possibile assumere un ruolo chiamando un'azione AWS CLI o API AWS oppure utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere Uso IAM dei ruoli nella Guida per l'IAMutente.

I ruoli IAM 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, vedere Creazione di un ruolo per un provider di identità di terze parti nella Guida per l'IAMutente. Se utilizzi IAM Identity Center, configura 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 ulteriori 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, per alcuni dei Servizi AWS, è possibile collegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra i ruoli e le politiche basate sulle risorse per l'accesso tra account diversi, consulta In che modo i IAM 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 ruolo AWS a un'istanza EC2, affinché sia disponibile per tutte le relative applicazioni, puoi creare un profilo dell'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 Usare 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'IAMutente.

Gestione dell'accesso con policy

Per controllare l'accesso a AWS è possibile creare policy e collegarle a identità o risorse AWS. Una policy è un oggetto in AWS che, quando associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste policy quando un principale IAM (utente, utente root o sessione ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della

richiesta. La maggior parte delle policy viene archiviata in AWSsotto forma di documenti JSON. Per ulteriori informazioni sulla struttura e il contenuto dei documenti relativi alle policy JSON, consulta Panoramica delle politiche JSON nella Guida per l'IAMutente.

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti il permesso di eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM politiche. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, indipendentemente dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dalla AWS Management Console, la AWS CLI o l'API AWS.

Policy basate su identità

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 Creazione di IAM politiche nella Guida per l'IAMutente.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono incorporate direttamente in un singolo utente, gruppo o ruolo. Le policy gestite sono policy autonome che possono essere collegate a più utenti, gruppi e ruoli in Account AWS. Le policy gestite includono le policy gestite da AWS e le policy gestite dal cliente. Per informazioni su come scegliere tra una policy gestita o una politica in linea, consulta Scelta tra policy gestite e policy in linea nella Guida per l'utente. IAM

Policy basate su risorse

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 policy di trust dei IAM ruoli e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le politiche AWS gestite contenute in una politica basata sulle risorseIAM.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy 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 le ACL. Per maggiori informazioni sulle ACL, consulta Panoramica delle liste di controllo degli accessi (ACL) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta altri tipi di policy meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite di autorizzazioni è una funzionalità avanzata in cui si impostano le autorizzazioni massime che una politica basata sull'identità può concedere a un'entità (utente o ruolo IAM). IAM È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità nella Guida per l'utente. IAM IAM
- Policy di controllo dei servizi (SCP): le SCP sono policy JSON che specificano il numero massimo di autorizzazioni per un'organizzazione o unità organizzativa (OU) in AWS Organizations. AWS Organizationsè un servizio per il raggruppamento e la gestione centralizzata degli Account AWSmultipli di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. La SCP limita le autorizzazioni per le entità negli account membri, compreso ogni Utente root dell'account AWS. Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire

da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per informazioni su come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle politiche nella Guida per l'IAMutente.

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.

IAMfunzionalità che puoi utilizzare con Amazon Verified Permissions

Caratteristica IAM	Supporto per le autorizzazioni verificate
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	No
Liste di controllo degli accessi (ACL)	No
ABAC (tag nelle policy)	No
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
☻Ruoli di servizio	No

Caratteristica IAM	Supporto per le autorizzazioni verificate
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 AWSi servizi con cui funzionano IAM nella Guida per l'IAMutente.

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 Creazione di politiche nella Guida per l'utente. IAM IAM

Con le policy IAM IAM basate su identità, puoi specificare operazioni e risorse consentite o rifiutate, nonché le condizioni in base alle quali le operazioni sono consentite o rifiutate. 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 puoi 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 policy di trust dei IAM ruoli e le policy dei bucket

di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Per consentire l'accesso a più account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata su risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversiAccount AWS, un IAM amministratore dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta In che modo IAM i ruoli differiscono dalle politiche basate sulle risorse nella Guida per l'IAMutente.

Azioni politiche per le autorizzazioni verificate

Supporta le azioni di policy	Sì

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire azioni 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 a un criterio. Le azioni di policy hanno spesso lo stesso nome dell'operazione API AWS. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi 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 operazioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le operazioni che iniziano con la parola `Get`, includi la seguente operazione:

```
"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 JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale 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'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, 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 dei relativi ARN, 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 JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi 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.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND` logica. Se specifichi più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione `OR` logica. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi IAM della politica: variabili e tag nella Guida per l'IAMutente.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche per il servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida IAM per l'utente.

ACL nelle autorizzazioni verificate

Supporta le ACL	No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con autorizzazioni verificate

Supporta ABAC (tag nelle policy)	No

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, tali attributi sono denominati tag. È possibile allegare tag a IAM entità (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci 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 Yes (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 Che cos'è ABAC? nella Guida per l'IAMutente. 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

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle da utilizzare con credenziali temporanee, consulta la sezione relativa alla modalità Servizi AWSdi utilizzo IAM nella Guida per l'IAMutente. Servizi AWS

Le credenziali temporanee sono utilizzate se si accede alla AWS Management Console utilizzando qualsiasi metodo che non sia la combinazione di nome utente e password. Ad esempio, quando accedi ad AWS utilizzando il collegamento Single Sign-On (SSO) della tua azienda, tale processo

crea in automatico credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sul cambio di ruolo, vedere Passaggio a un ruolo (console) nella Guida per l'IAMutente.

È possibile creare manualmente credenziali temporanee utilizzando la AWS CLIo l'API AWS. È quindi possibile utilizzare tali credenziali temporanee per accedere ad AWS. AWSconsiglia di generare le credenziali temporanee dinamicamente anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni principali multiservizio per le autorizzazioni verificate

Supporta le autorizzazioni delle entità principali	Sì

Quando si utilizza un utente o un ruolo IAM per eseguire operazioni in AWS, si viene considerati un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'azione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che effettua la chiamata a un Servizio AWS, combinate con il Servizio AWS richiedente, per effettuare richieste a servizi a valle. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che necessita di interazioni con altri Servizi AWS o risorse per essere portata a termine. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

Ruoli di servizio per le autorizzazioni verificate

Supporta i ruoli di servizio	No

Un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio da IAM. Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.

Ruoli collegati ai servizi per le autorizzazioni verificate

Supporta i ruoli collegati ai servizi	No

Un ruolo collegato ai servizi è un tipo di ruolo di servizio che è collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati ai servizi sono visualizzati nell'account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare, ma non modificare le autorizzazioni dei ruoli collegati ai servizi.

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.

Esempi di policy basate sull'identità per Amazon Verified Permissions

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare risorse di Autorizzazioni verificate. Inoltre, non sono in grado di eseguire attività utilizzando la AWS Management Console, l'AWS Command Line Interface (AWS CLI) o l'API AWS. 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 degli ARN per ciascun 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 operazioni possono comportare costi aggiuntivi per l'Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Nozioni di base sulle policy gestite da AWSe passaggio alle autorizzazioni con privilegio minimo: per le informazioni di base su come concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite da AWSche concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo Account AWS. Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente di AWSspecifiche 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'utente. IAM
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le IAM politiche, concedi solo le autorizzazioni necessarie per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo per applicare le autorizzazioni, consulta Politiche e autorizzazioni nella Guida IAM per l'utente. IAM IAM
- 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. Puoi inoltre utilizzare le condizioni per concedere l'accesso alle operazioni di servizio, ma solo se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM: condizione nella Guida per l'utente di IAM.
- Usa 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. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta la convalida delle policy di IAM Access Analyzer nella Guida per l'utente. IAM
- Richiesta dell'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o utenti root nel tuo Account AWS, attiva MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso all'API protetto da MFA nella Guida per l'utente. IAM

Per ulteriori informazioni sulle best practice inIAM, consulta la sezione Procedure consigliate per la sicurezza IAM nella Guida per l'utente. IAM

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 concedere le autorizzazioni minime della console agli utenti che effettuano chiamate solo alla AWS CLI o all'API AWS. Al contrario, concedi l'accesso solo alle operazioni 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 allegate alla relativa identità utente. La policy include le autorizzazioni per completare questa azione sulla console o a livello di programmazione utilizzando la AWS CLIo l'API 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": "*"
        }
    ]
}
```

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 di 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`*.

Per ulteriore assistenza con l'accesso, contatta l'amministratore AWS. 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 AWSconsentono di trasmettere un ruolo esistente a tale servizio, invece di creare un nuovo ruolo di servizio o un ruolo collegato ai servizi. 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 passare 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`.

Per ulteriore assistenza con l'accesso, contatta l'amministratore AWS. 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 di 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 servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori 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'IAMutente.
- Per scoprire come fornire l'accesso alle tue risorse a terze partiAccount AWS, consulta Fornire l'accesso a persone Account AWS di proprietà di terzi nella Guida per l'IAMutente.
- 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'IAMutente.
- Per conoscere la differenza tra l'utilizzo dei ruoli e delle politiche basate sulle risorse per l'accesso tra account diversi, consulta In che modo i IAM ruoli differiscono dalle politiche basate sulle risorse nella Guida per l'utente. IAM

Convalida della conformità per Amazon Verified Permissions

Per sapere se il Servizio AWSè coperto da programmi di conformità specifici, consulta i Servizi AWScoperti dal programma di conformità e scegli il programma di conformità desiderato. Per informazioni generali, consulta Programmi per la conformità di AWS.

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Download di report in AWS Artifact.

La responsabilità di conformità durante l'utilizzo dei Servizi AWSè determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle normative vigenti. Per semplificare il rispetto della conformità, AWSmette a disposizione le seguenti risorse:

- Guide Quick Start per la sicurezza e conformità: queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono la procedura per l'implementazione di ambienti di base su AWSincentrati sulla sicurezza e sulla conformità.
- Architecting for HIPAA Security and Compliance on Amazon Web Services: questo whitepaper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni conformi ai requisiti HIPAA.

> **Note**
>
> Non tutti i Servizi AWSsono conformi ai requisiti HIPAA. Per ulteriori informazioni, consulta la sezione Riferimenti sui servizi conformi ai requisiti HIPAA.

- Risorse per la conformità AWS: una raccolta di cartelle di lavoro e guide suddivise per settore e area geografica.
- AWSGuide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- Valutazione delle risorse con le regole nella Guida per gli sviluppatori di AWS Config: il servizio AWS Configvaluta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti.
- AWS Security Hub: questo Servizio AWSfornisce una visione completa dello stato di sicurezza all'interno di AWS. La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWSe verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- AWS Audit Manager: questo Servizio AWSaiuta a verificare continuamente l'utilizzo di AWSper semplificare la gestione dei rischi e della conformità alle normative e agli standard di settore.

Resilienza nelle autorizzazioni verificate di Amazon

L'infrastruttura globale dei servizi AWS è progettata attorno a regioni Regioni AWS e zone di disponibilità. Le regioni di Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate che sono connesse tramite reti altamente ridondanti, a bassa latenza e a velocità effettiva elevata. Con le Zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico 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.

Quando crei un archivio di norme sulle autorizzazioni verificate, questo viene creato all'interno di un individuoRegione AWSe 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 e le zone di disponibilità, consulta Infrastruttura globale di AWS.

Monitoraggio delle autorizzazioni verificate da Amazon

Monitora è importante per garantire l'affidabilità, per garantire l'affidabilità, la disponibilità, la disponibilità, la disponibilità, la disponibilità, la disponibilità, la disponibilità, la disponibilità, la disponibilità, la disponibilità, la disponibilità,AWS. AWSfornisce i seguenti strumenti di monitoraggio per tenere sotto controllo le prestazioni, segnalare un problema e intervenire automaticamente quando necessario:

- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo account AWS e fornisce i file di log a un bucket Amazon S3 specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consultare la Guida per l'utente AWS CloudTrail.

Registrazione delle chiamate API Amazon Verified Permissions utilizzando AWS CloudTrail

Amazon Verified Permissions è integrato conAWS 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 trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per le autorizzazioni verificate. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. 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 CloudTrailutente.

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 in

Account AWS. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi del tuoAccount AWS, compresi gli eventi per le autorizzazioni verificate, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il percorso registra gli eventi di tutte le Regioni nella partizione AWSe distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei CloudTrail log. 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 le credenziali utente AWS Identity and Access Management (IAM) o root.
- 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 servizio AWS.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Per impostazione predefinita, gli eventi di dati come IsAuthorizede 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 distribuzione di eventi come file di log in un bucket Amazon S3 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.

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 relative alle autorizzazioni verificate da Amazon con AWS CloudFormation

Amazon Verified Permissions è integrato conAWS CloudFormation, un servizio che ti aiuta a modellare e configurare AWS le tue risorse in modo da dedicare meno tempo alla creazione e alla gestione delle tue risorse e della tua infrastruttura. Crei un modello che descriva tutte le AWS risorse che desideri (come gli archivi delle politiche) e AWS CloudFormation disponi e configura tali risorse per te.

Quando lo usiAWS CloudFormation, puoi riutilizzare il modello per configurare le risorse relative alle autorizzazioni verificate in modo coerente e ripetuto. Basta descrivere le risorse una volta sola, dopodiché si può effettuare il provisioning di tali risorse quante volte si vuole in più Account AWS e regioni.

> **Important**
>
> Amazon Cognito Identity non è disponibile in tutte le Regioni AWS stesse condizioni delle autorizzazioni verificate di Amazon. Se ricevi un errore AWS CloudFormation relativo a Amazon Cognito Identity, ad esempio`Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient`, ti consigliamo di creare il pool di utenti e il client Amazon Cognito nell'area geografica più vicina a Regione AWS cui è disponibile Amazon Cognito Identity. Usa questo pool di utenti appena creato per creare la fonte di identità delle autorizzazioni verificate.

Autorizzazioni e AWS CloudFormation modelli verificati

Per fornire e configurare le risorse per le autorizzazioni verificate e i servizi correlati, è necessario comprendere i AWS CloudFormationmodelli. I modelli sono file di testo formattati in JSON o YAML. Questi modelli descrivono le risorse di cui intendi effettuare il provisioning negli stack AWS CloudFormation. Se non hai familiarità con JSON o YAML, puoi usare AWS CloudFormation Designer per iniziare a utilizzare i modelli AWS CloudFormation. Per ulteriori informazioni, consulta Che cos'è AWS CloudFormation Designer? nella Guida per l'utente di AWS CloudFormation.

Verified Permissions supporta la creazione di fonti di identità, policy store e modelli di policy inAWS CloudFormation. Per ulteriori informazioni, inclusi esempi di modelli JSON e YAML per le risorse

Autorizzazioni verificate, consulta il riferimento al tipo di risorsa Amazon Verified Permissions nella Guida per l'utente. AWS CloudFormation

Ulteriori informazioni su AWS CloudFormation

Per ulteriori informazioni su AWS CloudFormation, consulta le seguenti risorse:

- AWS CloudFormation
- Guida per l'utente di AWS CloudFormation
- Documentazione di riferimento dell'API AWS CloudFormation
- Guida per l'utente dell'interfaccia a riga di comando di AWS CloudFormation

Accesso alle autorizzazioni per Amazon utilizzando un endpoint di interfaccia () AWS PrivateLink

Puoi utilizzare AWS PrivateLink per creare una connessione privata tra il tuo VPC e le autorizzazioni per Amazon. Puoi accedere alle Autorizzazioni come se fosse nel tuo VPC, senza utilizzare un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect Le istanze presenti nel VPC non richiedono indirizzi IP pubblici per accedere alle Autorizzazioni.

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. Queste sono interfacce di rete gestite dal richiedente che fungono da punto di ingresso per il traffico destinato alle Autorizzazioni

Per ulteriori informazioni, consulta la sezione Accesso a Servizi AWS tramite AWS PrivateLink nella Guida di AWS PrivateLink.

Consider

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 operazioni API attraverso l'endpoint di interfaccia.

Le policy degli endpoint VPC non sono supportate per le Autorizzazioni Per impostazione predefinita, l'accesso completo alle Autorizzazioni per le Autorizzazioni è consentito attraverso 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 tramite l'endpoint di interfaccia.

Crea un endpoint di interfaccia per le autorizzazioni

Puoi creare un endpoint di interfaccia per le autorizzazioni per Verified 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 per Verified utilizzando uno dei seguenti nomi di servizi:

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

Se si abilita il DNS privato per l'endpoint di interfaccia, è possibile effettuare richieste API verso le Autorizzazioni per l'endpoint di interfaccia. Ad esempio, `verifiedpermissions.us-east-1.amazonaws.com`.

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

Nome	Predefinita	Adattabile	Descrizione
Le policy vengono archiviate per regione e per account	Ogni regione supportata: 1.000	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.

Nome	Predefinita	Adattabile	Descrizione
Dimensione della richiesta di autorizzazione[1]	1 MB	No	La dimensione massima di una richiesta di autorizzazione.
Dimensione della politica	10,000 byte	No	La dimensione massima di una singola politica.
Dimensioni dello schema	100.000 byte	No	La dimensione massima dello schema di un archivio di politiche.
Dimensione della policy per risorsa	200.000 byte[2]	No	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.

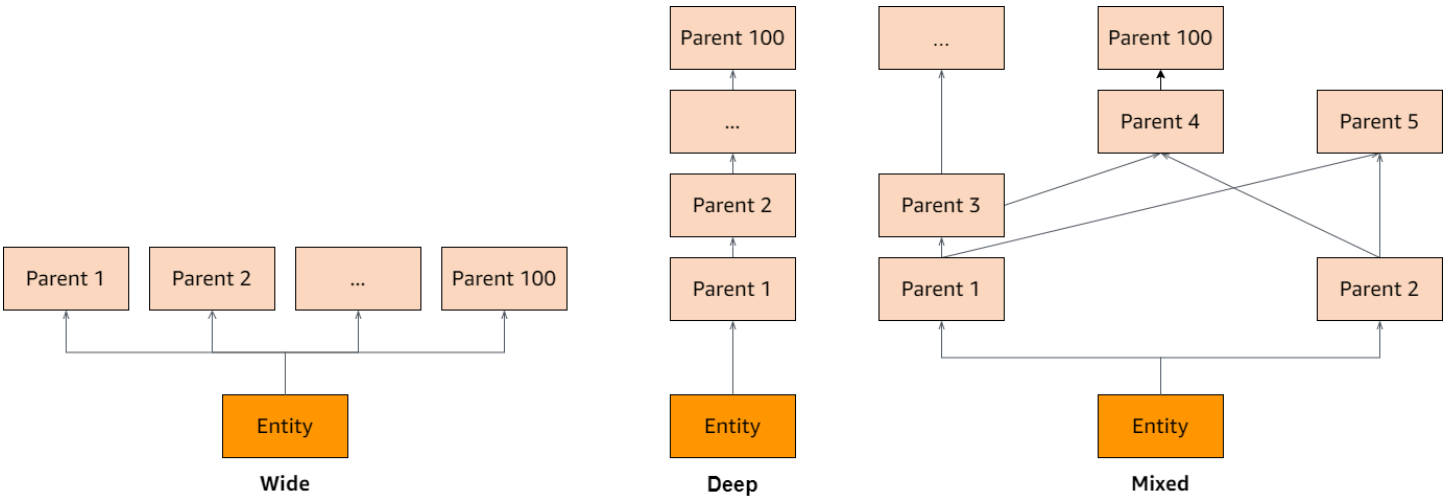
[2] La dimensione totale di tutte le politiche relative a una singola risorsa non può superare i 200.000 byte. Per le policy collegate al modello, la dimensione del modello di policy viene conteggiata una sola volta, più la dimensione di ogni set di parametri utilizzato per creare un'istanza di ogni policy collegata al modello.

Quote per le gerarchie

Nome	Predefinita	Adattabile	Descrizione
Genitori transitivi per preside	100	No	Il numero massimo di genitori transitivi per ogni principale.

Nome	Predefinita	Adatta e	Descrizione
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.

Nome	Predefinita	Adattabile	Descrizione
BatchIsAuthorized richieste al secondo per regione per account	Ogni regione supportata: 30	Sì	Il numero massimo di BatchIsAuthorized richieste al secondo.
CreatePolicy richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di CreatePolicy richieste al secondo.
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 account	Ogni regione supportata: 10	Sì	Il numero massimo di CreatePolicyTemplate richieste al secondo.
DeletePolicy richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di DeletePolicy richieste al secondo.
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 account	Ogni regione supportata: 10	Sì	Il numero massimo di DeletePolicyTemplate richieste al secondo.
GetPolicy richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di GetPolicy richieste al secondo.
GetPolicyTemplate richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di GetPolicyTemplate richieste al secondo.

Nome	Predefinita	Adattabile	Descrizione
GetSchema richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di GetSchema richieste al secondo.
IsAuthorized richieste al secondo per regione per account	Ogni Regione supportata: 200	Sì	Il numero massimo di IsAuthorized richieste al secondo.
IsAuthorizedWithToken richieste al secondo per regione per account	Ogni Regione supportata: 200	Sì	Il numero massimo di IsAuthorizedWithToken richieste al secondo.
ListPolicies richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di ListPolicies richieste al secondo.
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 account	Ogni regione supportata: 10	Sì	Il numero massimo di ListPolicyTemplates richieste al secondo.
PutSchema richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di PutSchema richieste al secondo.
UpdatePolicy richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di UpdatePolicy richieste al secondo.
UpdatePolicyTemplate richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di UpdatePolicyTemplate richieste al secondo.

Cronologia dei documenti per la Guida per l'utente di Amazon Verified Permissions

La tabella seguente descrive le versioni della documentazione per le autorizzazioni verificate.

Modifica	Descrizione	Data
Integrazione di AWS CloudFormation	Verified Permissions supporta la creazione di fonti di identità, policy store e modelli di policy inAWS CloudFormation.	30 giugno 2023
Versione iniziale	Versione iniziale della Guida per l'utente di Amazon Verified Permissions	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.