



Guida per l'utente

Autorizzazioni verificate da Amazon



Autorizzazioni verificate da Amazon: Guida per l'utente

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è Amazon Verified Permissions?	1
Autorizzazione nelle autorizzazioni verificate	1
Linguaggio delle politiche Cedar	1
Vantaggi delle autorizzazioni verificate	2
Accelera lo sviluppo delle applicazioni	2
Applicazioni più sicure	2
Funzionalità per l'utente finale	2
Servizi correlati	2
Accesso alle autorizzazioni verificate	3
Prezzi delle autorizzazioni verificate	5
Nozioni di base	6
Registrati per un Account AWS	6
Crea un utente con accesso amministrativo	6
.....	8
Crea il tuo primo negozio di polizze	8
Creazione di un archivio di policy di esempio	8
Creazione di politiche collegate a modelli per un archivio di policy di esempio	9
Test di un esempio di policy store	10
Crea un archivio API di policy collegato	13
Progettazione di un modello di autorizzazione	15
Nessun modello corretto	16
Concentrati sulle risorse	17
Autorizzazione composta	18
Prendi in considerazione la multi-locazione	19
Confronto tra archivi di policy condivisi e archivi di policy per tenant	21
Come scegliere	22
Compila l'ambito della politica	22
Metti tutte le risorse in contenitori	23
Separare i principi dalle risorse	25
Rappresentare le relazioni	27
Relazioni basate sugli attributi	28
Relazioni basate su modelli	30
Autorizzazioni granulari	31
Altri motivi per richiedere l'autorizzazione	32

Archivi di policy	34
Creazione di archivi di policy	34
API-archivi di polizze collegati	43
Come funziona	44
Considerazioni	46
Aggiungere ABAC	47
Passaggio alla produzione	48
Risoluzione dei problemi	51
Eliminazione degli archivi delle politiche	54
Schema del Policy Store	56
Modifica dello schema - Visual	58
Modifica dello schema - JSON	60
Modalità di convalida delle politiche	61
Policy	63
Formattazione delle entità	64
Creazione di politiche statiche	69
Modifica delle politiche statiche	71
Politiche di test	73
Policy di esempio	75
Consente l'accesso a singole entità	76
Consente l'accesso a gruppi di entità	76
Consente l'accesso a qualsiasi entità	78
Consente l'accesso agli attributi di un'entità (ABAC)	78
Nega l'accesso	81
Utilizza la notazione tra parentesi per fare riferimento agli attributi del token	82
Utilizza la notazione a punti per fare riferimento agli attributi	83
Riflette gli attributi del token Amazon Cognito ID	83
Riflette gli OIDC attributi del token ID	84
Riflette gli attributi del token di accesso di Amazon Cognito	84
Riflette gli attributi del token di OIDC accesso	84
Modelli di policy e policy collegate a modelli	86
Creazione di modelli di policy	86
Creazione di politiche collegate ai modelli	88
Modifica dei modelli di policy	90
Esempi di politiche collegate a modelli	91
PhotoFlashesempi	92

DigitalPetStore esempi	93
TinyToDoesempi	94
Provider di identità	95
Utilizzo delle fonti di identità di Amazon Cognito	95
Lavorare con le fonti di identità OIDC	98
Convalida del cliente e del pubblico	99
Autorizzazione lato client per JWTs	100
Creazione di fonti di identità	102
Fonte di identità Amazon Cognito	103
OIDCfonte di identità	105
Modifica delle fonti di identità	108
Fonte di identità dei pool di utenti di Amazon Cognito	109
Fonte di identità OpenID Connect (OIDC)	111
Mappatura dei token sullo schema	112
Cose da sapere sulla mappatura degli schemi	113
Mappatura dei token ID	117
Mappatura dei token di accesso	121
Notazione alternativa per le dichiarazioni delimitate da due punti di Amazon Cognito	125
Autorizza le richieste	128
APIoperazioni	129
Modello di test	130
Integrazione con le applicazioni	132
.....	135
Valuta il contesto di esempio	137
Sicurezza	143
Protezione dei dati	143
Crittografia dei dati	145
Gestione dell'identità e degli accessi	145
Destinatari	146
Autenticazione con identità	146
Gestione dell'accesso con policy	150
Come funziona Amazon Verified Permissions con IAM	152
IAM politiche per le autorizzazioni verificate	159
Esempi di policy basate su identità	161
Risoluzione dei problemi	164
Convalida della conformità	166

Resilienza	167
Monitoraggio	168
CloudTrail registri	168
Informazioni sulle autorizzazioni verificate in CloudTrail	169
Informazioni sulle voci del file di registro delle autorizzazioni verificate	170
Lavorare con AWS CloudFormation	188
Autorizzazioni e modelli verificati AWS CloudFormation	188
AWS CDKcostrutti	189
Scopri di più su AWS CloudFormation	189
Usando AWS PrivateLink	190
Considerazioni	190
Creazione di un endpoint di interfaccia	190
Quote	192
Quote per le risorse	192
Quote per le gerarchie	193
Quote per operazioni al secondo	194
Termini e concetti	198
Modello di autorizzazione	199
Richiesta di autorizzazione	199
Risposta di autorizzazione	199
Politiche considerate	199
Dati contestuali	200
Definizione delle politiche	200
Dati dell'entità	200
Autorizzazioni, autorizzazioni e principi	200
Applicazione delle politiche	200
Archivio delle politiche	201
Politiche soddisfatte	201
Differenze con Cedar	201
Definizione dello spazio dei nomi	201
Supporto per modelli di policy	202
Supporto dello schema	202
Supporto per tipi di estensione	202
JSONFormato Cedar per le entità	202
Definizione dei gruppi di azione	203
Limiti di lunghezza e dimensione	203

Cronologia dei documenti	205
.....	ccvii

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](#)
- [Linguaggio delle politiche Cedar](#)
- [Vantaggi delle autorizzazioni verificate](#)
- [Servizi correlati](#)
- [Accesso alle autorizzazioni verificate](#)
- [Prezzi delle autorizzazioni verificate](#)

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 da come 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 richiede l'autorizzazione APIs per valutare le politiche Cedar archiviate con il servizio e fornire una decisione di accesso sull'opportunità o meno di un'azione.

Linguaggio delle politiche Cedar

Le politiche di autorizzazione in Verified Permissions sono scritte utilizzando il linguaggio di policy Cedar. Cedar è un linguaggio open source per scrivere politiche di autorizzazione e prendere decisioni di autorizzazione basate su tali politiche. Quando 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 un elenco utenti, un server di autenticazione e un servizio di autorizzazione per token e credenziali di accesso OAuth 2.0. AWS Quando crei un policy store, hai la possibilità di creare i tuoi principali e gruppi da un pool di utenti di Amazon Cognito. 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 RESTHTTP, WebSocket APIs su

qualsiasi scala. Quando crei un archivio di policy, hai la possibilità di creare azioni e risorse da un API Gateway API interno. Per ulteriori informazioni su API Gateway, consulta la [APIGateway Developer Guide](#).

- AWS IAM Identity Center— Con IAM Identity Center, è possibile gestire la sicurezza degli accessi per le identità dei dipendenti, noti anche come utenti della forza lavoro. IAM Identity Center offre un unico posto in cui è possibile creare o connettere gli utenti della forza lavoro e gestire centralmente il loro accesso a tutte le loro applicazioni. Account AWS Per ulteriori informazioni, consulta la [Guida per l'utente AWS IAM Identity Center](#).

Accesso alle autorizzazioni verificate

Puoi utilizzare Amazon Verified Permissions in uno dei seguenti modi.

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' AWS](#)utente. Accedi ad AWS

- [Console Amazon Verified Autorizzazioni](#)

AWS Strumenti da riga di comando

È possibile utilizzare gli strumenti della riga di AWS comando per impartire comandi dalla riga di comando del sistema per eseguire autorizzazioni e AWS attività verificate. L'utilizzo della riga di comando può essere più veloce e semplice rispetto all'uso della console. Gli strumenti a riga di comando sono inoltre utili per creare script che eseguono le attività di AWS .

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

- [verifiedpermissions](#) nel Command Reference AWS CLI
- [Autorizzazioni verificate da Amazon](#) in AWS Tools for Windows PowerShell

AWS SDKs

AWS fornisce SDKs (kit di sviluppo software) costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Python, Ruby,. NET, iOS, Android, ecc.).

SDKs Forniscono un modo conveniente per creare un accesso programmatico alle autorizzazioni verificate e. AWS Ad esempio, SDKs si occupano di attività come la firma crittografica delle richieste, la gestione degli errori e il ritentativo automatico delle richieste.

[Per ulteriori informazioni e per il download AWS SDKs, consulta Strumenti per. Amazon Web Services](#)

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

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

AWS CDKcostrutti

AWS Cloud Development Kit (AWS CDK) è un framework di sviluppo software open source per definire l'infrastruttura cloud in codice e fornirla tramite. AWS CloudFormation I costrutti, o componenti cloud riutilizzabili, possono essere utilizzati per creare modelli. AWS CloudFormation Questi modelli possono quindi essere utilizzati per implementare l'infrastruttura cloud.

Per ulteriori informazioni e per il download AWS CDKs, consulta [AWS Cloud Development Kit](#).

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

- [Autorizzazioni verificate Amazon L2 Construct CDK](#)

Autorizzazioni verificate API

Puoi accedere alle autorizzazioni verificate e in modo AWS programmatico utilizzando le autorizzazioni verificateAPI, che ti consentono di inviare HTTPS richieste direttamente al servizio. Quando utilizzi ilAPI, devi includere il codice per firmare digitalmente le richieste utilizzando le tue credenziali.

- [Guida di API riferimento per le autorizzazioni verificate da Amazon](#)

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 API richieste di policy c URL (clientURL) inviate ogni mese dalle vostre candidature 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 Account AWS fatturazione, consulta la Guida per l'AWS Billing utente](#).

[Se hai domande sulla AWS fatturazione, sugli account e sugli eventi, contatta. AWS Support](#)

Guida introduttiva ad Amazon Verified Permissions

Per iniziare con le autorizzazioni verificate, hai bisogno di un AWS account e di un utente IAM dell'Identity Center con le autorizzazioni per creare risorse in Autorizzazioni verificate.

Le seguenti sezioni ti aiuteranno a creare un AWS account e gli utenti necessari:

Registrati per un Account AWS

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

Per iscriverti a un Account AWS

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

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

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

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. 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.

Crea un utente con accesso amministrativo

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

Proteggi i tuoi Utente root dell'account AWS

1. Accedi [AWS Management Console](#) come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Signing in as the root user](#) della Guida per l'utente di Accedi ad AWS .

2. Attiva l'autenticazione a più fattori (MFA) per il tuo utente root.

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

Crea un utente con accesso amministrativo

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 IAM Identity Center, concedi l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta [Configurare l'accesso utente con i valori predefiniti IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

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

Per informazioni sull'accesso con un utente di IAM Identity Center, consulta [Accesso al portale di AWS accesso](#) nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

Segui le istruzioni riportate nella pagina [Creazione di un set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

Per istruzioni, consulta [Aggiungere gruppi](#) nella Guida per l'utente di AWS IAM Identity Center .

Ora che hai creato un AWS account e alcuni utenti, sei pronto per creare un policy store. Scegli una delle seguenti opzioni per iniziare a usare le autorizzazioni verificate:

- [Crea il tuo primo Amazon Verified Permissions Policy Store](#)
- [Creare un archivio di policy per l'utilizzo di API Gateway con un provider di identità](#)

Crea il tuo primo Amazon Verified Permissions Policy Store

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. Scegli Avvia.

Creazione di un archivio di policy di esempio

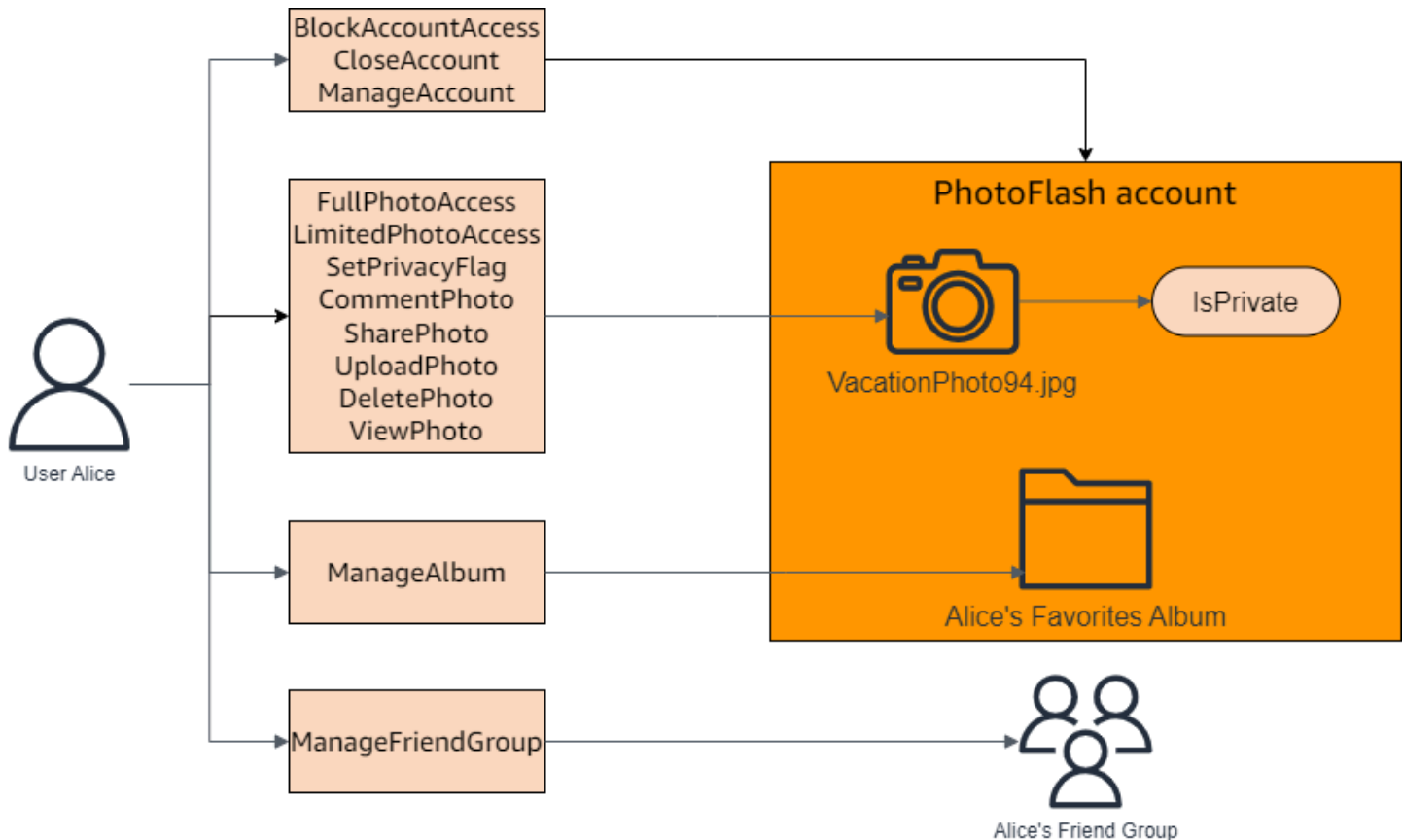
Se è la prima volta che utilizzi le autorizzazioni verificate, ti consigliamo di utilizzare uno degli archivi di policy di esempio per acquisire familiarità con il funzionamento delle autorizzazioni verificate. Gli archivi di policy di esempio forniscono policy e uno schema predefiniti.

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

1. Nella [console Autorizzazioni verificate](#), selezionare Crea nuovo policy store.
2. Nella sezione Opzioni di avvio, scegli Sample policy store.
3. 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.
4. Un namespace per lo schema del tuo policy store di esempio viene generato automaticamente in base al progetto di esempio che hai scelto.
5. 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, immettere `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, immettere `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 esempi](#)

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

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

Creare un archivio di policy per l'utilizzo di API Gateway con un provider di identità

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

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

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

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

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

Per ulteriori informazioni, consulta [API-archivi di polizze collegati](#).

Procedure consigliate per la progettazione di un modello di autorizzazione

Mentre ti prepari a utilizzare il servizio Amazon Verified Permissions all'interno di un'applicazione software, può essere difficile passare immediatamente alla stesura di dichiarazioni politiche come primo passo. Sarebbe come iniziare lo sviluppo di altre parti di un'applicazione scrivendo SQL dichiarazioni o API specifiche 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 entrambe le cose?
- 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»](#)
- [Concentratevi sulle vostre risorse oltre API che sulle operazioni](#)
- [L'autorizzazione composta è normale](#)
- [Considerazioni sulla multi-tenancy](#)
- [Quando possibile, compila l'ambito della policy](#)
- [Ogni risorsa vive in un contenitore](#)
- [Separate i principali dai contenitori di risorse](#)
- [Utilizzo di attributi o modelli per rappresentare le relazioni](#)
- [Preferisci le autorizzazioni granulari nel modello e le autorizzazioni aggregate nell'interfaccia utente](#)
- [Prendi in considerazione altri motivi per richiedere l'autorizzazione](#)

Non esiste un modello canonico «corretto»

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

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

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

Concentratevi sulle vostre risorse oltre API che sulle operazioni

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 valido e semplice che astrae l'implementazione sottostante e le operazioni di backend. API

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

Scoprirai che la procedura [introduttiva](#) nella console Autorizzazioni verificate include la possibilità di creare risorse e azioni direttamente da un API. Questa è una base utile: una mappatura diretta tra il tuo archivio delle politiche e API quello per cui autorizza.

Tuttavia, questo approccio API incentrato può essere tutt'altro che ottimale, perché APIs funge semplicemente da indicatore di ciò che i clienti stanno realmente cercando di proteggere: i dati e le risorse sottostanti. Se più utenti APIs controllano l'accesso alle stesse risorse, può essere difficile per gli amministratori ragionare sui percorsi verso tali risorse e gestire l'accesso di conseguenza.

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

- `listMembersOfGroup`
- `listGroupMembershipsForUser`

I clienti possono utilizzare una di queste operazioni per scoprire l'appartenenza al gruppo. Pertanto, l'amministratore delle autorizzazioni deve ricordarsi di coordinare l'accesso a entrambe le operazioni.

Ciò si complica ulteriormente se in seguito si sceglie di aggiungere una nuova API operazione per risolvere casi d'uso aggiuntivi, come i seguenti.

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

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

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

API Nome	Autorizzazioni	
<code>listMembersOfGroup</code>	richiede <code>viewGroupMembership</code>	l'autorizzazione per il gruppo
<code>listGroupMembershipsForUser</code>	richiede <code>viewGroupMembership</code>	l'autorizzazione dell'utente
<code>isUserInGroups</code>	richiede <code>viewGroupMembership</code>	l'autorizzazione dell'utente

Definendo quest'unica autorizzazione, l'amministratore controlla con successo l'accesso alla scoperta delle appartenenze ai gruppi, ora e per sempre. Come compromesso, ogni API operazione 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 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. Quando esistono gli handshake N-way, è comune osservare richieste di autorizzazione N composte 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 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 strategia di autorizzazione multi-tenant con Autorizzazioni Verificate.

Isolamento delle politiche degli inquilini

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

Flusso di autorizzazione

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

Gestione dei modelli e degli schemi

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 relativi ai singoli inquilini
Isolamento delle politiche degli inquilini	Medio. È necessario includere gli identificatori degli inquilini nelle politiche e nelle richieste di autorizzazione.	Basso. L'isolamento è un comportamento predefinito. Le politiche specifiche degli inquilini sono inaccessibili agli altri inquilini.
Flusso di autorizzazione	Basso. Tutte le interrogazioni hanno come target un archivio di policy.	Medio. Deve mantenere le mappature tra ogni tenant e il relativo ID dell'archivio delle politiche.
Modelli e gestione degli schemi	Basso. Deve far funzionare uno schema per tutti gli inquilini.	Alto. Gli schemi e i modelli potrebbero essere meno complessi singolarmente, ma le modifiche richiedono maggiore coordinamento e complessità.
Gestione delle politiche globali	Bassa. Tutte le politiche sono globali e possono essere aggiornate centralmente.	Alto. È necessario aggiungere e politiche globali a ciascun archivio di polizze in fase di onboarding. Replica gli aggiornamenti delle policy

globali tra molti archivi di policy.

Disimbarco da parte di un inquilino

Medio. È necessario identificare ed eliminare solo le politiche specifiche del tenant.

Basso. Eliminare l'archivio delle politiche.

Quote di risorse di servizio

Alto. I tenant condividono le quote di risorse che influiscono sugli archivi delle politiche, come la dimensione dello schema, la dimensione dei criteri per risorsa e le fonti di identità per l'archivio delle politiche.

Basso. Ogni inquilino dispone di quote di risorse dedicate.

Come scegliere

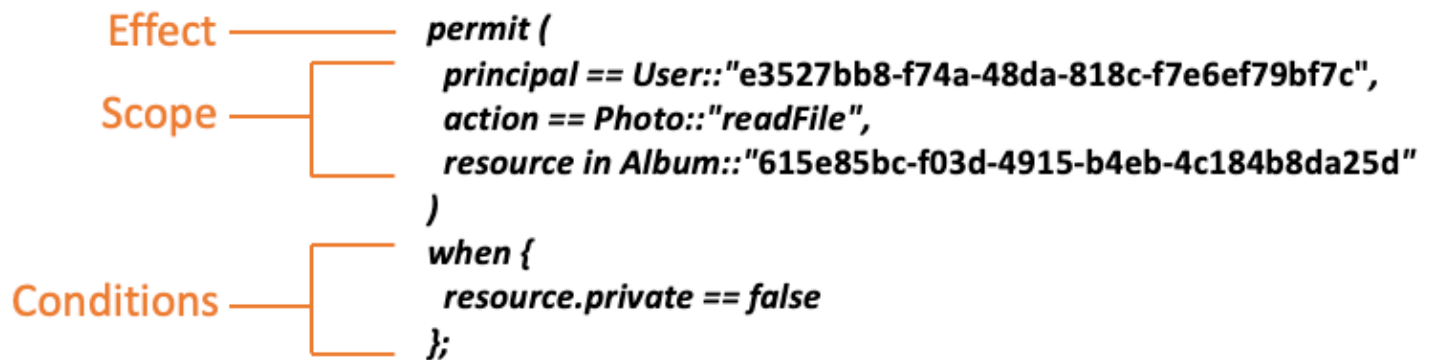
Ogni applicazione multi-tenant è diversa. Confrontate attentamente i due approcci e le relative considerazioni prima di prendere una decisione architettonica.

Se l'applicazione non richiede policy specifiche per i tenant e utilizza un'unica [fonte di identità](#), un archivio di policy condiviso per tutti i tenant è probabilmente la soluzione più efficace. Ciò si traduce in un flusso di autorizzazione più semplice e nella gestione delle policy globali. L'eliminazione di un tenant utilizzando un archivio di policy condiviso richiede meno sforzi perché l'applicazione non deve eliminare le politiche specifiche del tenant.

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

Quando possibile, compila l'ambito della policy

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



Ti consigliamo di compilare i valori ogni volta che è possibile. `principal resource` Ciò consente alle autorizzazioni verificate di indicizzare le politiche per un recupero più efficiente e quindi di migliorare 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 principi e risorse in una clausola. `when` In questo modo potresti incorrere in 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 `questaviewFile`, la risorsa a cui è possibile applicarla è intuitiva: un singolo file o forse una raccolta di file all'interno di una cartella. Tuttavia, un'operazione come questa `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.

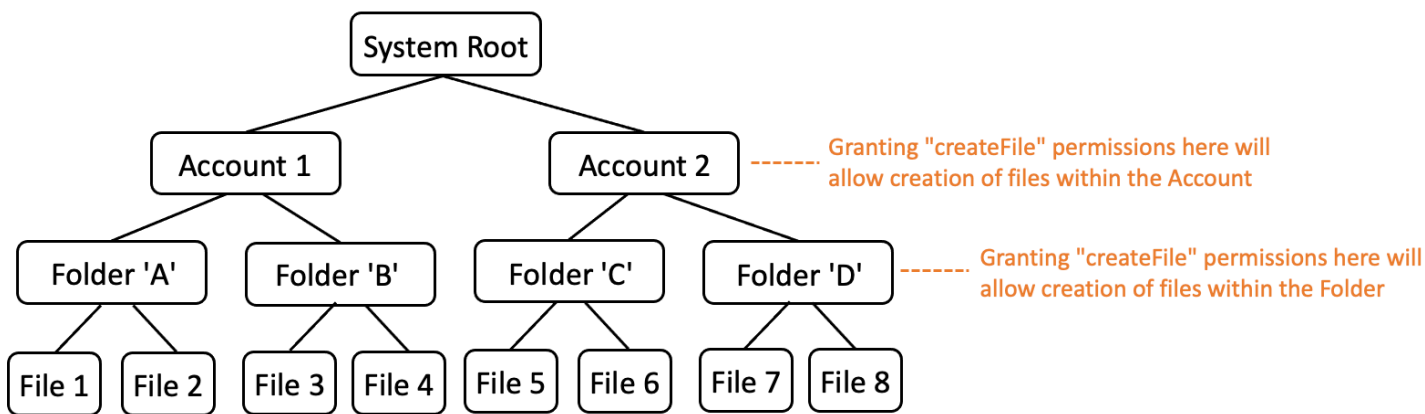
```
permit (  
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",  
    action == Action::"createFile",  
    resource == Folder::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"  
);
```

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.

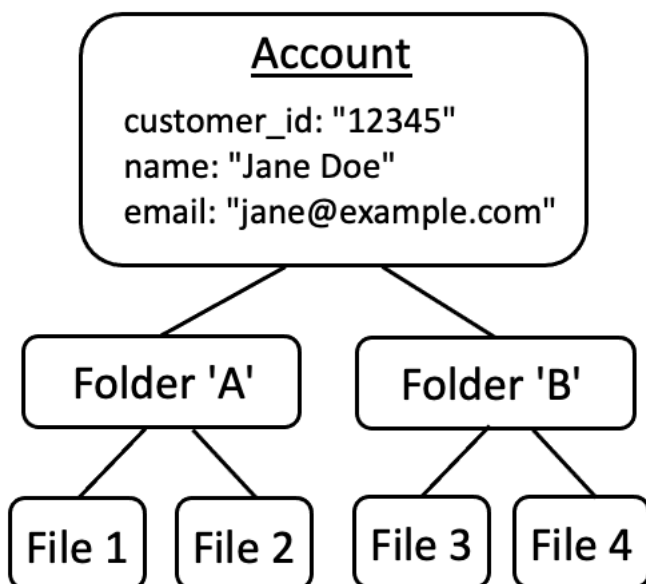
Per questa applicazione di esempio, la gerarchia dei contenitori risultante apparirebbe quindi come segue:



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.

Separate 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.

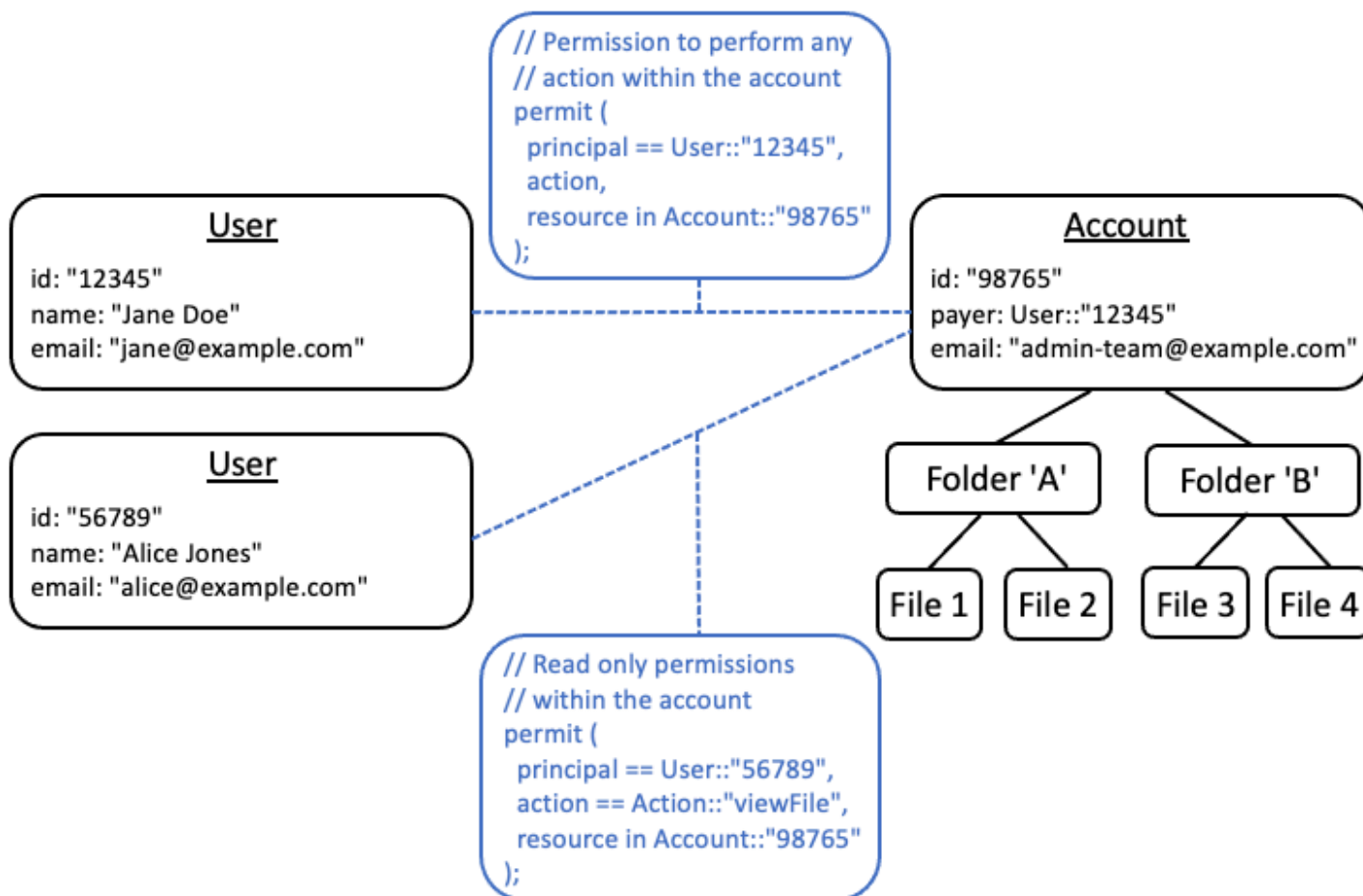


Ti consigliamo di considerare questa strategia come un anti-pattern. Questo perché c'è una tendenza naturale nelle applicazioni più ricche a delegare l'accesso ad altri utenti. Ad esempio,

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.

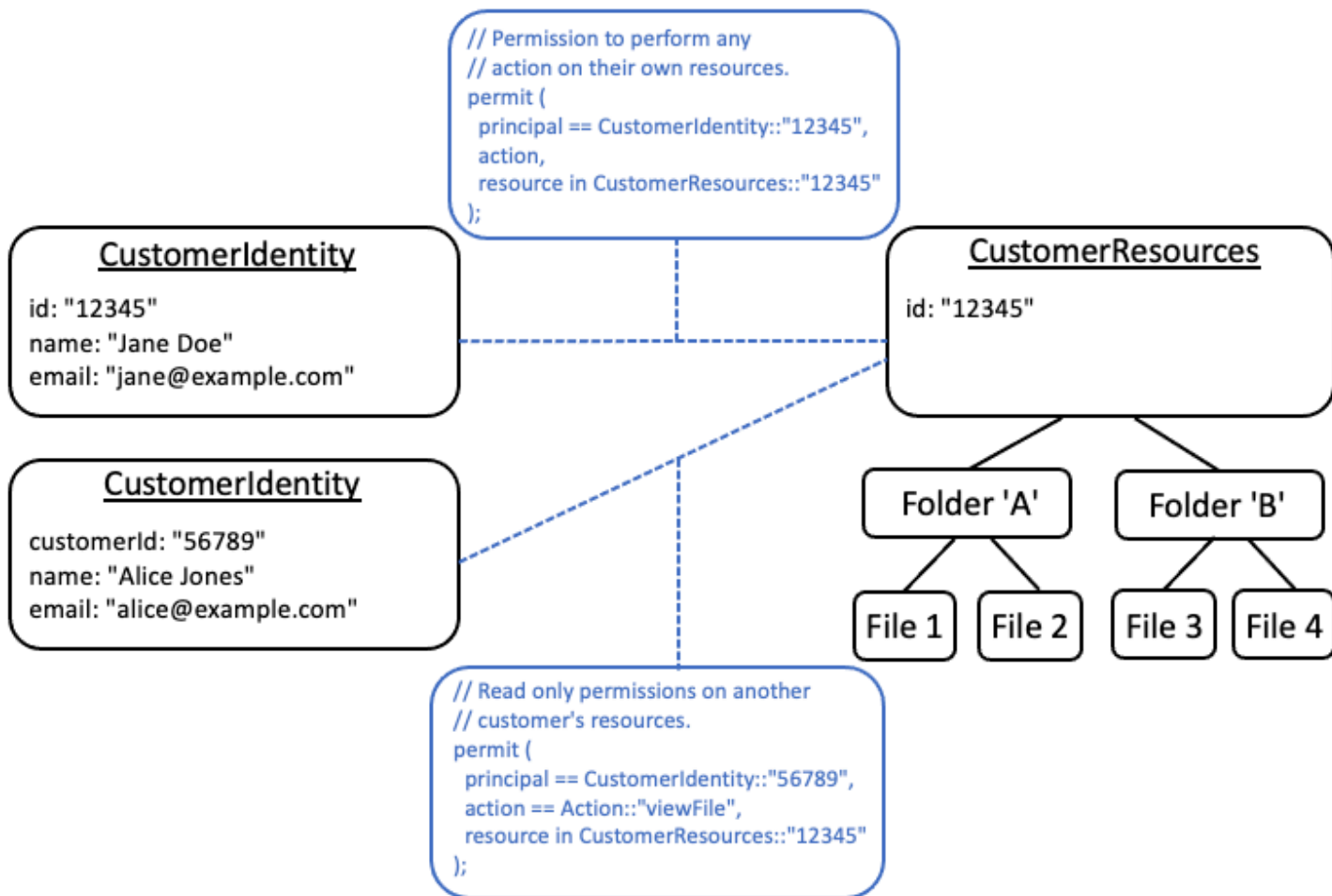
Quando un'identità utente viene utilizzata come contenitore di risorse per un account, gli scenari precedenti diventano più difficili da realizzare. Ancora più allarmante, se ad altri viene concesso l'accesso al contenitore dell'account con questo approccio, potrebbe inavvertitamente essere autorizzato 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 cui è proprietaria, potrebbe mapparla a un modello di autorizzazione che contiene un'entità logica per Customer Identity (contenente nome, e-mail, ecc.) e un'entità logica separata per Customer Resources o Customer Account, che funge da nodo principale per tutte le risorse di cui è proprietaria. Entrambe le entità possono condividere la stessa cosaId, ma con un'entità diversa. Type



Utilizzo di attributi o modelli per rappresentare le relazioni

Esistono due modi principali per esprimere le relazioni tra le risorse. Il momento in cui utilizzare l'una o l'altra dipende dal fatto che la relazione sia già memorizzata o meno nel database dell'applicazione e utilizzata per altri motivi, come la conformità. Se lo è, adotta l'approccio [basato sugli attributi](#). In caso contrario, adotta l'approccio basato su modelli.

Relazioni basate sugli attributi

Gli attributi possono essere utilizzati come input per la decisione di autorizzazione per rappresentare una relazione tra un principale e una o più risorse.

Questo modello è appropriato quando la relazione viene tracciata e gestita per scopi che vanno oltre la semplice gestione delle autorizzazioni. Ad esempio, la registrazione del titolare principale del conto è necessaria per la conformità finanziaria alle regole Know Your Customer. Le autorizzazioni derivano da queste relazioni. I dati sulla relazione vengono gestiti all'esterno del sistema di autorizzazione e recuperati come input quando si prende una decisione di autorizzazione.

L'esempio seguente mostra come potrebbe essere rappresentata una relazione tra un utente Alice e una serie di account di cui è il principale titolare dell'account:

```
// Using a user attribute to represent the primary account holder relationship
{
  "id": "df82e4ad-949e-44cb-8acf-2d1acda71798",
  "name": "alice",
  "email": "alice@example.com",
  "primaryOnAccounts": [
    "Account:\c943927f-d803-4f40-9a53-7740272cb969\"",
    "Account:\b8ee140c-fa09-46c3-992e-099438930894\""
  ]
}
```

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

```
// Derived relationship permissions
permit (
  principal,
  action in Action:"primaryAccountHolderActions",
  resource
)when {
  resource in principal.primaryOnAccounts
};
```

Al contrario, la stessa relazione potrebbe essere rappresentata come un attributo sulla risorsa chiamata `primaryAccountHolders` che contiene un insieme di utenti.

Se esistono più tipi di relazione tra i principali e le risorse, questi devono essere modellati come attributi diversi. Ad esempio, se gli account possono avere anche firmatari autorizzati e queste

persone dispongono di autorizzazioni diverse sull'account, questo verrà rappresentato come un attributo diverso.

Nel caso precedente, Alice potrebbe anche essere un firmatario autorizzato su un terzo account. L'esempio seguente mostra come ciò potrebbe essere rappresentato:

```
// Using user attributes to represent the primary account holder and authorized
  signatory relationships
{
  "id": "df82e4ad-949e-44cb-8acf-2d1acda71798",
  "name": "alice",
  "email": "alice@example.com",
  "primaryOnAccounts": [
    "Account::\"c943927f-d803-4f40-9a53-7740272cb969\"",
    "Account::\"b8ee140c-fa09-46c3-992e-099438930894\""
  ],
  "authorizedSignatoryOnAccounts": [
    "Account::\"661817a9-d478-4096-943d-4ef1e082d19a\""
  ]
}
```

Le seguenti sono le politiche corrispondenti:

```
// Derived relationship permissions

permit (
  principal,
  action in Action::"primaryAccountHolderActions",
  resource
)when {
  resource in principal.primaryOnAccounts
};

permit (
  principal,
  action in Action::"authorizedSignatoryActions",
  resource
)when {
  resource in principal.authorizedSignatoryOnAccounts
};
```

Relazioni basate su modelli

Se la relazione tra le risorse esiste esclusivamente ai fini della gestione delle autorizzazioni, è opportuno archiviare questa relazione come una politica o un modello collegato al modello. Puoi anche pensare a questi modelli come a ruoli assegnati a una risorsa specifica.

Ad esempio, in un sistema di gestione dei documenti, il proprietario del documento può scegliere di concedere l'autorizzazione a un altro utente per contribuire al documento. Alice Bob Ciò stabilisce una relazione di collaboratore tra Bob e il documento di Alice. L'unico scopo di questa relazione è concedere il permesso di modificare e commentare il documento, e quindi questa relazione può essere rappresentata come modello. In questi casi l'approccio consigliato consiste nel creare un modello per ogni tipo di relazione. Negli esempi seguenti sono presenti due tipi di relazione e quindi due modelli. `Contributor Reviewer`

I seguenti modelli possono essere utilizzati per creare politiche collegate ai modelli per singoli utenti.

```
// Managed relationship permissions - Contributor template
permit (
  principal == ?principal,
  action in Action::"DocumentContributorActions",
  resource in ?resource
);

// Managed relationship permissions - Reviewer template
permit (
  principal == ?principal,
  action in Action::"DocumentReviewerActions",
  resource in ?resource
);
```

I seguenti modelli possono essere utilizzati per creare politiche collegate ai modelli per gruppi di utenti. L'unica differenza rispetto ai modelli per singoli utenti è l'uso dell'`in` operatore anziché `di ==`

```
// Managed relationship permissions - Contributor template
permit (
  principal in ?principal,
  action in Action::"DocumentContributorActions",
  resource in ?resource
);

// Managed relationship permissions - Reviewer template
```

```
permit (  
  principal in ?principal,  
  action in Action::"DocumentReviewerActions",  
  resource in ?resource  
);
```

È quindi possibile utilizzare questi modelli per creare politiche, come le seguenti, che rappresentano le autorizzazioni relative alle relazioni gestite ogni volta che viene concesso l'accesso a un documento.

```
//Managed relationship permissions  
permit (  
  principal in User::"df82e4ad-949e-44cb-8acf-2d1acda71798",  
  action in Action::"DocumentContributorActions",  
  resource in Document::"c943927f-d803-4f40-9a53-7740272cb969"  
);  
  
permit (  
  principal in UserGroup::"df82e4ad-949e-44cb-8acf-2d1acda71798",  
  action in Action::"DocumentReviewerActions",  
  resource == Document::"661817a9-d478-4096-943d-4ef1e082d19a"  
);  
  
permit (  
  principal in User::"df82e4ad-949e-44cb-8acf-2d1acda71798",  
  action in Action::"DocumentContributorActions",  
  resource in Folder::"b8ee140c-fa09-46c3-992e-099438930894"  
);
```

Amazon Verified Permissions è in grado di gestire in modo efficiente molte policy individuali e dettagliate durante la valutazione delle autorizzazioni e modellare le cose in questo modo significa che Verified Permissions mantiene un registro di controllo completo di tutte le decisioni di autorizzazione. AWS CloudTrail

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

Una strategia che i progettisti spesso rimpiangono in seguito consiste nel progettare un modello di autorizzazione con azioni molto ampie, come ad esempio «and»Write, Read e rendersi 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 granulari. 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 dettagliati come `e.ReadWrite`. Per ottenere il meglio da entrambi i mondi, puoi raggruppare le autorizzazioni granulari in raccolte predefinite, ad esempio utilizzando meccanismi come modelli di policy o gruppi di azioni. `ReadWrite` Utilizzando questo approccio, i clienti vedono solo le autorizzazioni granulari 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 (ABAC) basate sugli attributi 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 rispetto 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 In questo modo RBAC, può essere più semplice eseguire iterazioni su tutte le politiche associate a un utente per determinare l'accesso alle risorse.

Archivi di policy di Amazon Verified Permissions

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

Si consiglia di creare un archivio delle politiche per applicazione o un archivio delle politiche per tenant per le 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. Questi namespace aiutano a mantenere le cose chiare quando lavori con più applicazioni simili. Ad esempio, nelle applicazioni multi-tenant, l'utilizzo di uno spazio dei nomi per aggiungere il nome del tenant ai tipi definiti nello schema li distinguerà dalle controparti simili utilizzate dagli altri tenant. Esaminando i log delle richieste di autorizzazione, sarete in grado di identificare facilmente il tenant che ha elaborato la richiesta di autorizzazione. Per ulteriori informazioni, consulta [Namespaces](#) nella Cedar Policy Language Reference Guide.

Argomenti

- [Creazione di archivi di policy per le autorizzazioni verificate](#)
- [API-archivi di polizze collegati](#)
- [Eliminazione degli archivi delle politiche](#)

Creazione di archivi di policy per le autorizzazioni verificate

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

- Segui una configurazione guidata: definirai un tipo di risorsa con azioni valide e un tipo principale prima di creare la tua prima politica.
- Configurazione con API Gateway e una fonte di identità: definisci le tue entità principali con gli utenti che accedono con un provider di identità (IdP) e le tue azioni e le entità di risorse da un

Amazon API Gateway. API Ti consigliamo questa opzione se desideri che la tua applicazione autorizzi API le richieste di appartenenza ai gruppi di utenti.

- Inizia da un esempio di policy store: scegli un esempio predefinito di policy store di progetto. Ti consigliamo questa opzione se stai imparando a conoscere le autorizzazioni verificate e desideri visualizzare e testare politiche di esempio.
- Crea un archivio delle politiche vuoto: definirai tu stesso lo schema e tutte le politiche di accesso. Consigliamo questa opzione se avete già dimestichezza con la configurazione di un policy store.

Guided setup

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

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

1. Nella [console Autorizzazioni verificate](#), seleziona Crea nuovo archivio di politiche.
2. Nella sezione Opzioni di avvio, scegli Configurazione guidata.
3. Inserisci una descrizione del Policy store. Questo testo può essere quello che più si addice all'organizzazione come riferimento esplicito alla funzione dell'attuale archivio delle politiche, ad esempio Weather updates.
4. Nella sezione Dettagli, digita un Namespace per lo schema.
5. Scegli Next (Successivo).
6. Nella finestra Tipo di risorsa, digita un nome per il tipo di risorsa.
7. (Facoltativo) Scegliete Aggiungi un attributo per aggiungere gli attributi della risorsa. Digita il nome dell'attributo e scegliete un tipo di attributo per ogni attributo della risorsa. Scegli 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.
8. Nel campo Azioni, digita le azioni da autorizzare per il tipo di risorsa specificato. Per aggiungere azioni aggiuntive per il tipo di risorsa, scegli Aggiungi un'azione. Per rimuovere un'azione che è stata aggiunta per il tipo di risorsa, scegli Rimuovi accanto all'azione.

9. Nel campo Nome del tipo principale, digita il nome di un tipo di principale che utilizzerà le azioni specificate per il tipo di risorsa.
10. Scegli Next (Successivo).
11. Nella finestra Tipo principale, scegli la fonte di identità per il tuo tipo principale.
 - Scegli Personalizzato se l'ID e gli attributi del principale verranno forniti direttamente dall'applicazione Autorizzazioni verificate. Scegli Aggiungi un attributo per aggiungere gli attributi principali. 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 AWS e 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.
12. Scegli Next (Successivo).
13. Nella sezione Dettagli della politica, digita una descrizione facoltativa della politica per la tua prima politica Cedar.
14. Nel campo Ambito dei principi, scegli i principali a cui verranno concesse le autorizzazioni previste dalla politica.
 - Scegli Principio specifico per applicare la politica a un principio specifico. Scegli il principale nel campo Principal a cui sarà consentito intraprendere azioni e digita un identificatore di entità per il principale.
 - Scegli Tutti i mandanti per applicare la politica a tutti i mandanti del tuo archivio polizze.
15. Nel campo Ambito delle risorse, scegli su quali risorse i responsabili specificati saranno autorizzati ad agire.
 - Scegli Risorsa specifica per applicare la politica a una risorsa specifica. Scegli la risorsa nel campo Risorsa a cui questo criterio dovrebbe applicarsi e digita un identificatore di entità per la risorsa.
 - Scegli Tutte le risorse per applicare la politica a tutte le risorse del tuo archivio delle politiche.

16. Nel campo Ambito delle azioni, scegli le azioni che i responsabili specificati saranno autorizzati a eseguire.
 - Scegli Set specifico di azioni per applicare la politica a azioni specifiche. Seleziona le caselle di controllo accanto alle azioni nel campo Azioni a cui questo criterio dovrebbe applicarsi.
 - Scegli Tutte le azioni per applicare la politica a tutte le azioni nel tuo archivio delle politiche.
17. Consulta la politica nella sezione Anteprima della politica. Scegli Crea archivio di politiche.

Set up with API Gateway and an identity source

Per creare un policy store utilizzando il metodo di configurazione Setup with API Gateway e un metodo di configurazione Identity Source

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

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

1. Nella [console Autorizzazioni verificate](#), seleziona Crea nuovo archivio di politiche.
2. Nella sezione Opzioni di avvio, scegli Configura con API gateway e un'origine di identità e seleziona Avanti.
3. Nella fase Importa risorse e azioni, sotto API, scegli un'opzione API che fungerà da modello per le risorse e le azioni del tuo policy store.
 - a. Scegli una fase di implementazione tra le fasi configurate nel tuo API e seleziona Importa API. Per ulteriori informazioni sulle API fasi, consulta [Configurazione di una fase per un REST API nella Amazon API Gateway Developer Guide](#).
 - b. Visualizza un'anteprima della mappa delle risorse e delle azioni importate.

- c. Per aggiornare risorse o azioni, modifica i API percorsi o i metodi e seleziona Importa API.
 - d. Quando sei soddisfatto delle tue scelte, scegli Avanti.
4. In Identity source, scegli un tipo di provider di identità. Puoi scegliere un pool di utenti Amazon Cognito o un tipo di IdP OpenID Connect (OIDC).
5. Se hai scelto Amazon Cognito:
- a. Scegli un pool di utenti nello stesso archivio delle Regione AWS polizze Account AWS .
 - b. Scegli il tipo di token da passare a API cui desideri inviare l'autorizzazione. Entrambi i tipi di token contengono gruppi di utenti, la base di questo modello API di autorizzazione collegato.
 - c. In App client validation, puoi limitare l'ambito di un policy store a un sottoinsieme dei client dell'app Amazon Cognito in un pool di utenti multi-tenant. Per richiedere l'autenticazione dell'utente con uno o più client di app specifici nel tuo pool di utenti, seleziona Accetta token solo con il client di app previsto. IDs Per accettare qualsiasi utente che si autentichi con il pool di utenti, seleziona Don't validate app client. IDs
 - d. Scegli Next (Successivo).
6. Se hai scelto il provider: OIDC
- a. In Emittente URL, inserisci il nome URL dell'OIDCemittente. Questo è l'endpoint del servizio che fornisce, ad esempio, il server di autorizzazione, le chiavi di firma e altre informazioni sul provider. `https://auth.example.com` L'emittente URL deve ospitare un documento di OIDC scoperta presso `/.well-known/openid-configuration`
 - b. In Tipo di token, scegli il tipo di token OIDC JWT che desideri che la tua richiesta invii per l'autorizzazione. Per ulteriori informazioni, consulta [Mappatura dei token del provider di identità allo schema](#).
 - c. In Token claims, scegli come configurare gli attributi utente nel tuo policy store. Questi attributi definiscono le affermazioni a cui possono fare riferimento le tue politiche.
 - i. Scegli una fonte di reclamo.
 - A. Per fornire un token di esempio, scegli Estrai dal JWT payload e incolla il payload di un JWT tipo di token scelto. JWTscontengono un'intestazione, un payload e una firma. Il campione JWT deve essere decodificato e deve essere utilizzato solo per il payload. Per analizzare il payload, selezionate Extract.
 - B. Per inserire il tuo set di attributi, scegli Inserisci reclami manualmente.

- ii. Inserisci o conferma il nome di ogni attestazione Token e il tipo di valore di Claim che desideri aggiungere agli attributi del principale utente o del contesto di azione dello schema.
 - d. In Attestazioni utente e di gruppo, scegli un'attestazione utente per l'origine dell'identità. Si tratta in genere sub di un'attestazione derivante dal tuo ID o token di accesso che contiene l'identificatore univoco dell'entità da valutare. Le identità dell'OIDCIdP connesso verranno mappate al tipo di utente nel tuo policy store.
 - e. In Attestazioni utente e di gruppo, scegli un'attestazione di gruppo come origine dell'identità. Si tratta in genere groups di un'affermazione basata sul tuo ID o token di accesso che contiene un elenco dei gruppi dell'utente. Il tuo archivio delle politiche autorizzerà le richieste in base all'appartenenza al gruppo.
 - f. In Audience validation o Client IDs, inserisci il cliente IDs o il pubblico URLs che desideri che il tuo policy store accetti nelle richieste di autorizzazione, se presenti. Per i token di accesso, inserisci un valore di Audience claim come. `https://myapp.example.com`
Per i token ID, inserisci un ID cliente come. `1example23456789`
 - g. Scegli Next (Successivo).
7. Se hai scelto Amazon Cognito, Verified Permissions interroga il tuo pool di utenti per i gruppi. Per i OIDC provider, inserisci i nomi dei gruppi manualmente. Il passaggio Assegna azioni ai gruppi crea politiche per l'archivio delle politiche che consentono ai membri del gruppo di eseguire azioni.
 - a. Scegli o aggiungi i gruppi che desideri includere nelle tue politiche.
 - b. Assegna azioni a ciascuno dei gruppi che hai selezionato.
 - c. Scegli Next (Successivo).
 8. In Deploy app integration, esamina i passaggi che Verified Permissions eseguirà per creare il tuo policy store e l'autorizzatore Lambda.
 9. Quando sei pronto per creare le nuove risorse, scegli Crea e distribuisci.
 10. Tieni aperta la fase di stato del Policy store nel browser per monitorare l'avanzamento della creazione delle risorse tramite Autorizzazioni verificate.
 11. Dopo qualche tempo, in genere circa un'ora, o quando la fase di autorizzazione Deploy Lambda mostra l'esito positivo, configura l'autorizzatore.

Verified Permissions avrà creato una funzione Lambda e un autorizzatore Lambda nel tuo API Scegli Apri API per accedere al tuo API

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


- a. Vai alla sezione Authorizers for your API e annota il nome dell'autorizzatore creato da Verified Permissions.
 - b. Vai a Risorse e seleziona un metodo di primo livello nel tuo. API
 - c. Seleziona Modifica nelle impostazioni di richiesta del metodo.
 - d. Imposta l'Autorizzatore in modo che sia il nome dell'autorizzatore che hai annotato in precedenza.
 - e. Espandi le intestazioni della HTTP richiesta, inserisci un nome o e seleziona **AUTHORIZATION** Obbligatorio.
 - f. Distribuisci lo stage. API
 - g. Salva le modifiche.
12. Testa il tuo sistema di autorizzazione con un token del pool di utenti del tipo Token selezionato nel passaggio Scegli l'origine dell'identità. Per ulteriori informazioni sull'accesso al pool di utenti e sul recupero dei token, consulta [Flusso di autenticazione del pool di utenti](#) nella Amazon Cognito Developer Guide.
13. Prova nuovamente l'autenticazione con un token del pool di utenti nell'AUTHORIZATIONintestazione di una richiesta al tuo. API
14. Esamina il tuo nuovo archivio di politiche. Aggiungi e perfeziona le politiche.

Sample policy store

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

1. Nella sezione Opzioni di avvio, scegli Sample policy store.
2. Nella sezione Progetto di esempio, scegli il tipo di applicazione di esempio per le autorizzazioni verificate da utilizzare.
 - PhotoFlashè un'applicazione web di esempio rivolta ai clienti che consente agli utenti di condividere foto e album individuali con gli amici. Gli utenti possono impostare autorizzazioni dettagliate su chi è autorizzato a visualizzare, commentare e condividere nuovamente le proprie foto. I proprietari di account possono anche creare gruppi di amici e organizzare le foto in album.

- DigitalPetStore è 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 DigitalPetStore di policy di esempio non include modelli di policy. Gli archivi TinyTodo di policy PhotoFlashe di esempio includono modelli di policy.

- TinyTodo è un'applicazione di esempio che consente agli utenti di creare attività ed elenchi di attività. I proprietari degli elenchi possono gestire e condividere i propri elenchi e specificare chi può visualizzare o modificare i propri elenchi.
3. Uno spazio dei nomi per lo schema del tuo archivio di policy di esempio viene generato automaticamente in base al progetto di esempio scelto.
 4. Scegli Crea archivio di politiche.

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

Empty policy store

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

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

Un policy store vuoto viene creato senza uno schema, il che significa che i criteri non vengono convalidati. Per ulteriori informazioni sull'aggiornamento dello schema per il policy store, vedere [Schema 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](#) e [Creazione di politiche collegate ai modelli di Amazon Verified Permissions](#).

AWS CLI

Per creare un archivio delle politiche vuoto utilizzando AWS CLI.

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

Note

Un archivio delle politiche creato utilizzando il AWS CLI è vuoto.

- Per aggiungere uno schema, vedere [Schema di archiviazione delle politiche di Amazon Verified Permissions](#).
- Per aggiungere politiche, vedere [Creazione di politiche statiche per le autorizzazioni verificate di Amazon](#).
- Per aggiungere modelli di policy, consulta [Creazione di modelli di policy Amazon Verified Permissions](#).

```
$ 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 delle politiche utilizzando `CreatePolicyStoreAPI`. Per ulteriori informazioni, consulta [CreatePolicyStore](#) la Amazon Verified Permissions API Reference Guide.

API-archivi di polizze collegati

Quando crei un nuovo archivio di policy nella console Amazon Verified Permissions, puoi scegliere l'opzione Configura con API Gateway e un'origine di identità. Con questa opzione, crei un archivio di policy API collegato, un modello di autorizzazione per le applicazioni che si autenticano con pool di utenti di Amazon Cognito o OIDC un provider di identità (IdP) e ottieni dati da Amazon Gateway. API APIs Per iniziare, consulta [Creare un archivio di policy per l'utilizzo di API Gateway con un provider di identità](#).

Argomenti

- [In che modo Verified Permissions autorizza le richieste API](#)
- [Considerazioni per gli archivi API di policy collegati](#)
- [Aggiungere il controllo degli accessi basato sugli attributi \(\) ABAC](#)
- [Passare alla produzione con AWS CloudFormation](#)
- [Archivi di policy collegati alla risoluzione dei problemi API](#)

Important

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

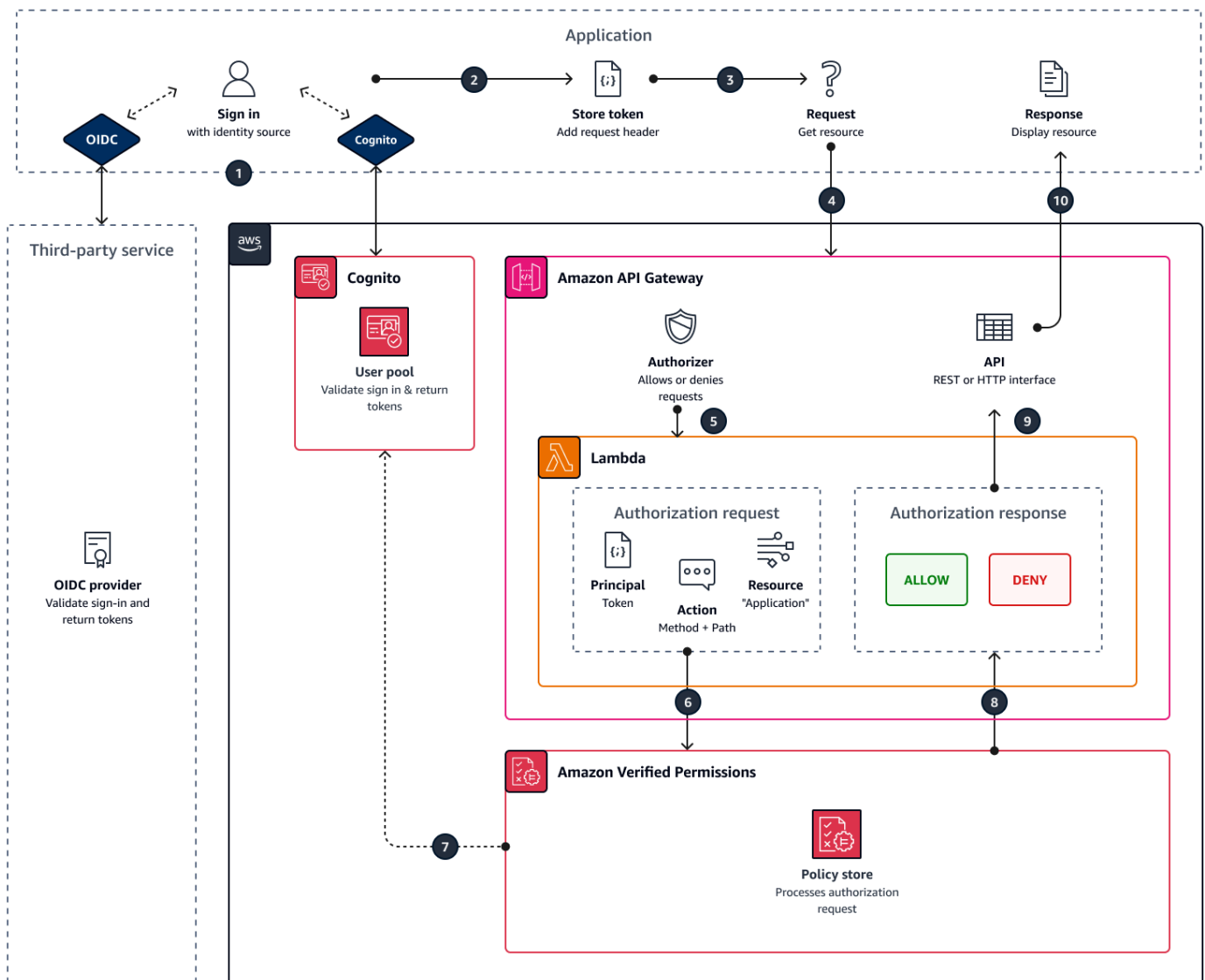
In un archivio di policy collegato a una fonte di identità API and, l'applicazione presenta un token del pool di utenti in un'intestazione di autorizzazione quando effettua una richiesta a. API La fonte di identità del tuo policy store fornisce la convalida dei token per le autorizzazioni verificate. Il token forma le richieste `principal` di autorizzazione con. [IsAuthorizedWithToken](#) API Verified Permissions crea politiche relative all'appartenenza ai gruppi degli utenti, come illustrato in una dichiarazione di gruppo in termini di identità (ID) e token di accesso, ad esempio `cognito:groups` per i pool di utenti. Il token dell'applicazione viene elaborato in un programma di autorizzazione Lambda e lo invia a Verified Permissions per una decisione di autorizzazione. API Quando ricevi API la decisione di autorizzazione dall'autorizzatore Lambda, trasmette la richiesta alla tua fonte di dati o la nega.

Componenti dell'origine dell'identità e dell'autorizzazione del API gateway con autorizzazioni verificate

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

In che modo Verified Permissions autorizza le richieste API

Quando si crea un nuovo Policy Store e si seleziona l'opzione Configura con Cognito e API Gateway, Verified Permissions crea lo schema e i criteri del Policy Store. Lo schema e le politiche riflettono API le azioni e i gruppi di utenti che desideri autorizzare a intraprendere tali azioni. [Verified Permissions crea anche la funzione e l'autorizzatore Lambda](#). È necessario configurare il nuovo autorizzatore su un metodo in. API



1. L'utente accede con la tua applicazione tramite Amazon Cognito o un altro OIDC IdP. L'IdP emette ID e token di accesso con le informazioni dell'utente.
2. L'applicazione memorizza il JWTs Per ulteriori informazioni, consulta [Usare i token con i pool di utenti](#) nella Amazon Cognito Developer Guide.
3. L'utente richiede i dati che l'applicazione deve recuperare da un dispositivo esterno. API
4. L'applicazione richiede dati da un API gateway REST API interno. Aggiunge un ID o un token di accesso come intestazione della richiesta.
5. Se si API dispone di una cache per la decisione di autorizzazione, restituisce la risposta precedente. Se la memorizzazione nella cache è disabilitata o non API dispone di una cache

- corrente, API Gateway passa i parametri della richiesta a un autorizzatore [Lambda basato su token](#).
6. La funzione Lambda invia una richiesta di autorizzazione a un archivio di criteri di autorizzazioni verificate con. [IsAuthorizedWithToken](#) API La funzione Lambda trasmette gli elementi di una decisione di autorizzazione:
 - a. Il token dell'utente come principale.
 - b. Il API metodo combinato con il API percorso, ad esempio `GetPhoto`, come azione.
 - c. Il termine `Application` come risorsa.
 7. Verified Permissions convalida il token. Per ulteriori informazioni su come vengono convalidati i token Amazon Cognito, consulta Authorization [with Amazon Verified Permissions nella Amazon Cognito Developer Guide](#).
 8. Verified Permissions valuta la richiesta di autorizzazione rispetto alle politiche del tuo archivio di politiche e restituisce una decisione di autorizzazione.
 9. L'autorizzatore Lambda restituisce una `Deny` risposta `Allow` o a `Gateway`. API
 - 10 API Restituisce dati o una `ACCESS_DENIED` risposta all'applicazione. L'applicazione elabora e visualizza i risultati della API richiesta.

Considerazioni per gli archivi API di policy collegati

Quando crei un archivio di policy API collegato a -link nella console Verified Permissions, stai creando un test per un'eventuale implementazione di produzione. Prima di passare alla produzione, stabilisci una configurazione fissa per il tuo pool API e quello di utenti. Considerate i seguenti fattori:

API gateway memorizza nella cache le risposte

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

I gruppi Amazon Cognito possono essere riutilizzati

Amazon Verified Permissions determina l'appartenenza al gruppo per gli utenti del pool di utenti in base alla `cognito:groups` dichiarazione contenuta nell'ID o nel token di accesso di un utente.

Il valore di questa dichiarazione è una matrice dei nomi descrittivi dei gruppi di pool di utenti a cui l'utente appartiene. Non è possibile associare i gruppi di pool di utenti a un identificatore univoco.

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

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

Verified Permissions rileva le tue informazioni in un determinato API momento: ti interroga solo API quando crei il tuo policy store. Quando lo schema o il nome delle API modifiche apportate, è necessario aggiornare il policy store e l'autorizzatore Lambda oppure creare un nuovo API policy store collegato. Verified Permissions ricava lo spazio dei nomi del policy store dal [nome](#) del tuo API

La funzione Lambda non ha alcuna configurazione VPC

La funzione Lambda che Verified Permissions crea per il tuo API autorizzatore non è connessa a un VPC Per impostazione predefinita. API scon accesso alla rete limitato ai soli utenti privati non VPCs possono comunicare con la funzione Lambda che autorizza le richieste di accesso con autorizzazioni verificate.

Verified Permissions distribuisce le risorse di autorizzazione in CloudFormation

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

Vedi [Passare alla produzione con AWS CloudFormation](#) per una panoramica delle risorse create da Verified Permissions.

Aggiungere il controllo degli accessi basato sugli attributi () ABAC

Una tipica sessione di autenticazione con un IdP restituisce ID e token di accesso. Puoi passare uno di questi tipi di token come token portante nelle richieste dell'applicazione al tuo API A seconda delle scelte effettuate al momento della creazione del policy store, Verified Permissions prevede uno dei due tipi di token. Entrambi i tipi contengono informazioni sull'appartenenza al gruppo dell'utente. Per

ulteriori informazioni sui tipi di token in Amazon Cognito, consulta [Using tokens with user pool](#) nella Amazon Cognito Developer Guide.

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

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

Token ID

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

Token di accesso

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

Vedi ad [Mappatura dei token del provider di identità allo schema](#) esempio le politiche che elaborano i reclami in ID e token di accesso.

Passare alla produzione con AWS CloudFormation

API-Linked Policy Storage è un modo per creare rapidamente un modello di autorizzazione per un API Gateway. API Sono progettati per fungere da ambiente di test per il componente di autorizzazione dell'applicazione. Dopo aver creato l'archivio delle politiche di test, dedica del tempo a perfezionare le politiche, lo schema e l'autorizzazione Lambda.

Potresti modificare l'architettura del tuo Policy Store API, richiedendo modifiche equivalenti allo schema e alle policy del tuo Policy Store. API-Linked Policy Store non aggiornano automaticamente il proprio schema dall'API architettura: Verified Permissions esegue un sondaggio solo nel momento

in cui viene creato un archivio di politiche. API Se le API modifiche sono sufficienti, potrebbe essere necessario ripetere la procedura con un nuovo policy store.

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

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

1. [AWS::VerifiedPermissions:PolicyStore](#): Copia lo schema nell'SchemaDefinitionoggetto. Esci " dai personaggi come \"
2. [AWS::VerifiedPermissions:: IdentitySource](#): Copia i valori dall'output del [GetIdentitySource](#) tuo Test Policy Store e modificali secondo necessità.
3. Uno o più tra [AWS::VerifiedPermissions: :Policy](#): Copia la tua dichiarazione di policy sull'Definitionoggetto. Fuggi " dai personaggi come \"
4. [AWS: :Lambda: :Function](#), [AWS::: :Role,IAM:AWS: :Policy,IAM::: :Authorizer,AWS ApiGateway: :Lambda AWS: :Permission](#): Copia il modello dalla scheda Modello dello stack distribuito da Verified Permissions quando hai creato il tuo policy store.

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

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyExamplePolicyStore": {
      "Type": "AWS::VerifiedPermissions::PolicyStore",
      "Properties": {
        "ValidationSettings": {
          "Mode": "STRICT"
        },
        "Description": "ApiGateway: PetStore/test",
        "Schema": {
```



```

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

```

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

Archivi di policy collegati alla risoluzione dei problemi API

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

Argomenti

- [Ho aggiornato la mia politica ma la decisione di autorizzazione non è cambiata](#)
- [Ho collegato l'autorizzatore Lambda al mio API ma non genera richieste di autorizzazione](#)
- [Ho ricevuto una decisione di autorizzazione inaspettata e desidero rivedere la logica di autorizzazione](#)
- [Voglio trovare i log del mio autorizzatore Lambda](#)
- [Il mio autorizzatore Lambda non esiste](#)
- [My API è in modalità privata VPC e non può richiamare l'autorizzatore](#)
- [Voglio elaborare attributi utente aggiuntivi nel mio modello di autorizzazione](#)
- [Desidero aggiungere nuove azioni, attributi del contesto dell'azione o attributi delle risorse](#)

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

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

Ho collegato l'autorizzatore Lambda al mio API ma non genera richieste di autorizzazione

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

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

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

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

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

Voglio trovare i log del mio autorizzatore Lambda

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

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

Il mio autorizzatore Lambda non esiste

Dopo aver completato la configurazione di un policy store API collegato, devi collegare l'autorizzatore Lambda al tuo. API Se non riesci a individuare l'autorizzatore nella console API Gateway, le risorse aggiuntive per il tuo policy store potrebbero non essere riuscite o non essere ancora state distribuite. API-gli archivi di policy collegati distribuiscono queste risorse in uno stack. AWS CloudFormation

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

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

My API è in modalità privata VPC e non può richiamare l'autorizzatore

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

Voglio elaborare attributi utente aggiuntivi nel mio modello di autorizzazione

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

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

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

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

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

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

Eliminazione degli archivi delle politiche

Puoi eliminare gli archivi di policy di Amazon Verified Permissions utilizzando AWS Management Console o il AWS CLI. L'eliminazione di un Policy Store elimina definitivamente lo schema e tutte le policy presenti nel Policy Store.

È possibile eliminare gli archivi delle politiche per i seguenti motivi:

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

AWS Management Console

Per eliminare un 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 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 JSON salvarli in una stringa a riga singola, e prefigurarla 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 JSON schema.

Note

L'esempio qui è una riga racchiusa 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 delle politiche utilizzando PutSchemaAPI. Per ulteriori informazioni, consulta [PutSchema](#) la Amazon Verified Permissions API Reference Guide.

Modifica degli schemi degli archivi di policy in modalità Visual

Quando selezioni Schema nella console Amazon Verified Permissions, la modalità visiva mostra i tipi di entità e le azioni che compongono lo schema. In questa visualizzazione di primo livello o dai dettagli di qualsiasi entità, puoi scegliere Modifica schema per iniziare ad apportare aggiornamenti allo 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 dei diagrammi. Sono disponibili due diagrammi:

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

Quando selezioni Schema nella console Amazon Verified Permissions, la JSONmodalità visualizza i tipi di entità e le azioni che compongono lo schema. Quando scegli Modifica schema, puoi iniziare ad aggiornare il JSON codice dello schema direttamente nell'JSONeditor. Durante gli aggiornamenti, noterai che l'JSONeditor convalida il codice in base alla JSON sintassi e identifica errori e avvisi durante la modifica, facilitando la ricerca rapida dei problemi. Inoltre, non devi preoccuparti della formattazione diJSON, basta scegliere Formato JSON dopo aver effettuato gli aggiornamenti e il formato verrà aggiornato in base alla formattazione prevista. 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 JSONla modalità, quindi scegli Modifica schema.
4. Inserisci il contenuto dello JSON schema nel campo Contenuto. Non è possibile salvare gli aggiornamenti dello schema finché non si risolvono tutti gli errori di sintassi. Puoi scegliere Formato JSON per formattare la JSON sintassi dello schema con la spaziatura e l'indentazione consigliate.
5. Scegli Save changes (Salva modifiche).

Attivazione della modalità di convalida delle policy di Amazon Verified Permissions

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

Important

Quando si attiva la convalida delle policy, tutti i tentativi di creare o aggiornare una policy o un modello di policy vengono convalidati in base allo schema presente nell'archivio delle policy. Verified Permissions rifiuta 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. Scegli Save changes (Salva modifiche).

AWS CLI

Per impostare la modalità di convalida per un archivio di politiche

È possibile modificare la modalità di convalida per un policy store utilizzando l'[UpdatePolicyStore](#) operazione e specificando un valore diverso per il parametro [ValidationSettings](#)

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

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:

- Utilizzate valori come identificatori univoci universali (UUIDs) per tutti gli identificatori principali e di risorse.

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

Se usi 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- «,//alice EXAMPLE11111`

- 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

Argomenti

- [Formattazione delle entità all'interno delle politiche di Amazon Verified Permissions](#)
- [Creazione di politiche statiche per le autorizzazioni verificate di Amazon](#)
- [Modifica delle politiche statiche di Amazon Verified Permissions](#)

- [Utilizzo del banco di prova Amazon Verified Permissions](#)
- [Esempi di politiche di Amazon Verified Permissions](#)

Formattazione delle entità all'interno delle politiche di 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 JSON formattazione delle entità in Verified Permissions differisce da Cedar nei seguenti modi:

- In Verified Permissions, un JSON oggetto deve avere tutte le sue coppie chiave-valore racchiavette in un oggetto con il nome di. `JSON Record`
- Un JSON elenco in Autorizzazioni verificate deve essere racchiuso in una coppia JSON chiave-valore in cui il nome della chiave è `Set` e il valore è l'elenco originale di Cedar. `JSON`
- Per i nomi `Boolean` di tipo e `StringLong`, ogni coppia chiave-valore di Cedar viene sostituita da un oggetto in Verified Permissions. `JSON` Il nome dell'oggetto è il nome della chiave originale. All'interno dell'`JSON` oggetto, c'è una coppia chiave-valore in cui il nome della chiave è il nome del tipo del valore scalare (`StringLong`, `oBoolean`) 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
<code>uid</code>	<code>Identifier</code>
<code>type</code>	<code>EntityType</code>
<code>id</code>	<code>EntityId</code>
<code>attrs</code>	<code>Attributes</code>
<code>parents</code>	<code>Parents</code>

Example - Elenchi

Gli esempi seguenti mostrano come un elenco di entità viene espresso rispettivamente in Cedar e Verified Permissions.

Cedar

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

Verified Permissions

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



```
}  
]  
}
```

Example - Valutazione delle politiche

Gli esempi seguenti mostrano come le entità sono formattate per la valutazione di una politica in una richiesta di autorizzazione in Cedar e Verified Permissions, rispettivamente.

Cedar

```
[  
  {  
    "uid": {  
      "type": "PhotoApp::User",  
      "id": "alice"  
    },  
    "attrs": {  
      "age": 25,  
      "name": "alice",  
      "userId": "123456789012"  
    },  
    "parents": [  
      {  
        "type": "PhotoApp::UserGroup",  
        "id": "alice_friends"  
      },  
      {  
        "type": "PhotoApp::UserGroup",  
        "id": "AVTeam"  
      }  
    ]  
  },  
  {  
    "uid": {  
      "type": "PhotoApp::Photo",  
      "id": "vacationPhoto.jpg"  
    },  
    "attrs": {  
      "private": false,  
      "account": {  
        "__entity": {
```

```

        "type": "PhotoApp::Account",
        "id": "ahmad"
      }
    },
    "parents": []
  },
  {
    "uid": {
      "type": "PhotoApp::UserGroup",
      "id": "alice_friends"
    },
    "attrs": {},
    "parents": []
  },
  {
    "uid": {
      "type": "PhotoApp::UserGroup",
      "id": "AVTeam"
    },
    "attrs": {},
    "parents": []
  }
]

```

Verified Permissions

```

[
  {
    "Identifier": {
      "EntityType": "PhotoApp::User",
      "EntityId": "alice"
    },
    "Attributes": {
      "age": {
        "Long": 25
      },
      "name": {
        "String": "alice"
      },
      "userId": {
        "String": "123456789012"
      }
    }
  }
]

```

```
    },
    "Parents": [
      {
        "EntityType": "PhotoApp::UserGroup",
        "EntityId": "alice_friends"
      },
      {
        "EntityType": "PhotoApp::UserGroup",
        "EntityId": "AVTeam"
      }
    ]
  },
  {
    "Identifier": {
      "EntityType": "PhotoApp::Photo",
      "EntityId": "vacationPhoto.jpg"
    },
    "Attributes": {
      "private": {
        "Boolean": false
      },
      "account": {
        "EntityIdentifier": {
          "EntityType": "PhotoApp::Account",
          "EntityId": "ahmad"
        }
      }
    },
    "Parents": []
  },
  {
    "Identifier": {
      "EntityType": "PhotoApp::UserGroup",
      "EntityId": "alice_friends"
    },
    "Parents": []
  },
  {
    "Identifier": {
      "EntityType": "PhotoApp::UserGroup",
      "EntityId": "AVTeam"
    },
    "Parents": []
  }
}
```

]

Creazione di politiche statiche per le autorizzazioni verificate di Amazon

È possibile creare una politica statica per consentire o vietare ai responsabili 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/>. 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. Scegli Next (Successivo).
9. Nella sezione Politica, consulta la tua politica Cedar. Puoi scegliere Formato per formattare la sintassi della tua politica con la spaziatura e l'indentazione consigliate. Per ulteriori informazioni, vedere [Costruzione delle politiche di base in Cedar nella Guida di riferimento al linguaggio delle politiche Cedar](#).
10. Nella sezione Dettagli, digita una descrizione facoltativa della politica.
11. Scegli Create Policy (Crea policy).

AWS CLI

Per creare una politica statica

È possibile creare una politica statica utilizzando l'[CreatePolicy](#) operazione. 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

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

È possibile modificare i seguenti elementi di una politica statica:

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

Non è possibile modificare i seguenti elementi di una politica statica:

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

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` o `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'[UpdatePolicy](#) operazione. 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 PSEXAMPLEabcdefgh111111

{
  "createdDate": "2023-06-12T20:33:37.382907+00:00",
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
  "policyId": "SPEXAMPLEabcdefgh111111",
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "policyType": "STATIC",
  "principal": {
    "entityId": "janeFriends",
    "entityType": "UserGroup"
  },
  "resource": {
    "entityId": "vacationFolder",
    "entityType": "Album"
  }
}
```

}

Utilizzo del banco di prova Amazon Verified Permissions

Utilizza il banco di prova delle autorizzazioni verificate per testare e risolvere i problemi delle politiche di autorizzazione verificate eseguendo richieste di [autorizzazione](#) su di esse. Il test bench 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 visiva alla JSONmodalità 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 Guida di riferimento al linguaggio delle politiche Cedar](#).

Note

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

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

Visual mode

Note

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

Per testare le politiche in modalità Visual

1. Apri la console delle autorizzazioni verificate 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. Digita 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 JSON la modalità.
4. Nella sezione Dettagli della richiesta, se hai definito uno schema, scegli il Principal che interviene tra i tipi principali del tuo schema. Digita un identificatore per il principale nella casella di testo.

Se non avete definito uno schema, digitate il principale nella casella di testo Principal taking action.

5. Se hai definito uno schema, scegli la risorsa tra i tipi di risorse presenti nello schema. Digitate un identificatore per la risorsa nella casella di testo.

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

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

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

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

Esempi di politiche di Amazon Verified Permissions

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 policy Cedar, vedere [Basic Policy building 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](#)
- [Utilizza la notazione tra parentesi per fare riferimento agli attributi del token](#)
- [Utilizza la notazione a punti per fare riferimento agli attributi](#)
- [Riflette gli attributi del token Amazon Cognito ID](#)
- [Riflette gli OIDC attributi del token ID](#)
- [Riflette gli attributi del token di accesso di Amazon Cognito](#)
- [Riflette gli attributi del token di OIDC accesso](#)

Consente l'accesso a singole entità

L'esempio seguente mostra come creare una politica che alice consenta all'utente di visualizzare la fotoVacationPhoto94.jpg.

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

Consente l'accesso a gruppi di entità

L'esempio seguente mostra come creare una politica che consenta a tutti i membri del gruppo alice_friends di visualizzare la fotoVacationPhoto94.jpg.

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

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

L'esempio seguente mostra come 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"  
);
```

L'esempio seguente mostra come 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"  
);
```

L'esempio seguente mostra come creare una politica che consenta le autorizzazioni per l'utente `alice` nell'album `alice_vacation`, dove `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"  
);
```

L'esempio seguente mostra come è possibile creare una politica che consenta le autorizzazioni per l'utente `alice` nell'album `alice_vacation`, dove `viewer` trova un gruppo definito nella gerarchia dello schema che contiene l'autorizzazione a visualizzare e commentare una foto. L'editautorizzazione `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

L'esempio seguente 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"  
);
```

L'esempio seguente mostra come creare una politica che consenta all'utente di `alice` elencare tutti gli album dell'`janeaccount`, 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"  
);
```

L'esempio seguente mostra come 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 degli accessi basato sugli attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. Le autorizzazioni verificate consentono di associare 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.

L'esempio seguente mostra come è possibile creare una politica che consenta a qualsiasi preside del HardwareEngineering dipartimento con un livello di lavoro 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  
};
```

L'esempio seguente mostra come creare una politica che consenta all'utente di `alice` visualizzare qualsiasi risorsa di tipo di `fileJPEG`.

```
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. L'esempio seguente mostra come creare una politica che consenta all'utente `alice` 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. L'esempio seguente mostra come 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"
};
```

L'esempio seguente 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
};
```

L'esempio seguente 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  
};
```

L'esempio seguente 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.

L'esempio seguente mostra come 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
};
```

L'esempio seguente 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
};
```

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

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

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere [Mappatura dei token del provider di identità allo schema](#)

```
permit (
  principal in MyCorp::UserGroup::"us-west-2_EXAMPLE|MyUserGroup",
  action,
  resource
) when {
```

```
principal["cognito:username"] == "alice" &&
principal["custom:employmentStoreCode"] == "petstore-dallas" &&
principal has email && principal.email == "alice@example.com" &&
context["ip-address"] like "192.0.2.*"
};
```

Utilizza la notazione a punti per fare riferimento agli attributi

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

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere [Mappatura dei token del provider di identità allo schema](#)

```
permit(principal, action, resource)
when {
    principal.cognito.username == "alice" &&
    principal.custom.employmentStoreCode == "petstore-dallas" &&
    principal.tenant == "x11app-tenant-1" &&
    principal has email && principal.email == "alice@example.com"
};
```

Riflette gli attributi del token Amazon Cognito ID

L'esempio seguente mostra come creare una policy che faccia riferimento agli attributi del token ID da Amazon Cognito.

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, consulta [Mappatura dei token del provider di identità allo schema](#)

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

Riflette gli OIDC attributi del token ID

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

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere [Mappatura dei token del provider di identità allo schema](#)

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

Riflette gli attributi del token di accesso di Amazon Cognito

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

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, consulta [Mappatura dei token del provider di identità allo schema](#)

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

Riflette gli attributi del token di OIDC accesso

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

Per ulteriori informazioni sull'utilizzo degli attributi dei token nelle politiche in Autorizzazioni verificate, vedere [Mappatura dei token del provider di identità allo schema](#)

```
permit(  

```

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

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

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

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

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

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

A differenza delle politiche statiche, le politiche collegate ai modelli sono dinamiche. Prendiamo l'esempio precedente, se si dovesse rimuovere l'Commentazione dal modello di policy, qualsiasi policy collegata o basata su quel modello verrebbe aggiornata di conseguenza e i principi specificati nelle policy non sarebbero più in grado di commentare le risorse corrispondenti.

Per altri esempi di policy collegati ai modelli, consulta [Esempio di politiche collegate a modelli di Amazon Verified Permissions](#)

Creazione di modelli di policy Amazon Verified Permissions

È possibile creare modelli di policy in Autorizzazioni verificate utilizzando il AWS Management Console, il AWS CLI, o il AWSSDKs. I modelli di policy consentono di definire una policy una sola

volta e di utilizzarla con più principi e risorse. Una volta creato un modello di policy, è possibile creare policy collegate al modello per utilizzare i modelli di policy con principi e risorse specifici. Per ulteriori informazioni, consulta [Creazione di politiche collegate ai modelli di Amazon Verified Permissions](#).

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 `?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'[CreatePolicyTemplate](#) operazione. 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 politiche collegate ai modelli di Amazon Verified Permissions

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

Per esempi di policy collegati ai modelli, consulta. [Esempio di politiche collegate a modelli di Amazon Verified Permissions](#)

AWS Management Console

Per creare una policy collegata a un 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,

`digitateGroup` : "AdminUsers". Se `digitateAdminUsers`, 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 a un modello creando un'istanza di un modello di policy

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

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

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

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

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

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

```
$ aws verifiedpermissions create-policy \
```



```
--definition file://definition.txt
--policy-store-id PSEXAMPLEabcdefg111111
{
  "createdDate": "2023-05-22T18:57:53.298278+00:00",
  "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",
  "policyId": "TPEXAMPLEabcdefg111111",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyType": "TEMPLATELINKED",
  "principal": {
    "entityId": "alice",
    "entityType": "PhotoFlash::User"
  },
  "resource": {
    "entityId": "VacationPhoto94.jpg",
    "entityType": "PhotoFlash::Photo"
  }
}
```

Modifica dei modelli di policy di Amazon Verified Permissions

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

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 selezionando il pulsante di opzione accanto a un modello di politica e quindi scegliendo Elimina. Scegli OK per confermare l'eliminazione del modello di policy.

AWS CLI

Per aggiornare un modello di policy

È possibile creare una politica statica utilizzando l'[UpdatePolicy](#) operazione. 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"  
}
```

Esempio di politiche collegate a modelli di Amazon Verified Permissions

Quando crei un archivio delle politiche in Autorizzazioni verificate utilizzando il metodo Sample policy store, il tuo archivio delle politiche viene creato con politiche predefinite, modelli di policy e uno

schema per il progetto di esempio che hai scelto. I seguenti esempi di policy collegati al modello Verified Permissions possono essere utilizzati con gli archivi di policy di esempio e i rispettivi criteri, modelli di policy e schemi.

PhotoFlashesempi

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

Note

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

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

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

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

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

```
permit (  
  principal in PhotoFlash::FriendGroup::"Jane::MySchoolFriends",
```

```
action in PhotoFlash::Action::"SharePhotoLimitedAccess",
resource in PhotoFlash::Photo::"VacationPhoto94.jpg"
);
```

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

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

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

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

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

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

DigitalPetStore esempi

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

TinyToDoesempi

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

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

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

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

Utilizzo di Amazon Verified Permissions con provider di identità

Una fonte di identità è una rappresentazione di un provider di identità esterno (IdP) in Amazon Verified Permissions. Le fonti di identità forniscono informazioni su un utente che si è autenticato con un IdP che ha una relazione di fiducia con il tuo policy store. Quando l'applicazione effettua una richiesta di autorizzazione con un token proveniente da una fonte di identità, il policy store può prendere decisioni di autorizzazione sulla base delle proprietà dell'utente e delle autorizzazioni di accesso. Le fonti di identità Verified Permissions migliorano l'autorizzazione grazie a una connessione diretta all'archivio centrale delle identità e al servizio di autenticazione.

Puoi utilizzare i provider di identità [OpenID Connect \(OIDC IdPs\)](#) con autorizzazioni verificate. L'applicazione può generare richieste di autorizzazione con OIDC identità (ID) o token JSON web di accesso (). JWTs Con i token ID, Verified Permissions legge le dichiarazioni degli utenti IDs e degli attributi come principi per il controllo degli accessi basato sugli attributi (). ABAC [Con i token di accesso, Verified Permissions legge l'utente come mandante e le altre attestazioni come contesto.](#) IDs Con entrambi i tipi di token, puoi mappare un claim come se fosse groups un gruppo principale e creare policy che valutino il controllo degli accessi basato sui ruoli (). RBAC

Puoi aggiungere un pool di utenti Amazon Cognito o un OIDC IdP OpenID Connect () personalizzato come fonte di identità.

Argomenti

- [Utilizzo delle fonti di identità di Amazon Cognito](#)
- [Lavorare con le fonti di identità OIDC](#)
- [Convalida del cliente e del pubblico](#)
- [Autorizzazione lato client per JWTs](#)
- [Creazione di fonti di identità Amazon Verified Permissions](#)
- [Modifica delle fonti di identità di Amazon Verified Permissions](#)
- [Mappatura dei token del provider di identità allo schema](#)

Utilizzo delle fonti di identità di Amazon Cognito

Verified Permissions lavora a stretto contatto con i pool di utenti di Amazon Cognito. Amazon Cognito JWTs ha una struttura prevedibile. Verified Permissions riconosce questa struttura e trae il massimo

vantaggio dalle informazioni in essa contenute. Ad esempio, è possibile implementare un modello di autorizzazione access control (RBAC) basato sui ruoli con token ID o token di accesso.

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

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

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

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

1. Dichiarazioni relative al nome utente e al gruppo con prefisso cognito:
2. [Attributi utente personalizzati](#) con un `custom: prefix`
3. Affermazioni personalizzate aggiunte in fase di esecuzione
4. OIDCaffermazioni standard come `sub` e `email`

Tratteremo queste affermazioni in dettaglio e spieghiamo come gestirle nelle politiche sulle autorizzazioni verificate, in [Mappatura dei token del provider di identità allo schema](#).

Important

Sebbene sia possibile revocare i token Amazon Cognito prima della scadenzaJWTs, sono considerati risorse stateless autonome con firma e validità. I servizi conformi [al JSON Web](#)

[Token RFC 7519 dovrebbero convalidare i token](#) da remoto e non sono tenuti a convalidarli con l'emittente. Ciò significa che è possibile che Verified Authorizations conceda l'accesso in base a un token che è stato 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.

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

Note

Se un token contiene un'attestazione con un `custom:` prefisso `cognito:` or e un nome di attestazione con valore letterale `cognito ocustom`, una richiesta di autorizzazione con [IsAuthorizedWithToken](#) un. `ValidationException`

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

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


Per ulteriori informazioni sulla mappatura delle rivendicazioni, consulta [Mappatura dei token ID allo schema](#). Per ulteriori informazioni sull'autorizzazione per gli utenti di Amazon Cognito, consulta [Authorization with Amazon Verified Permissions nella Amazon Cognito Developer Guide](#).

Lavorare con le fonti di identità OIDC

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

Una nuova fonte di OIDC identità richiede le seguenti informazioni:

- L'emittente. URL Verified Permissions deve essere in grado di rilevare un `.well-known/openid-configuration` endpoint in questo modo. URL
- Il tipo di token che desideri utilizzare nelle richieste di autorizzazione. In questo caso, hai scelto Identity token.
- Il tipo di entità utente che desideri associare alla fonte della tua identità, ad esempio `MyCorp::User`.
- Il tipo di entità di gruppo che desideri associare alla tua fonte di identità, ad esempio `MyCorp::UserGroup`.
- Un esempio di token ID o una definizione delle attestazioni nel token ID.
- Il prefisso che desideri applicare all'entità IDs utente e di gruppo. Alla fine CLI API, puoi scegliere questo prefisso. Negli archivi di policy creati con l'opzione Configura con API Gateway e un'origine di identità o l'opzione di configurazione guidata, Verified Permissions assegna un prefisso al nome dell'emittente meno, ad esempio. `https://MyCorp::User::"auth.example.com|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"`

L'autorizzazione con fonti di OIDC identità utilizza le stesse API operazioni delle fonti di identità del pool di utenti: and. [IsAuthorizedWithTokenBatchIsAuthorizedWithToken](#)

L'esempio seguente mostra come è possibile creare una politica che consenta l'accesso ai report di fine anno ai dipendenti del reparto contabilità, abbiano una classificazione riservata e non lavorino in un ufficio secondario. Verified Permissions ricava questi attributi dalle attestazioni contenute nel token ID del preside.

```
permit(  
    principal in MyCorp::UserGroup::"MyOIDCProvider|Accounting",
```

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

Convalida del cliente e del pubblico

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

Un ID client è un identificatore associato a un'OIDC applicazione OAuth o configurata con il provider, ad esempio. `1example23456789` Un pubblico è un URL percorso associato al relying party, o destinazione, previsto per l'applicazione di destinazione, ad esempio. `https://myapplication.example.com` L'audaffermazione non è sempre associata al pubblico.

Verified Permissions esegue la convalida dell'identità, dell'origine, del pubblico e del client come segue:

Amazon Cognito

I token ID Amazon Cognito hanno un'audaffermazione che contiene l'ID client dell'[app](#). I token di accesso hanno un `client_id` claim che contiene anche l'ID client dell'app.

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

OIDC

OIDC I token ID hanno un'audattestazione che contiene un elenco di clienti. IDs I token di accesso hanno un `aud` claim che contiene il pubblico URL del token. I token di accesso hanno anche un'`client_id` attestazione che contiene l'ID client desiderato.

Puoi inserire uno o più valori per la convalida dell'Audience con un OIDC provider. Quando scegli un tipo di token o token ID, Verified Permissions convalida l'ID cliente, verificando che almeno un membro del cliente indicato IDs nel aud claim corrisponda a un valore di convalida del pubblico.

Verified Permissions convalida il pubblico per i token di accesso, verificando che l'audattestazione corrisponda a un valore di convalida del pubblico. Questo valore del token di accesso deriva principalmente dal claim, ma può provenire dal aud claim cid or client_id se non esiste alcun claim. aud Rivolgiti al tuo IdP per conoscere la dichiarazione e il formato del pubblico corretti.

Un esempio di valore di convalida del pubblico del token ID è. 1example23456789

Un esempio di valore di convalida del pubblico del token di accesso è. https://myapplication.example.com

Autorizzazione lato client per JWTs

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

Questo esempio mostra come è possibile chiamare le autorizzazioni verificate da un OIDC IdP.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 = "PSEXAMPLEEabcdefg111111"; // set your own policy store id

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

    return authResult;
}

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

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

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

¹ Questo esempio di codice utilizza la [aws-jwt-verify](#) libreria per la verifica JWTs firmata da - compatible. OIDC IdPs

Creazione di fonti di identità Amazon Verified Permissions

La procedura seguente aggiunge una fonte di identità a un archivio di policy esistente. Dopo aver aggiunto la fonte di identità, è necessario [aggiungere gli attributi allo schema](#).

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

Note

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

È possibile omettere il tipo di entità principale quando si crea una fonte di identità [create-identity-source](#) nelle AWS CLI o [CreateIdentitySource](#) nelle Autorizzazioni API verificate. Tuttavia, un tipo di entità vuoto crea una fonte di identità con un tipo di entità di AWS: `Cognito`. Questo nome di entità non è compatibile con lo schema dell'archivio delle politiche. Per integrare 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.

Argomenti

- [Fonte di identità Amazon Cognito](#)
- [OIDC fonte di identità](#)

Fonte di identità Amazon Cognito

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. Nei dettagli del pool di utenti di Cognito, seleziona Regione AWS e inserisci l'ID del pool di utenti per la tua origine di identità.
5. Nella configurazione principale, scegli un tipo principale per l'origine dell'identità. Le identità dei pool di utenti Amazon Cognito connessi verranno mappate sul tipo principale selezionato.
6. Nella configurazione del gruppo, seleziona Usa il gruppo Cognito se desideri mappare il claim del pool `cognito:groups` di utenti. Scegli un tipo di entità che sia padre del tipo principale.
7. In Convalida dell'applicazione client, scegli se convalidare l'applicazione client. IDs

- Per convalidare l'applicazione clientIDs, scegli Accetta solo token con l'applicazione client corrispondente. IDs Scegli Aggiungi nuovo ID dell'applicazione client per ogni ID dell'applicazione client da convalidare. Per rimuovere un ID dell'applicazione client che è stato aggiunto, scegli Rimuovi accanto all'ID dell'applicazione client.
 - Scegliete Non convalidare l'applicazione client IDs se non desiderate convalidare l'applicazione client. IDs
8. Scegli Crea origine di identità.
 9. 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 del provider di identità allo schema](#)

Quando crei un [archivio di policy API collegato](#), Verified Permissions interroga il tuo pool di utenti per verificare gli attributi utente e crea uno schema in cui il tipo principale viene popolato con gli attributi del pool di utenti.

AWS CLI

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

Puoi creare una fonte di identità utilizzando l'[CreateIdentitySource](#) operazione. 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"],
    "groupConfiguration": {
      "groupEntityType": "MyCorp::UserGroup"
    }
  }
}
```

```
}
```

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 del provider di identità allo schema](#)

Quando crei un [archivio di policy API collegato](#), Verified Permissions interroga il tuo pool di utenti per verificare gli attributi utente e crea uno schema in cui il tipo principale viene popolato con gli attributi del pool di utenti.

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](#).

OIDC fonte di identità

AWS Management Console

Per creare una fonte di identità OpenID Connect (OIDC)

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. Scegli OIDCProvider esterno.
5. In Emittente URL, inserisci il nome URL dell'OIDCemittente. Questo è l'endpoint del servizio che fornisce, ad esempio, il server di autorizzazione, le chiavi di firma e altre informazioni sul provider. `https://auth.example.com` L'emittente URL deve ospitare un documento di OIDC scoperta presso `/.well-known/openid-configuration`
6. In Tipo di token, scegli il tipo di token OIDC JWT che desideri che la tua richiesta invii per l'autorizzazione. Per ulteriori informazioni, consulta [Mappatura dei token del provider di identità allo schema](#).
7. In Attestazioni utente e di gruppo, scegli un'entità utente e un'attestazione utente per l'origine dell'identità. L'entità Utente è un'entità nel tuo archivio delle politiche a cui desideri fare riferimento agli utenti del tuo OIDC provider. L'attestazione Utente è in genere sub un'attestazione derivante dal tuo ID o token di accesso che contiene l'identificatore univoco dell'entità da valutare. Le identità dell'OIDCIdP connesso verranno mappate al tipo principale selezionato.
8. In Attestazioni utente e di gruppo, scegli un'entità di gruppo e un'attestazione di gruppo come origine dell'identità. L'entità del Gruppo è la capogruppo dell'entità Utente. Le rivendicazioni di gruppo vengono mappate su questa entità. L'attestazione di gruppo è in genere groups un'attestazione derivante dall'ID o dal token di accesso che contiene una stringa o una stringa di nomi di gruppi di utenti delimitata da spazi per l'entità da valutare. JSON Le identità dell'OIDCIdP connesso verranno mappate al tipo principale selezionato.
9. In Audience validation, inserisci il client IDs o il pubblico URLs che desideri che il tuo policy store accetti nelle richieste di autorizzazione, se presenti.
10. Scegli Crea fonte di identità.
11. Aggiorna 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 si desidera fare riferimento nelle politiche Cedar. Per ulteriori informazioni sulla mappatura degli attributi del token Amazon Cognito agli attributi principali di Cedar, consulta. [Mappatura dei token del provider di identità allo schema](#)

Quando crei un [archivio di policy API collegato](#), Verified Permissions interroga il tuo pool di utenti per verificare gli attributi utente e crea uno schema in cui il tipo principale viene popolato con gli attributi del pool di utenti.

AWS CLI

Per creare una fonte di identità OIDC

È possibile creare una fonte di identità utilizzando l'[CreateIdentitySource](#) operazione. 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 di un OIDC IdP da utilizzare con il `--configuration` parametro del `create-identity-source` comando. Questo esempio crea una fonte di OIDC identità per i token ID.

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

Il `config.txt` file seguente contiene i dettagli di un OIDC IdP da utilizzare con il `--configuration` parametro del `create-identity-source` comando. Questo esempio crea una fonte di OIDC identità per i token di accesso.

```
{
  "openIdConnectConfiguration": {
    "issuer": "https://auth.example.com",
    "tokenSelection": {
      "accessTokenOnly": {
        "audiences": ["https://auth.example.com"],
        "principalIdClaim": "sub"
      },
    },
  },
}
```

```
    "entityIdPrefix": "MyOIDCProvider",
    "groupConfiguration": {
      "groupClaim": "groups",
      "groupEntityType": "MyCorp::UserGroup"
    }
  }
}
```

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": "ISEXAMPLEabcdefghijklmnop111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111"
}
```

Prima di poter fare riferimento agli attributi estratti dall'identità o dai token di accesso nelle politiche Cedar, è necessario aggiornare lo schema per rendere Cedar consapevole del tipo di principale creato dalla 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 del provider di identità allo schema](#)

Quando crei un [archivio di policy API collegato](#), Verified Permissions interroga il tuo pool di utenti per verificare gli attributi utente e crea uno schema in cui il tipo principale viene popolato con gli attributi del pool di utenti.

Modifica delle fonti di identità di Amazon Verified Permissions

Puoi modificare alcuni parametri della tua fonte di identità dopo averla creata. Se lo schema del policy store corrisponde agli attributi di origine dell'identità, tieni presente che devi aggiornare lo schema separatamente per riflettere le modifiche apportate alla tua fonte di identità.

Argomenti

- [Fonte di identità dei pool di utenti di Amazon Cognito](#)

- [Fonte di identità OpenID Connect \(OIDC\)](#)

Fonte di identità dei pool di utenti di Amazon Cognito

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. Nei dettagli del pool di utenti di Cognito, seleziona Regione AWS e digita l'ID del pool di utenti per la tua origine di identità.
6. Nei dettagli del principale, puoi aggiornare il tipo di Principal per la fonte dell'identità. Le identità dei pool di utenti Amazon Cognito connessi verranno mappate sul tipo principale selezionato.
7. Nella configurazione del gruppo, seleziona Usa il gruppo Cognito se desideri mappare il claim del pool `cognito:groups` di utenti. Scegli un tipo di entità che sia padre del tipo principale.
8. In Convalida dell'applicazione client, scegli se convalidare l'applicazione client. IDs
 - Per convalidare l'applicazione client IDs, scegli Accetta solo token con l'applicazione client corrispondente. IDs Scegli Aggiungi nuovo ID dell'applicazione client per ogni ID dell'applicazione client da convalidare. Per rimuovere un ID dell'applicazione client che è stato aggiunto, scegli Rimuovi accanto all'ID dell'applicazione client.
 - Scegliete Non convalidare l'applicazione client IDs se non desiderate convalidare l'applicazione client. IDs
9. Scegli Save changes (Salva modifiche).
10. Se hai modificato il tipo principale per l'origine dell'identità, devi aggiornare lo schema in modo che rifletta correttamente il tipo principale aggiornato.

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

AWS CLI

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

Puoi aggiornare una fonte di identità utilizzando l'[UpdateIdentitySource](#) operazione. 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"],
    "groupConfiguration": {
      "groupEntityType": "MyCorp::UserGroup"
    }
  }
}
```

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 l'origine dell'identità, devi aggiornare lo schema in modo che rifletta correttamente il tipo principale aggiornato.

Fonte di identità OpenID Connect (OIDC)

AWS Management Console

Per aggiornare una fonte di OIDC identità

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. Nei dettagli OIDC del provider, modifica l'emittente in base URL alle esigenze.
6. In Map token claim to schema, modificate le associazioni tra le attestazioni utente e di gruppo e i tipi di entità del Policy Store, se necessario. Dopo aver modificato i tipi di entità, è necessario aggiornare le politiche e gli attributi dello schema per applicarli ai nuovi tipi di entità.
7. Nella convalida dell'audience, aggiungi o rimuovi i valori di audience che desideri applicare.
8. Scegli Save changes (Salva modifiche).

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

AWS CLI

Per aggiornare una fonte di OIDC identità

È possibile aggiornare una fonte di identità utilizzando l'[UpdateIdentitySource](#) operazione. L'esempio seguente aggiorna l'origine di identità specificata per utilizzare un OIDC provider 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`

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

```

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

```

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 l'origine dell'identità, devi aggiornare lo schema in modo che rifletta correttamente il tipo principale aggiornato.

Mappatura dei token del provider di identità allo schema

Potresti scoprire di voler aggiungere una fonte di identità a un archivio di politiche e le dichiarazioni del provider di mappe allo schema del tuo archivio di politiche. Puoi automatizzare questo processo o aggiornare lo schema manualmente. Dopo aver mappato i token allo schema, è possibile creare politiche che vi fanno riferimento.

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

- Quando è possibile compilare automaticamente gli attributi in uno schema di policy store
- Come utilizzare Amazon Cognito e le attestazioni dei OIDC token nelle tue politiche di autorizzazione verificata

- Come creare manualmente uno schema per una fonte di identità

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

- Hai creato un policy store o un policy store vuoto a partire da un esempio.
- Desiderate estendere l'uso dei token di accesso oltre il controllo degli accessi basato sui ruoli (). RBAC
- Crei archivi di policy con Autorizzazioni verificate RESTAPI, un AWS SDK, o il. AWS CDK

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

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

Argomenti

- [Cose da sapere sulla mappatura degli schemi](#)
- [Mappatura dei token ID allo schema](#)
- [Mappatura dei token di accesso](#)
- [Notazione alternativa per le dichiarazioni delimitate da due punti di Amazon Cognito](#)

Cose da sapere sulla mappatura degli schemi

La mappatura degli attributi differisce tra i tipi di token

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

Gli attributi di origine dell'identità non sono obbligatori

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

RBAC non richiede attributi nello schema

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

Le opzioni Configura con API gateway e un'origine di identità e Configurazione guidata nella console Autorizzazioni verificate assegnano le attestazioni dei token ID allo schema. Questo non è il caso delle rivendicazioni relative ai token di accesso. Per aggiungere rivendicazioni di token di accesso non di gruppo allo schema, è necessario modificare lo schema in JSON modalità e aggiungere attributi. [commonTypes](#) Per ulteriori informazioni, consulta [Mappatura dei token di accesso](#).

OIDC groups claim supporta più formati

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

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

Note

Verified Permissions interpreta ogni stringa contenuta in un'affermazione relativa ai gruppi separati da spazi come un gruppo separato. Per interpretare il nome di un gruppo con un carattere di spazio come un singolo gruppo, sostituisci o rimuovi lo spazio nell'attestazione. Ad esempio, formatta un gruppo denominato My Group comeMyGroup.

Scegli un tipo di token

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

Cedar parser richiede parentesi per alcuni caratteri

Le politiche in genere fanno riferimento agli attributi dello schema in un formato simile.

`principal.username` Nel caso della maggior parte dei caratteri non alfanumerici come `.`, `/` che potrebbero apparire nei nomi delle rivendicazioni dei token, Verified Permissions non è in grado di analizzare un valore di condizione come `o.principal.cognito:username context.ip-address` È invece necessario formattare queste condizioni con la notazione tra parentesi nel formato `o, rispettivamente. principal["cognito:username"] context["ip-address"]` Il carattere

di sottolineatura `_` è un carattere valido nei nomi delle rivendicazioni e rappresenta l'unica eccezione non alfanumerica a questo requisito.

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

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

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

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

```
    "User"  
  ]  
}  
}
```

Per un esempio di politica che verrà convalidata rispetto a questo schema, vedi [Utilizza la notazione tra parentesi per fare riferimento agli attributi del token](#).

Mappatura dei token ID allo schema

Verified Permissions elabora le dichiarazioni relative ai token ID come attributi dell'utente: nomi e titoli, appartenenza al gruppo, informazioni di contatto. I token ID sono molto utili in un modello di autorizzazione access control () basato sugli attributi. ABAC Se desideri che Verified Permissions analizzi l'accesso alle risorse in base a chi effettua la richiesta, scegli i token ID come fonte di identità.

Token ID Amazon Cognito

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

Affermazioni utili nei token ID di Amazon Cognito

cognito:username e preferred_username

Varianti del nome utente.

sub

L'identificatore utente univoco dell'utente () UUID

Affermazioni con un *custom:* prefisso

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

Affermazioni standard

OIDCReclami standard come *email* e *phone_number*. Per ulteriori informazioni, consulta [Dichiarazioni standard](#) in OpenID Connect Core 1.0 che incorporano il set di errata 2.

cognito:groups

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

Reclami transitori

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

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

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

Il seguente esempio di token ID ha ciascuno dei quattro tipi di attributi. Include l'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_EXAMPLE",
  "cognito:username": "alice",
  "custom:employmentStoreCode": "petstore-dallas",
  "origin_jti": "5b9f50a3-05da-454a-8b99-b79c2349de77",
  "aud": "1example23456789",
  "event_id": "0ed5ad5c-7182-4ecf-XXX",
  "token_use": "id",
```

```
"auth_time": 1687885407,  
"department": "engineering",  
"exp": 1687889006,  
"iat": 1687885407,  
"tenant": "x11app-tenant-1",  
"jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",  
"email": "alice@example.com"  
}
```

Quando crei una fonte di identità con il tuo pool di utenti Amazon Cognito, specifichi il tipo di entità principale con cui Verified Permissions genera nelle richieste di autorizzazione.

IsAuthorizedWithToken Le tue politiche possono quindi testare gli attributi di tale principale come parte della valutazione della richiesta. Lo schema definisce il tipo e gli attributi principali per una fonte di identità, quindi è possibile farvi riferimento nelle politiche Cedar.

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

L'esempio seguente mostra come riflettere gli attributi del token di identità di esempio nello schema di autorizzazioni verificate. Per ulteriori informazioni sulla modifica dello schema, consulta [Modifica degli schemi dell'archivio delle politiche in modalità JSON](#). Se la configurazione dell'origine dell'identità specifica il tipo principale `User`, puoi includere qualcosa di simile al seguente esempio per rendere tali attributi disponibili a Cedar.

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

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

Per un esempio di politica che verrà convalidata rispetto a questo schema, vedi. [Riflette gli attributi del token Amazon Cognito ID](#)

OIDCToken ID

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

Per ulteriori informazioni, consulta [Creazione di archivi di policy per le autorizzazioni verificate](#).

Di seguito è riportato uno schema di esempio per un policy store con un'origine di OIDC identità.

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

```
        "type": "Boolean"
      }
    }
  }
}
```

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

Mappatura dei token di accesso

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

Mappatura dei token di accesso di Amazon Cognito

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

Affermazioni utili nei token di accesso di Amazon Cognito

client_id

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

scope

Gli [ambiti OAuth 2.0](#) che rappresentano i permessi di accesso del portatore del token.

cognito:groups

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

Reclami transitori

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

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

Un token di 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_EXAMPLE",
  "client_id": "1example23456789",
  "origin_jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111",
  "event_id": "bda909cb-3e29-4bb8-83e3-ce6808f49011",
  "token_use": "access",
  "scope": "MyAPI/mydata.write",
  "auth_time": 1688092966,
  "exp": 1688096566,
  "iat": 1688092966,
  "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN22222222",
  "username": "alice"
}
```

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, [consulta Modifica degli schemi dell'archivio delle politiche in modalità JSON](#).

```
{
  "MyApplication": {
```

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

Per un esempio di politica che verrà convalidata rispetto a questo schema, vedi [Riflette gli attributi del token di accesso di Amazon Cognito](#).

Mappatura OIDC dei token di accesso

La maggior parte dei token di accesso di OIDC provider esterni si allinea strettamente ai token di accesso di Amazon Cognito. Un token di OIDC accesso viene mappato su un oggetto di contesto quando viene passato a Verified Permissions. È possibile fare riferimento agli attributi del token di accesso utilizzando `context.token.attribute_name`. Il seguente token di OIDC accesso di esempio include affermazioni di base di esempio.

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

L'esempio seguente mostra come riflettere gli attributi del token di accesso di esempio nello schema di autorizzazioni verificate. Per ulteriori informazioni sulla modifica dello schema, consulta [Modifica degli schemi dell'archivio delle politiche in modalità JSON](#).

```
{
  "MyApplication": {
    "actions": {
      "Read": {
        "appliesTo": {
          "context": {
            "type": "ReusedContext"
          },
          "resourceTypes": [
            "Application"
          ],
          "principalTypes": [
            "User"
          ]
        }
      }
    }
  }
}
```

```
    }
  },
  ...
  ...
  "commonTypes": {
    "ReusedContext": {
      "attributes": {
        "token": {
          "type": "Record",
          "attributes": {
            "scope": {
              "type": "Set",
              "element": {
                "type": "String"
              }
            },
            "client_id": {
              "type": "String"
            }
          }
        }
      }
    },
    "type": "Record"
  }
}
```

Per un esempio di politica che verrà convalidata rispetto a questo schema, vedi [Riflette gli attributi del token di OIDC accesso](#).

Notazione alternativa per le dichiarazioni delimitate da due punti di Amazon Cognito

Al momento del lancio di Verified Permissions, lo schema consigliato per il token Amazon Cognito dichiarava «cognito:groupsmi piace» custom:store e convertiva queste stringhe delimitate da due punti per utilizzare . il carattere come delimitatore gerarchico. Questo formato è chiamato notazione a punti. Ad esempio, un riferimento a cognito:groups became principal.cognito.groups nelle tue politiche. Sebbene sia possibile continuare a utilizzare questo formato, si consiglia di creare lo schema e le politiche utilizzando la [notazione tra parentesi](#).

In questo formato, un riferimento a `cognito:groups` diventa `principal["cognito:groups"]` nelle tue politiche. Gli schemi generati automaticamente per i token ID del pool di utenti dalla console Verified Permissions utilizzano la notazione tra parentesi.

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

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

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

```
}  
}
```

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

Implementazione dell'autorizzazione in Amazon Verified Permissions

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

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

API autorizzazioni

Controlla l'accesso degli utenti ai dati. Le applicazioni fanno spesso parte di un sistema distribuito e importano informazioni dall'esterno API. 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. Le autorizzazioni verificate possono autorizzare la API richiesta che elenca gli account di destinazione idonei al trasferimento e 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 DigitalPetStore di esempio nel vostro ambiente di test.

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

Argomenti

- [Operazioni disponibili per l'autorizzazione API](#)
- [Verifica del tuo modello di autorizzazione](#)
- [Integrazione dei modelli di autorizzazione con le applicazioni](#)

Operazioni disponibili per l'autorizzazione API

Le autorizzazioni verificate API prevedono le seguenti operazioni di autorizzazione.

[IsAuthorized](#)

L'IsAuthorizedAPIoperazione 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'IsAuthorizedWithTokenoperazione genera una richiesta di autorizzazione dai dati utente nei token JSON web di Amazon Cognito (). JWTs Verified Permissions funziona direttamente con Amazon Cognito come fonte di identità nel tuo archivio di politiche. Verified Permissions compila tutti gli attributi relativi all'indirizzo principale della tua richiesta utilizzando le attestazioni contenute nell'ID degli utenti o nei token di accesso. Puoi autorizzare azioni e risorse dagli attributi utente o dall'appartenenza a un gruppo in un pool di utenti di Amazon Cognito.

Non puoi includere informazioni sui tipi principali di gruppi o utenti in una IsAuthorizedWithToken richiesta. È necessario compilare tutti i dati principali in base a JWT quelli forniti.

[BatchIsAuthorized](#)

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

BatchIsAuthorizedWithToken

L'BatchIsAuthorizedWithToken operazione elabora più decisioni di autorizzazione per un singolo principale in un'unica API richiesta. Il principale viene fornito dalla fonte di identità del Policy Store in un ID o token di accesso. Questa operazione raggruppa le richieste in un'unica operazione batch che riduce al minimo l'[utilizzo delle quote](#) e restituisce le decisioni di autorizzazione per ciascuna delle 30 richieste di azioni e risorse. Nelle tue politiche, puoi autorizzare il loro accesso dai loro attributi o la loro appartenenza al gruppo in un pool di utenti di Amazon Cognito.

Ad esempio IsAuthorizedWithToken, non puoi includere informazioni sui tipi principali di gruppi o utenti in una BatchIsAuthorizedWithToken richiesta. È necessario compilare tutti i dati principali in base a JWT quelli forniti.

Verifica del tuo modello di autorizzazione

Per comprendere l'effetto della decisione di autorizzazione di Amazon Verified Permissions quando distribuisce la tua applicazione, puoi valutare le tue politiche man mano che le sviluppi con [Utilizzo del banco di prova Amazon Verified Permissions](#) e con le HTTPS REST API richieste di Autorizzazioni verificate. Il banco di prova è uno strumento AWS Management Console per valutare le richieste e le risposte di autorizzazione nel tuo archivio di politiche.

Le autorizzazioni verificate rappresentano REST API il passaggio successivo dello sviluppo da una comprensione concettuale alla progettazione dell'applicazione. [Le autorizzazioni verificate API accettano le richieste di autorizzazione con IsAuthorized BatchIsAuthorized come AWS API richieste firmate agli endpoint di servizio regionali. IsAuthorizedWithToken](#) Per testare il tuo modello di autorizzazione, puoi generare richieste con qualsiasi API cliente e verificare che le tue politiche restituiscano le decisioni di autorizzazione previste.

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

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 esempio con il nome DigitalPetStore.
2. Seleziona Test bench nel tuo nuovo policy store.

3. Compila la tua richiesta di test bench dal [IsAuthorized](#) riferimento Autorizzazioni API verificate. I seguenti dettagli replicano le condizioni dell'Esempio 4 che fanno riferimento al campione. DigitalPetStore
 - a. Imposta Alice come principale. Affinché il preside agisca, scegli `DigitalPetStore::User` ed entra Alice.
 - b. Imposta il ruolo di Alice come cliente. Scegli Aggiungi un genitore `DigitalPetStore::Role`, scegli e inserisci Cliente.
 - c. Imposta la risorsa come ordine «1234». Per la risorsa su cui agisce il principale, scegli `DigitalPetStore::Order` ed inserisci 1234.
 - d. La `DigitalPetStore::Order` risorsa richiede un `owner` attributo. Imposta Alice come proprietaria dell'ordine. Scegli `DigitalPetStore::User` ed entra Alice
 - e. Alice ha richiesto di visualizzare l'ordine. Per Azione che il preside sta intraprendendo, scegli `DigitalPetStore::Action::"GetOrder"`.
4. Scegli Esegui richiesta di autorizzazione. In un archivio di policy non modificato, questa richiesta genera una ALLOW decisione. Nota la politica di soddisfazione che ha restituito la decisione.
5. Scegli Politiche dalla barra di navigazione a sinistra. Consulta la politica statica con la descrizione Customer Role - Get Order.
6. Tieni presente che Verified Permissions ha consentito la richiesta perché il responsabile ricopriva il ruolo di cliente ed era il proprietario della risorsa.

REST API

1. Apri la console delle autorizzazioni verificate all'indirizzo. <https://console.aws.amazon.com/verifiedpermissions/> Crea un policy store dal Policy Store di esempio con il nome DigitalPetStore.
2. Annota l'ID del Policy Store del tuo nuovo Policy Store.
3. Dal [IsAuthorized](#) API riferimento Autorizzazioni verificate, copia il corpo della richiesta dell'Esempio 4 che fa riferimento all'DigitalPetStore esempio.
4. Apri il API client 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'ownerattributo dell'ordine 1234 inBob.
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 oInactiveUsers. Aggiorna la politica in modo che solo gli utenti attivi possano visualizzare i propri ordini. Aggiorna le entità della richiesta in modo che Alice sia un utente attivo o inattivo.

Integrazione dei modelli di autorizzazione con le applicazioni

Per implementare Amazon Verified Permissions nella tua applicazione, devi definire le politiche e lo schema che desideri che l'app applichi. Una volta implementato e testato il modello di autorizzazione, il passo successivo è iniziare a generare API richieste 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, ad esempio un cookie JWT 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 di autorizzazione API verificate sono integrate. AWS SDKs Per includere le autorizzazioni verificate in un'app, integra AWS SDK la lingua prescelta nel pacchetto dell'app.

Per saperne di più e per effettuare il download AWS SDKs, consulta [Strumenti per Amazon Web Services](#).

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

- [AWS SDK 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 [Semplifica 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 fornitore di politiche di autorizzazione personalizzate perASP. NETApp principali che utilizzano 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 DigitalPetStore di esempio nel vostro ambiente di test.

Il seguente oggetto di contesto dichiara uno di ogni tipo di dati Cedar per un'applicazione basata sul DigitalPetStore policy 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'`DigitalPetStore` app 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 `DigitalPetStore` è 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 costruito 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": "PSEXAMPLEabcdefgh111111"
}
```

Sicurezza nelle autorizzazioni verificate da Amazon

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per maggiori informazioni sui programmi di conformità che si applicano alle autorizzazioni verificate di Amazon, consulta [AWS Services in Scope by Compliance Program AWS](#) .
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando utilizzi le autorizzazioni verificate. I seguenti argomenti mostrano come configurare le autorizzazioni verificate per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse relative alle autorizzazioni verificate.

Argomenti

- [Protezione dei dati in Amazon Verified Permissions](#)
- [Gestione delle identità e degli accessi per Amazon Verified Permissions](#)
- [Convalida della conformità per Amazon Verified Permissions](#)
- [Resilienza nelle autorizzazioni verificate da Amazon](#)

Protezione dei dati in Amazon Verified Permissions

Il modello di [responsabilità AWS condivisa modello](#) si applica alla protezione dei dati in Amazon Verified Permissions. Come descritto in questo modello, AWS è responsabile della protezione

dell'infrastruttura globale che esegue tutto l' Cloud AWS. L'utente è responsabile di mantenere il controllo sui contenuti ospitati su questa infrastruttura. Questo contenuto include le attività di configurazione e gestione della sicurezza per AWS servizi quello che utilizzi. Per ulteriori informazioni sulla privacy dei dati, consulta la sezione [Privacy dei dati FAQ](#). Per informazioni sulla protezione dei dati in Europa, consulta il [Modello di responsabilitàAWS condivisa e GDPR](#) il post sul blog sulla AWS sicurezza.

- Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e di configurare i singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti.
- Ti consigliamo di proteggere i tuoi dati nei seguenti modi:
 - Utilizza l'autenticazione a più fattori (MFA) con ogni account.
 - UsaSSL/TLSper comunicare con AWS le risorse. Abbiamo bisogno di TLS 1.2.
 - Configurazione API e registrazione delle attività degli utenti con AWS CloudTrail.
 - Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno AWS servizi.
 - Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
 - Se hai bisogno di FIPS 140-2 moduli crittografici convalidati per accedere AWS tramite un'interfaccia a riga di comando o unAPI, usa un endpoint. FIPS Per ulteriori informazioni sugli FIPS endpoint disponibili, vedere [Federal Information Processing Standard \(\) 140-2](#). FIPS
- Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con autorizzazioni verificate o altro AWS servizi utilizzando la console,API, AWS CLI o. AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Se fornisci un URL a un server esterno, ti consigliamo vivamente di non includere le informazioni sulle credenziali URL per convalidare la tua richiesta a quel server.
- I nomi delle azioni non devono includere informazioni sensibili.
- Inoltre, ti consigliamo vivamente di utilizzare sempre identificatori unici, non modificabili e non riutilizzabili per le tue entità (risorse e principali). In un ambiente di test, puoi scegliere di utilizzare identificatori di entità semplici, come jane o bob per il nome di un'entità di tipo. User Tuttavia, in un sistema di produzione, è fondamentale per motivi di sicurezza utilizzare valori univoci che non possano essere riutilizzati. Ti consigliamo di utilizzare valori come identificatori univoci universali

(). UUIDs Ad esempio, si consideri l'utente `jane` che lascia l'azienda. Successivamente, consenti a qualcun altro di usare il nome `jane`. Quel nuovo utente ottiene automaticamente l'accesso a tutto ciò che è concesso dalle politiche a cui fanno ancora riferimento `User : : "jane"`. `Verified Permissions` e `Cedar` non riescono a distinguere tra il nuovo utente e l'utente precedente.

Questa guida si applica sia agli identificatori principali che a quelli di risorse. Utilizza sempre identificatori che siano univoci garantiti e mai riutilizzati per assicurarti di non concedere l'accesso involontariamente a causa della presenza di un vecchio identificatore in una politica.

- Assicurati che le stringhe che fornisci per definire `Long` e `Decimal` i valori rientrino nell'intervallo valido di ogni tipo. Inoltre, assicuratevi che l'utilizzo di qualsiasi operatore aritmetico non produca un valore al di fuori dell'intervallo valido. Se l'intervallo viene superato, l'operazione genera un'eccezione di overflow. Una politica che genera un errore viene ignorata, il che significa che una politica di autorizzazione potrebbe inaspettatamente non consentire l'accesso o una politica di divieto potrebbe inaspettatamente non riuscire a bloccare l'accesso.

Crittografia dei dati

Amazon `Verified Permissions` crittografa automaticamente tutti i dati dei clienti, ad esempio le politiche, con una chiave gestita dal cliente. Chiave gestita da AWS, quindi 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 AWS servizio che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. IAM gli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse di Autorizzazioni verificate. IAM è un software AWS servizio che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)

- [Come funziona Amazon Verified Permissions con IAM](#)
- [IAM politiche per le autorizzazioni verificate](#)
- [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. È quindi necessario inviare richieste all' IAM amministratore per modificare le autorizzazioni degli utenti del servizio. Consulta le informazioni contenute in questa pagina per comprendere i concetti di base di IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM le autorizzazioni verificate, consulta [Come funziona Amazon Verified Permissions con IAM](#).

IAM amministratore: 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 il modo in cui accedi AWS utilizzando le tue credenziali di identità. È necessario autenticarsi (accedere a AWS) come Utente root dell'account AWS, come IAM utente o assumendo un ruolo. IAM

È possibile accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Quando accedi come identità federata, l'amministratore aveva precedentemente configurato la federazione delle identità utilizzando i ruoli. IAM Quando si accede AWS utilizzando la federazione, si assume indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando () per firmare crittograficamente le tue richieste utilizzando le tue credenziali. CLI Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [Firmare AWS API le richieste](#) nella Guida per l'IAM utente.

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

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte AWS servizi le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso 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 utilizzale 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'IAM utente.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere AWS servizi utilizzando credenziali temporanee.

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

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per informazioni su IAM Identity Center, vedi [Cos'è IAM Identity Center?](#) nella Guida AWS IAM Identity Center per l'utente.

IAM users and groups

Un [IAM utente](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Laddove possibile, consigliamo di fare affidamento su credenziali temporanee anziché creare IAM utenti con credenziali a lungo termine come password e chiavi di accesso. Tuttavia, se hai casi d'uso specifici che richiedono credenziali a lungo termine con IAM gli utenti, ti consigliamo 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'utente.IAM

Un [IAM gruppo](#) è un'identità che specifica un insieme 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 ulteriori informazioni, consulta [Quando creare un IAM utente \(anziché un ruolo\)](#) nella Guida per l'IAM utente.

IAM ruoli

Un [IAM ruolo](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un IAM utente, ma non è associato a una persona specifica. È possibile assumere temporaneamente un IAM ruolo in AWS Management Console [cambiando ruolo](#). È possibile assumere un ruolo chiamando un' AWS API/operazione AWS CLI or o utilizzando un'operazione

personalizzata URL. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere [Utilizzo IAM dei ruoli](#) nella Guida per l'IAM utente.

IAM i ruoli con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per informazioni sui ruoli per la federazione, vedere [Creazione di un ruolo per un provider di identità di terze parti](#) nella Guida per l'IAM utente. Se utilizzi IAM Identity Center, configuri un set di autorizzazioni. Per controllare a cosa possono accedere le identità dopo l'autenticazione, IAM Identity Center correla il set di autorizzazioni a un ruolo in IAM. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .
- **Autorizzazioni IAM utente temporanee:** un IAM utente o un ruolo può assumere il IAM ruolo di assumere temporaneamente autorizzazioni diverse per un'attività specifica.
- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (principale attendibile) di un account diverso di accedere alle risorse nel tuo account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni AWS servizi, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta [In che modo IAM i ruoli differiscono dalle politiche basate sulle risorse](#) nella Guida per l'utente IAM.
- **Applicazioni in esecuzione Amazon EC2:** è possibile utilizzare un IAM ruolo per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza e che effettuano o effettuano richieste. EC2 AWS CLI AWS API È preferibile archiviare le chiavi di accesso all'interno dell'EC2istanza. Per assegnare un AWS ruolo a un'EC2istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull'EC2istanza di ottenere 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'IAM utente.

Per sapere se utilizzare IAM ruoli o IAM utenti, consulta [Quando creare un IAM ruolo \(anziché un utente\)](#) nella Guida per l'IAM utente.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come JSON documenti. Per ulteriori informazioni sulla struttura e il contenuto dei documenti relativi alle JSON politiche, vedere [Panoramica delle JSON politiche](#) nella Guida per l'IAM utente.

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. 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 l'autorizzazione a 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.

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

Policy basate su identità

I criteri basati sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. [Per informazioni su come creare una politica basata sull'identità, consulta Creazione di politiche nella Guida per l'utente. IAM IAM](#)

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli all'interno del tuo. Account AWS Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una politica gestita o una politica in linea, consulta [Scelta tra politiche gestite e politiche in linea nella Guida](#) per l'IAM utente.

Policy basate su risorse

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di policy basate sulle risorse sono le policy di IAM role trust e le policy di Amazon S3 bucket. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le 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 AWS servizi

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

Elenchi di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la [panoramica di Access control list \(ACL\)](#) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e 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](#)
- **Politiche di controllo del servizio (SCPs):** SCPs sono JSON politiche che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account

AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna Utente root dell'account AWS. Per ulteriori informazioni su Organizations and SCPs, consulta [le politiche di controllo dei servizi](#) nella Guida AWS Organizations per l'utente.

- **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 sapere 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'IAM utente.

Come funziona Amazon Verified Permissions con IAM

Prima di utilizzare IAM per gestire l'accesso alle autorizzazioni verificate, scopri quali IAM funzionalità sono disponibili per l'uso con le autorizzazioni verificate.

IAM funzionalità che puoi utilizzare con Amazon Verified Permissions

IAM funzionalità	Supporto per le autorizzazioni verificate
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	No

IAM funzionalità	Supporto per le autorizzazioni verificate
ACLs	No
ABAC(tag nelle politiche)	No
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	No
Ruoli collegati al servizio	No

Per avere una panoramica generale del funzionamento delle Autorizzazioni verificate e degli altri AWS servizi con la maggior parte delle IAM funzionalità, consulta [AWS i servizi che funzionano con IAM nella Guida](#) per l'IAM utente.

Politiche basate sull'identità per le autorizzazioni verificate

Supporta le policy basate su identità	Sì
---------------------------------------	----

Le politiche basate sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. [Per informazioni su come creare una politica basata sull'identità, consulta Creazione di politiche nella Guida per l'utente. IAM IAM](#)

Con le politiche IAM basate sull'identità, puoi specificare azioni e risorse consentite o negate, nonché le condizioni in base alle quali le azioni sono consentite o negate. 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 ulteriori informazioni su tutti gli elementi che è possibile utilizzare in una JSON politica, vedere il [riferimento agli elementi IAM JSON della politica](#) nella Guida per l'IAM utente.

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 politiche basate sulle risorse sono documenti di policy allegati a JSON una risorsa. Esempi di policy basate sulle risorse sono le policy di IAM role trust e le policy di Amazon S3 bucket. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le 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 AWS servizi

Per abilitare l'accesso tra più account, puoi specificare un intero account o IAM entità in un altro account come principale in una politica basata sulle risorse. 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 diversi Account AWS, un IAM amministratore dell'account fidato 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 [Cross Account Resource Access IAM nella Guida IAM per l'utente](#).

Azioni politiche per le autorizzazioni verificate

Supporta le operazioni di policy

Sì

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'Actionelemento di una JSON policy descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a una policy. Le azioni politiche in genere hanno lo stesso nome dell' AWS APIoperazione associata. Esistono alcune eccezioni, come le azioni basate solo sulle autorizzazioni che non hanno un'operazione corrispondente. API 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 azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola Get, includi la seguente azione:

```
"Action": "verifiedpermissions:Get*"
```

Per visualizzare esempi di politiche basate sull'identità delle autorizzazioni verificate, consulta [Esempi di policy basate sull'identità per Amazon Verified Permissions](#)

Risorse politiche per le autorizzazioni verificate

Supporta le risorse di policy	Sì
-------------------------------	----

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento Resource JSON policy specifica l'oggetto o gli oggetti a cui si applica l'azione. Le istruzioni devono includere un elemento Resource o un elemento NotResource. Come best practice, specifica una risorsa utilizzando il relativo [Amazon Resource Name \(ARN\)](#). 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 relativi ARNs, consulta [Tipi di risorse definiti da Amazon Verified Permissions](#) nel Service Authorization Reference. Per sapere con quali azioni puoi specificare il tipo ARN di ciascuna risorsa, consulta [Azioni definite da Amazon Verified Permissions](#).

Chiavi relative alle condizioni delle policy per le autorizzazioni verificate

Supporta le chiavi di condizione delle policy specifiche del servizio	No
---	----

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. 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 si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica OR. 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, è possibile concedere a un IAM utente l'autorizzazione ad accedere a una risorsa solo se è contrassegnata con il suo nome IAM utente. Per ulteriori informazioni, consulta [gli elementi IAM della politica: variabili e tag](#) nella Guida IAM per l'utente.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'IAM utente.

ACLsin Autorizzazioni verificate

Supporti ACLs	No
---------------	----

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

ABACcon autorizzazioni verificate

Supporti ABAC (tag nelle politiche)	No
-------------------------------------	----

Il controllo degli accessi basato sugli attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. È possibile allegare tag a IAM entità (utenti o ruoli) e a molte AWS risorse. L'etichettatura di entità e risorse è il primo passo di ABAC. Quindi si progettano ABAC politiche per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa a cui sta tentando di accedere.

ABAC è utile in ambienti in rapida crescita e aiuta in situazioni in cui la gestione delle politiche diventa complicata.

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, vedere [Cos'è? ABAC](#) nella Guida IAM per l'utente. Per visualizzare un tutorial con i passaggi per la configurazione ABAC, consulta [Utilizzare il controllo di accesso basato sugli attributi \(ABAC\)](#) nella Guida per l'IAM utente.

Utilizzo di credenziali temporanee con autorizzazioni verificate

Supporta le credenziali temporanee	Sì
------------------------------------	----

Alcune AWS servizi non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che AWS servizi funzionano con credenziali temporanee, consulta la sezione [AWS servizi relativa alla funzionalità IAM nella Guida](#) per l'IAM utente.

Si utilizzano credenziali temporanee se si accede AWS Management Console utilizzando qualsiasi metodo tranne il nome utente e la password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-on (SSO) della vostra azienda, tale processo crea automaticamente 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, consulta [Passare a un ruolo \(console\)](#) nella Guida per l'IAM utente.

È possibile creare manualmente credenziali temporanee utilizzando AWS CLI o AWS API. È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, vedere [Credenziali di sicurezza temporanee](#) in IAM.

Autorizzazioni principali multiservizio per le autorizzazioni verificate

Supporta le autorizzazioni delle entità principali Sì

Quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un AWS servizio, in combinazione con la richiesta AWS servizio per effettuare richieste ai servizi downstream. FAS le richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri AWS servizi o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta [Forward access sessions](#).

Ruoli di servizio per le autorizzazioni verificate

Supporta i ruoli di servizio No

Un ruolo di servizio è un [IAM ruolo](#) che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'interno IAM.

Per ulteriori informazioni, consulta la pagina relativa alla [creazione di un ruolo per delegare le autorizzazioni a un AWS servizio](#) 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 al servizio è un tipo di ruolo di servizio collegato a un. AWS servizio Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.

[Per informazioni dettagliate sulla creazione o la gestione di ruoli collegati ai servizi, consulta AWS Servizi compatibili con. IAM](#) Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

IAM politiche per le autorizzazioni verificate

Verified Permissions gestisce le autorizzazioni degli utenti all'interno dell'applicazione. Affinché l'applicazione possa richiamare le autorizzazioni verificate APIs o 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 politiche basate sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. [Per informazioni su come creare una politica basata sull'identità, consulta Creazione di politiche nella Guida per l'utente. IAM](#) IAM

Con le politiche IAM basate sull'identità, puoi specificare azioni e risorse consentite o negate, nonché le condizioni in base alle quali le azioni sono consentite o negate (elencate di seguito). Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per maggiori informazioni su tutti gli elementi che è possibile utilizzare in una JSON policy, consulta il [riferimento agli elementi IAM JSON della policy](#) nella Guida per l' IAM utente.

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 in Account 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": "*"
    }
  ]
}
```

```
}
```

Esempi di policy basate sull'identità per Amazon Verified Permissions

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse con autorizzazioni verificate. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS API. Un IAM amministratore deve creare IAM politiche che concedano a utenti e ruoli l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno. L'amministratore deve quindi collegare queste policy agli utenti che ne hanno bisogno.

Per informazioni su come creare una politica IAM basata sull'identità utilizzando questi documenti di esempio JSON, consulta [Creazione di IAM politiche](#) nella Guida per l'IAM utente.

Per dettagli sulle azioni e sui tipi di risorse definiti da Verified Permissions, incluso il formato di ARNs per ogni tipo di risorsa, consulta [Azioni, risorse e chiavi di condizione per Amazon Verified Permissions](#) nel Service Authorization Reference.

Argomenti

- [Best practice per le policy](#)
- [Utilizzo della console delle autorizzazioni verificate](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse Verified Permissions nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [le politiche AWS gestite o le politiche AWS gestite per le funzioni lavorative](#) nella Guida per l'IAM utente.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le IAM politiche, concedi solo le autorizzazioni necessarie per eseguire un'attività. 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. IAMIAM

- Utilizza le condizioni nelle IAM politiche per limitare ulteriormente l'accesso: puoi aggiungere una condizione alle tue politiche per limitare l'accesso ad azioni e risorse. Ad esempio, puoi scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. È inoltre possibile utilizzare condizioni per concedere l'accesso alle azioni di servizio se vengono utilizzate tramite uno specifico AWS servizio, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta [Elementi IAM JSON della politica: Condizione](#) nella Guida IAM per l'utente.
- 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 () e alle best practice. JSON IAM IAMAccess Analyzer fornisce più di 100 controlli delle politiche e consigli pratici per aiutarti a creare policy sicure e funzionali. Per ulteriori informazioni, vedere [Convalida delle policy di IAM Access Analyzer nella Guida per l'utente.IAM](#)
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede l'utilizzo di IAM utenti o di un utente root Account AWS, attiva questa opzione MFA per una maggiore sicurezza. Per richiedere MFA quando vengono richiamate API le operazioni, aggiungi MFA delle condizioni alle tue politiche. Per ulteriori informazioni, vedere [Configurazione dell'APIaccesso MFA protetto nella Guida](#) per l'IAM utente.

Per ulteriori informazioni sulle procedure consigliate in IAM, consulta la sezione [Procedure consigliate in materia di sicurezza IAM nella](#) Guida per l'IAM utente.

Utilizzo della console delle autorizzazioni verificate

Per accedere alla console Amazon Verified Permissions, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse di Autorizzazioni verificate presenti nel tuo Account AWS. Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso la o la AWS CLI . AWS API Consenti invece l'accesso solo alle azioni che corrispondono all'APIoperazione 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 come è possibile creare una politica che consenta IAM agli utenti di visualizzare le politiche in linea e gestite allegate alla loro identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando o a livello di codice. AWS CLI
AWS API

```
{
  "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 relative alle autorizzazioni verificate](#)

Non sono autorizzato a eseguire un'azione in Autorizzazioni verificate

Se ricevi un errore che indica che non disponi dell'autorizzazione per eseguire un'operazione, le tue policy devono essere aggiornate in modo che ti sei consentito eseguire tale operazione.

L'errore di esempio seguente si verifica quando l'utente `mateojacksonIAMutente` tenta di utilizzare la console per visualizzare i dettagli su una `my-example-widget` risorsa fittizia ma non dispone delle autorizzazioni fittizie. `verifiedpermissions:GetWidget`

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

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa `my-example-widget` utilizzando l'azione `verifiedpermissions:GetWidget`.

Se hai bisogno di assistenza, 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 AWS servizi consentono di trasferire un ruolo esistente a quel servizio anziché creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un IAM utente 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`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse relative alle autorizzazioni verificate

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Verified Permissions supporta queste funzionalità, consulta [Come funziona Amazon Verified Permissions con IAM](#)
- Per sapere come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta [Fornire l'accesso a un IAM utente di un altro Account AWS utente di tua proprietà](#) nella Guida per l'IAM utente.

- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a persone Account AWS di proprietà di terzi](#) nella Guida per l'IAM utente.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso agli utenti autenticati esternamente \(federazione delle identità\)](#) nella Guida per l'IAM utente.
- Per conoscere la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle [risorse tra account nella Guida per l'utente](#).
IAMIAM

Convalida della conformità per Amazon Verified Permissions

Per sapere se un AWS servizio programma rientra nell'ambito di specifici programmi di conformità, consulta AWS servizi la sezione [Scope by Compliance Program AWS servizi](#) e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di [AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) .

La vostra responsabilità di conformità durante l'utilizzo AWS servizi è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide introduttive su sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla AWS sicurezza e la conformità.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#): questo white paper descrive in che modo le aziende possono utilizzare per creare applicazioni idonee. AWS HIPAA

Note

Non tutte sono idonee. AWS servizi HIPAA Per ulteriori informazioni, consulta la [Guida ai servizi HIPAA idonei](#).

- [AWS Risorse per AWS](#) per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe riguardare il tuo settore e la tua località.

- [AWS Guide alla conformità dei clienti](#): comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione AWS servizi 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
- [Evaluating Resources with Rules](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub](#)— Ciò AWS servizio fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e 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](#).
- [Amazon GuardDuty](#): AWS servizio rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, ad esempio PCI DSS soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- [AWS Audit Manager](#)— Ciò AWS servizio consente di verificare continuamente AWS l'utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Resilienza nelle autorizzazioni verificate da Amazon

L'infrastruttura AWS globale è costruita attorno a Regioni AWS zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Quando si crea un archivio di criteri per le autorizzazioni verificate, questo viene creato all'interno di un singolo Regione AWS utente e viene replicato automaticamente nei data center che costituiscono le zone di disponibilità di quella regione. Al momento, Verified Permissions non supporta alcuna replica tra regioni.

[Per ulteriori informazioni sulle Regioni AWS zone di disponibilità, consulta AWS Global Infrastructure.](#)

Monitoraggio delle chiamate Amazon Verified Permissions API

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon Verified Permissions e delle altre AWS soluzioni. AWS fornisce i seguenti strumenti per monitorare le autorizzazioni verificate, segnalare quando qualcosa non va e intraprendere azioni automatiche se necessario:

- AWS CloudTrail acquisisce le API chiamate e gli eventi correlati effettuati da o per conto del tuo AWS account e invia i file di registro a un bucket Amazon S3 da te specificato. Puoi identificare quali utenti e account hanno chiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la [Guida per l'utente AWS CloudTrail](#).

Per ulteriori informazioni sul monitoraggio delle autorizzazioni verificate con CloudTrail, consulta [Registrazione delle chiamate Amazon Verified Permissions tramite API AWS CloudTrail](#).

Registrazione delle chiamate Amazon Verified Permissions tramite API AWS CloudTrail

Amazon Verified Permissions è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in Autorizzazioni verificate. CloudTrail registra tutte le API chiamate per le autorizzazioni verificate come eventi. Le chiamate acquisite includono chiamate dalla console Autorizzazioni verificate e chiamate in codice alle operazioni Autorizzazioni verificate. API 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 CloudTrail utente](#).

Informazioni sulle autorizzazioni verificate in CloudTrail

CloudTrail è abilitato sul tuo Account AWS quando crei l'account. Quando si verifica un'attività in Autorizzazioni verificate, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti in Account AWS. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con la cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi del tuo Account AWS, compresi gli eventi per le autorizzazioni verificate, crea un percorso. Un trail consente di CloudTrail inviare file di 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 trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle SNS notifiche Amazon per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Tutte le azioni relative alle autorizzazioni verificate vengono registrate CloudTrail e documentate nella [Amazon Verified API Permissions](#) Reference Guide. Ad esempio, le chiamate alle ListPolicyStores azioni CreateIdentitySourceDeletePolicy, e generano voci nei file di registro. CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, vedete l'[CloudTrail userIdentity elemento](#).

Per impostazione predefinita, gli eventi relativi ai dati, come i [IsAuthorized](#), non [IsAuthorizedWithToken](#) vengono 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 stack ordinata delle API chiamate 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": "PSEXAMPLEabcdefg1111111"
  },
  "responseElements": null,
  "additionalEventData": {
    "decision": "ALLOW"
  },
  "requestID": "346c4b6a-d12f-46b6-bc06-6c857bd3b28e",
```

```

    "eventID": "8a4fed32-9605-45dd-a09a-5ebbf0715bbc",
    "readOnly": true,
    "resources": [
      {
        "accountId": "123456789012",
        "type": "AWS::VerifiedPermissions::PolicyStore",
        "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
  }

```

BatchIsAuthorized

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-11-20T23:02:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "BatchIsAuthorized",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
  "requestParameters": {
    "requests": [
      {
        "principal": {
          "entityType": "PhotoFlash::User",
          "entityId": "alice"
        },
        "action": {
          "actionType": "PhotoFlash::Action",

```

```
        "actionId": "ViewPhoto"
      },
      "resource": {
        "entityType": "PhotoFlash::Photo",
        "entityId": "VacationPhoto94.jpg"
      }
    },
    {
      "principal": {
        "entityType": "PhotoFlash::User",
        "entityId": "annalisa"
      },
      "action": {
        "actionType": "PhotoFlash::Action",
        "actionId": "DeletePhoto"
      },
      "resource": {
        "entityType": "PhotoFlash::Photo",
        "entityId": "VacationPhoto94.jpg"
      }
    }
  ],
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
},
"responseElements": null,
"additionalEventData": {
  "results": [
    {
      "request": {
        "principal": {
          "entityType": "PhotoFlash::User",
          "entityId": "alice"
        },
        "action": {
          "actionType": "PhotoFlash::Action",
          "actionId": "ViewPhoto"
        },
        "resource": {
          "entityType": "PhotoFlash::Photo",
          "entityId": "VacationPhoto94.jpg"
        }
      },
      "decision": "ALLOW"
    }
  ]
},
```

```

    {
      "request": {
        "principal": {
          "entityType": "PhotoFlash::User",
          "entityId": "annalisa"
        },
        "action": {
          "actionType": "PhotoFlash::Action",
          "actionId": "DeletePhoto"
        },
        "resource": {
          "entityType": "PhotoFlash::Photo",
          "entityId": "VacationPhoto94.jpg"
        }
      },
      "decision": "DENY"
    }
  ],
  "requestID": "a8a5caf3-78bd-4139-924c-7101a8339c3b",
  "eventID": "7d81232f-f3d1-4102-b9c9-15157c70487b",
  "readOnly": true,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "123456789012",
  "eventCategory": "Data"
}

```

CreatePolicyStore

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",

```

```

    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "validationSettings": {
      "mode": "OFF"
    }
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefgh111111",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/PSEXAMPLEabcdefgh111111",
    "createdDate": "2023-05-22T07:43:33.962794Z",
    "lastUpdatedDate": "2023-05-22T07:43:33.962794Z"
  },
  "requestID": "1dd9360e-e2dc-4554-ab65-b46d2cf45c29",
  "eventID": "b6edae-3584-4b4e-a48e-311de46d7532",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}

```

ListPolicyStores

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
}

```

```

"eventTime": "2023-05-22T07:43:33Z",
"eventSource": "verifiedpermissions.amazonaws.com",
"eventName": "ListPolicyStores",
"awsRegion": "us-west-2",
"sourceIPAddress": "203.0.113.0",
"userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
"requestParameters": {
  "maxResults": 10
},
"responseElements": null,
"requestID": "5ef238db-9f87-4f37-ab7b-6cf0ba5df891",
"eventID": "b0430fb0-12c3-4cca-8d05-84c37f99c51f",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

DeletePolicyStore

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeletePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "1368e8f9-130d-45a5-b96d-99097ca3077f",
  "eventID": "ac482022-b2f6-4069-879a-dd509123d8d7",
  "readOnly": false,
}

```

```

"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

PutSchema

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-16T12:58:57Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "PutSchema",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "lastUpdatedDate": "2023-05-16T12:58:57.513442Z",
    "namespaces": "[some_namespace]",
    "createdDate": "2023-05-16T12:58:57.513442Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
  },
  "requestID": "631fbfa1-a959-4988-b9f8-f1a43ff5df0d",
  "eventID": "7cd0c677-733f-4602-bc03-248bae581fe5",
  "readOnly": false,

```



```

"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

GetSchema

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::222222222222:role/ExampleRole",
    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-25T01:12:07Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetSchema",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "a1f4d4cd-6156-480a-a9b8-e85a71dcc7c2",
  "eventID": "0b3b8e3d-155c-46f3-a303-7e9e8b5f606b",
  "readOnly": true,
  "resources": [
    {
      "accountId": "222222222222",
      "type": "AWS::VerifiedPermissions::PolicyStore",

```

```

    "ARN": "arn:aws:verifiedpermissions::222222222222:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "222222222222",
"eventCategory": "Management"
}

```

CreatePolicyTemplate

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-16T13:00:24Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyTemplate",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "lastUpdatedDate": "2023-05-16T13:00:23.444404Z",
    "createdDate": "2023-05-16T13:00:23.444404Z",
    "policyTemplateId": "PTEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
  },
  "requestID": "73953bda-af5e-4854-afe2-7660b492a6d0",
  "eventID": "7425de77-ed84-4f91-a4b9-b669181cc57b",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",

```

```

    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

DeletePolicyTemplate

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::222222222222:role/ExampleRole",
    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-25T01:11:48Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeletePolicyTemplate",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "5ff0f22e-6bbd-4b85-a400-4fb74aa05dc6",
  "eventID": "c0e0c689-369e-4e95-a9cd-8de113d47ffa",
  "readOnly": false,
  "resources": [
    {
      "accountId": "222222222222",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::222222222222:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
}

```

```
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "222222222222",
"eventCategory": "Management"
}
```

CreatePolicy

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:42:30Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
      "entityType": "PhotoApp::Role",
      "entityId": "PhotoJudge"
    },
    "resource": {
      "entityType": "PhotoApp::Application",
      "entityId": "PhotoApp"
    },
    "lastUpdatedDate": "2023-05-22T07:42:30.70852Z",
    "createdDate": "2023-05-22T07:42:30.70852Z"
  },
  "requestID": "93ffa151-3841-4960-9af6-30a7f817ef93",
```

```

"eventID": "30ab405f-3dff-43ff-8af9-f513829e8bde",
"readOnly": false,
"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

GetPolicy

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:29Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetPolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "23022a9e-2f5c-4dac-b653-59e6987f2fac",
  "eventID": "9b4d5037-bafa-4d57-b197-f46af83fc684",
  "readOnly": true,
  "resources": [
    {

```

```

    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

CreateIdentitySource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-19T01:27:44Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreateIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN11111111",
    "configuration": {
      "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:000011112222:us-east-1:userpool/us-
east-1_aaaaaaaaaa"
      }
    }
  },
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "principalEntityType": "User"
},
"responseElements": {
  "createdDate": "2023-07-14T15:05:01.599534Z",
  "identitySourceId": "ISEXAMPLEabcdefg111111",

```

```

    "lastUpdatedDate": "2023-07-14T15:05:01.599534Z",
    "policyStoreId": "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/
PSEXAMPLEEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "333333333333",
"eventCategory": "Management"
}

```

DeleteIdentitySource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeleteIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEEabcdefg111111",
    "policyStoreId": "PSEXAMPLEEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "d554d964-0957-4834-a421-c417bd293086",
  "eventID": "fe4d867c-88ee-4e5d-8d30-2fbc208c9260",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEEabcdefg111111"
    }
  ]
}

```

```
],  
  "eventType": "AwsApiCall",  
  "managementEvent": true,  
  "recipientAccountId": "333333333333",  
  "eventCategory": "Management"  
}
```

Creazione di risorse Amazon Verified Permissions con AWS CloudFormation

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

Quando lo utilizzi AWS CloudFormation, puoi riutilizzare il modello per configurare le risorse relative alle autorizzazioni verificate in modo coerente e ripetuto. Descrivi le tue risorse una sola volta, quindi fornisci le stesse risorse più e più volte in più aree geografiche Account AWS .

Important

Amazon Cognito Identity non è affatto disponibile come Regioni AWS Amazon Verified Permissions. Se ricevi un errore relativo ad Amazon Cognito Identity, ad esempio, ti consigliamo di creare il pool di utenti e il client Amazon Cognito nel luogo geograficamente più vicino in Regione AWS cui è disponibile Amazon Cognito Identity. AWS CloudFormation Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient Usa questo pool di utenti appena creato per creare la fonte di identità Verified Permissions.

Autorizzazioni e modelli verificati AWS CloudFormation

[Per fornire e configurare le risorse per le autorizzazioni verificate e i servizi correlati, è necessario conoscere AWS CloudFormation i modelli.](#) I modelli sono file di testo formattati in JSON o YAML. Questi modelli descrivono le risorse che desideri inserire nei tuoi AWS CloudFormation stack. Se non conosci JSON oYAML, puoi usare AWS CloudFormation Designer per iniziare a usare i AWS CloudFormation modelli. Per ulteriori informazioni, consulta [Cos'è AWS CloudFormation Designer?](#) nella Guida AWS CloudFormation per l'utente.

Verified Permissions supporta la creazione di fonti di identità, policy, archivi di policy e modelli di policy in AWS CloudFormation. Per ulteriori informazioni, inclusi esempi JSON e YAML modelli per le risorse Verified Permissions, consulta il [riferimento al tipo di risorsa Amazon Verified Permissions nella Guida](#) per l'AWS CloudFormation utente.

AWS CDKcostrutti

AWS Cloud Development Kit (AWS CDK) è un framework di sviluppo software open source per definire l'infrastruttura cloud in codice e fornirla tramite AWS CloudFormation I costrutti, o componenti cloud riutilizzabili, possono essere utilizzati per creare modelli. AWS CloudFormation Questi modelli possono quindi essere utilizzati per implementare l'infrastruttura cloud.

Per ulteriori informazioni e per il download AWS CDKs, consulta [AWS Cloud Development Kit](#).

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

- [Autorizzazioni verificate Amazon L2 Construct CDK](#)

Scopri di più su AWS CloudFormation

Per ulteriori informazioni AWS CloudFormation, consulta le seguenti risorse:

- [AWS CloudFormation](#)
- [AWS CloudFormation Guida per l'utente](#)
- [AWS CloudFormation API Riferimento](#)
- [AWS CloudFormation Guida per l'utente dell'interfaccia a riga di comando](#)

Accedi alle autorizzazioni verificate di Amazon utilizzando AWS PrivateLink

Puoi utilizzarlo AWS PrivateLink per creare una connessione privata tra le tue autorizzazioni VPC e Amazon Verified Permissions. Puoi accedere alle autorizzazioni verificate come se fossero nelle tue VPC, senza l'uso di un gateway, NAT dispositivo, VPN connessione o AWS Direct Connect connessione Internet. Le istanze del tuo paese VPC non hanno bisogno di indirizzi IP pubblici per accedere alle autorizzazioni verificate.

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creato un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Si tratta di interfacce di rete gestite dai richiedenti che fungono da punto di ingresso per il traffico destinato alle autorizzazioni verificate.

Per ulteriori informazioni, consulta la sezione [Accesso a AWS servizi tramite AWS PrivateLink](#) nella Guida di AWS PrivateLink .

Considerazioni relative alle autorizzazioni verificate

Prima di configurare un endpoint di interfaccia per le autorizzazioni verificate, consulta le [considerazioni](#) nella Guida AWS PrivateLink

Verified Permissions supporta l'esecuzione di chiamate a tutte le sue API azioni tramite l'endpoint dell'interfaccia.

VPCle politiche degli endpoint non sono supportate per le autorizzazioni verificate. Per impostazione predefinita, l'accesso completo alle autorizzazioni verificate è consentito tramite l'endpoint dell'interfaccia. In alternativa, è possibile associare un gruppo di sicurezza alle interfacce di rete degli endpoint per controllare il traffico verso le autorizzazioni verificate attraverso l'endpoint dell'interfaccia.

Crea un endpoint di interfaccia per le autorizzazioni verificate

Puoi creare un endpoint di interfaccia per le autorizzazioni verificate utilizzando la VPC console Amazon o AWS Command Line Interface (AWS CLI). Per ulteriori informazioni, consulta la sezione [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di AWS PrivateLink .

Crea un endpoint di interfaccia per le autorizzazioni verificate utilizzando il seguente nome di servizio:

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

Se abiliti private DNS per l'endpoint dell'interfaccia, puoi effettuare API richieste alle autorizzazioni verificate utilizzando il nome regionale predefinito. DNS 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](#)
- [Quote per le gerarchie](#)
- [Quote per operazioni al secondo](#)

Quote per le risorse

Nome	Predefinita	Adatta e	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	Adatta	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 [IsAuthorized](#) e [IsAuthorizedWithToken](#).

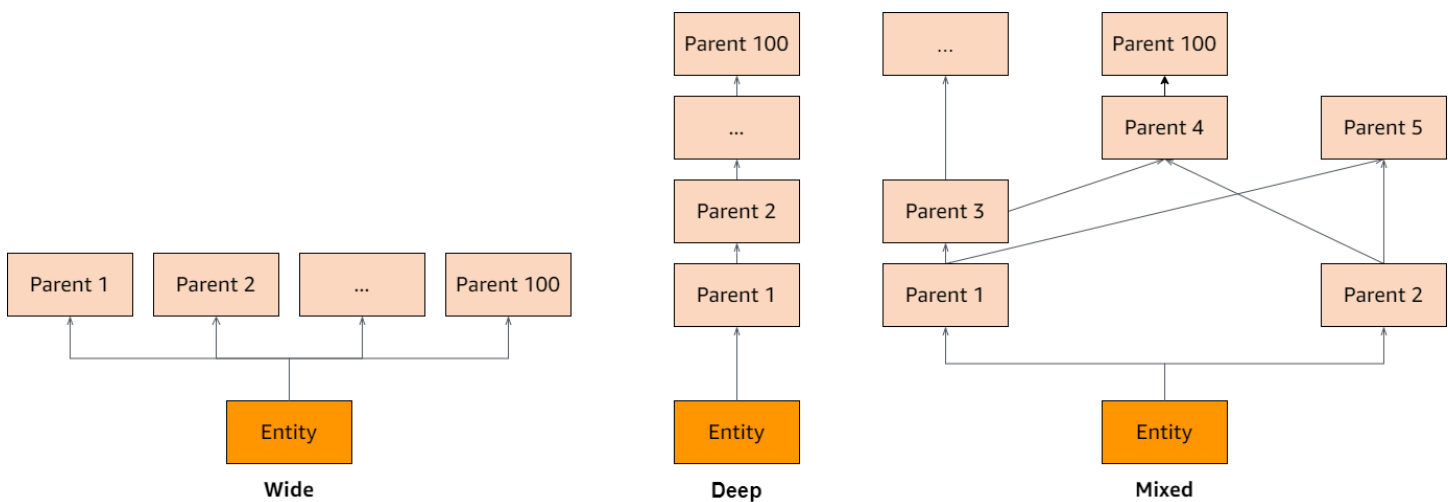
² La dimensione totale di tutte le politiche relative a una singola risorsa non può superare i 200.000 byte. Inoltre, la dimensione totale di tutte le politiche che specificano «Tutte le risorse» non può superare i 200.000 byte. Per le policy collegate al modello, la dimensione del modello di policy 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	Adatta	Descrizione
Genitori transitivi per preside	100	No	Il numero massimo di genitori transitivi per ogni principale.

Nome	Predefinita	Adattate	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 dell'applicazione 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. Puoi visualizzare le tue RPS quote attuali in [Service Quotas](#). Per evitare che le applicazioni superino la quota prevista 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 nei carichi di lavoro. API

Nome	Predefinita	Adatta	Descrizione
BatchIsAuthorized richieste al secondo per regione per account	Ogni regione supportata: 30	Sì	Il numero massimo di BatchIsAuthorized richieste al secondo.
BatchIsAuthorizedWithToken richieste al secondo per regione per account	Ogni regione supportata: 30	Sì	Il numero massimo di BatchIsAuthorizedWithToken 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.

Nome	Predefinita	Adatta e	Descrizione
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.
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.

Nome	Predefinita	Adattate	Descrizione
UpdatePolicy richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di UpdatePolicy richieste al secondo.
UpdatePolicyStore richieste al secondo per regione per account	Ogni regione supportata: 10	No	Il numero massimo di UpdatePolicyStore richieste al secondo.
UpdatePolicyTemplate richieste al secondo per regione per account	Ogni regione supportata: 10	Sì	Il numero massimo di UpdatePolicyTemplate richieste al secondo.

Termini e concetti relativi al linguaggio delle politiche Amazon Verified Permissions e Cedar

È necessario comprendere i seguenti concetti per utilizzare Amazon Verified Permissions.

Concetti relativi alle autorizzazioni verificate

- [Modello di autorizzazione](#)
- [Richiesta di autorizzazione](#)
- [Risposta di autorizzazione](#)
- [Politiche considerate](#)
- [Dati contestuali](#)
- [Definizione delle politiche](#)
- [Dati dell'entità](#)
- [Autorizzazioni, autorizzazioni e principi](#)
- [Applicazione delle politiche](#)
- [Archivio delle politiche](#)
- [Politiche soddisfatte](#)
- [Differenze tra Amazon Verified Permissions e il linguaggio delle policy Cedar](#)

Concetti del linguaggio Cedar Policy

- [Autorizzazione](#)
- [Entità](#)
- [Gruppi e gerarchie](#)
- [Spazi dei nomi](#)
- [Policy](#)
- [Modello di politica](#)
- [Schema](#)

Modello di autorizzazione

Il modello di autorizzazione descrive l'ambito delle [richieste di autorizzazione](#) effettuate dall'applicazione e la base per la valutazione di tali richieste. È definito in termini di diversi tipi di risorse, azioni intraprese su tali risorse e tipi principali che eseguono tali azioni. Considera inoltre il contesto in cui vengono intraprese tali azioni.

Il controllo degli accessi basato sui ruoli (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 RBAC le decisioni attraverso l'uso di gruppi principali.

Access Control 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 ABAC le decisioni 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 effettuata da un'applicazione per valutare un insieme di politiche al fine di determinare se un responsabile può eseguire un'azione su una risorsa per un determinato contesto.

Risposta di autorizzazione

La risposta di autorizzazione è la risposta alla [richiesta di autorizzazione](#). Include una decisione di autorizzazione o rifiuto, oltre a informazioni aggiuntive, come le IDs politiche determinanti.

Politiche considerate

Le politiche considerate sono l'insieme completo di politiche che vengono selezionate da Verified Permissions per l'inclusione durante la valutazione di una [richiesta di autorizzazione](#).

Dati contestuali

I dati contestuali sono valori di attributo che forniscono informazioni aggiuntive da valutare.

Definizione delle politiche

Le politiche determinanti sono le politiche che determinano la [risposta di autorizzazione](#). Ad esempio, se esistono due [politiche soddisfatte](#), in cui una è una negazione e l'altra è una politica di autorizzazione, la politica di rifiuto sarà la politica determinante. Se esistono più politiche di autorizzazione soddisfatte e nessuna politica di divieto soddisfatto, esistono più politiche di determinazione. Nel caso in cui nessuna politica corrisponda e la risposta sia negata, non esistono politiche determinanti.

Dati dell'entità

I dati dell'entità sono dati relativi al principale, all'azione e alla risorsa. I dati delle entità rilevanti per la valutazione delle politiche sono l'appartenenza al gruppo fino alla gerarchia delle entità e i valori degli attributi del principale e della risorsa.

Autorizzazioni, autorizzazioni e principi

Verified Permissions gestisce autorizzazioni e autorizzazioni dettagliate all'interno delle applicazioni personalizzate create dall'utente.

Un principale è l'utente di un'applicazione, umano o automatico, che ha un'identità legata a un identificatore come un nome utente o un ID macchina. Il processo di autenticazione determina se il principale è realmente l'identità che dichiara di essere.

A tale identità è associato un insieme di autorizzazioni dell'applicazione che determinano le operazioni che tale preside è autorizzato a fare all'interno dell'applicazione. L'autorizzazione è il processo di valutazione di tali autorizzazioni per determinare se una persona principale è autorizzata a eseguire una particolare azione nell'applicazione. [Queste autorizzazioni possono essere espresse come politiche.](#)

Applicazione delle politiche

L'applicazione delle politiche è il processo di applicazione della decisione di valutazione all'interno dell'applicazione al di fuori delle autorizzazioni verificate. Se la valutazione delle autorizzazioni

verificate restituisce un rifiuto, l'applicazione assicurerà che al principale sia impedito l'accesso alla risorsa.

Archivio delle politiche

Un policy store è un contenitore per policy e modelli. Ogni negozio contiene uno schema utilizzato per convalidare le politiche aggiunte all'archivio. Per impostazione predefinita, ogni applicazione dispone del proprio archivio delle politiche, ma più applicazioni possono condividere un unico archivio delle politiche. Quando un'applicazione effettua una richiesta di autorizzazione, identifica l'archivio delle politiche utilizzato per valutare tale richiesta. Gli archivi di policy forniscono un modo per isolare un set di policy e possono quindi essere utilizzati in un'applicazione multi-tenant per contenere gli schemi e le politiche per ogni tenant. Una singola applicazione può avere archivi di policy separati per ogni tenant.

Nel valutare una [richiesta di autorizzazione](#), Verified Permissions considera solo il sottoinsieme delle politiche del policy store pertinenti alla richiesta. La pertinenza viene determinata in base all'ambito della politica. L'ambito identifica il principale e la risorsa specifici a cui si applica la politica e le azioni che il principale può eseguire sulla risorsa. La definizione dell'ambito aiuta a migliorare le prestazioni restringendo l'insieme delle politiche prese in considerazione.

Politiche soddisfatte

Le politiche soddisfatte sono le politiche che corrispondono ai parametri della [richiesta di autorizzazione](#).

Differenze tra Amazon Verified Permissions e il linguaggio delle policy Cedar

Amazon Verified Permissions utilizza il motore linguistico Cedar Policy per eseguire le proprie attività di autorizzazione. Tuttavia, ci sono alcune differenze tra l'implementazione nativa di Cedar e l'implementazione di Cedar che si trovano in Verified Permissions. Questo argomento identifica queste differenze.

Definizione dello spazio dei nomi

L'implementazione Verified Permissions di Cedar presenta le seguenti differenze rispetto all'implementazione nativa di Cedar:

- Verified Permissions supporta solo uno spazio dei [nomi in uno schema definito in un policy store](#).
- Le autorizzazioni verificate non consentono di creare uno spazio dei [nomi](#) con i seguenti valori: o, aws amazon cedar

Supporto per modelli di policy

Sia Verified Permissions che Cedar consentono di inserire i segnaposto nell'ambito solo per il termine `principal resource`. Tuttavia, le autorizzazioni verificate richiedono anche che nessuna delle due sia priva di vincoli. `principal resource`

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

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

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

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

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

Supporto dello schema

Verified Permissions richiede che tutti i nomi delle JSON chiavi dello schema siano stringhe non vuote. Cedar consente stringhe vuote in alcuni casi, ad esempio per le proprietà.

Supporto per tipi di estensione

Verified Permissions supporta i [tipi di estensione](#) Cedar nelle politiche, ma attualmente non supporta la loro inclusione nella definizione di uno schema o come parte del `entities` parametro delle operazioni `IsAuthorized` and `IsAuthorizedWithToken`.

I tipi di estensione includono i tipi di dati fixed point ([decimal](#)) e IP address ([ipaddr](#)).

JSONFormato Cedar per le entità

Al momento, Verified Permissions richiede di passare l'elenco delle entità da prendere in considerazione in una richiesta di autorizzazione utilizzando la struttura definita per the

[EntitiesDefinition](#), che è una matrice di elementi. [EntityItem Verified Permissions attualmente non supporta il passaggio dell'elenco di entità da considerare in una richiesta di autorizzazione in formato Cedar. JSON](#) Per requisiti specifici di formattazione delle entità da utilizzare nelle autorizzazioni verificate, consulta. [Formattazione delle entità all'interno delle politiche di Amazon Verified Permissions](#)

Definizione dei gruppi di azione

I metodi di autorizzazione Cedar richiedono un elenco delle entità da prendere in considerazione quando si valuta una richiesta di autorizzazione rispetto alle politiche.

È possibile definire le azioni e i gruppi di azioni utilizzati dall'applicazione nello schema. Tuttavia, Cedar non include lo schema come parte di una richiesta di valutazione. Invece, Cedar utilizza lo schema solo per convalidare le politiche e i modelli di policy inviati. Poiché Cedar non fa riferimento allo schema durante le richieste di valutazione, anche se nello schema sono stati definiti gruppi di azioni, è necessario includere anche l'elenco di tutti i gruppi di azioni come parte dell'elenco delle entità da passare alle operazioni di autorizzazione. API

Verified Permissions lo fa per te. Tutti i gruppi di azioni definiti nello schema vengono aggiunti automaticamente all'elenco di entità a cui si passa come parametro alle operazioni `IsAuthorized` o `IsAuthorizedWithToken`.

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

Oggetto	Limite di autorizzazioni verificate (in byte)	Limite Cedar
Dimensione della polizza ¹	10.000	Nessuno
Descrizione della politica in linea	150	Non applicabile a Cedar
Dimensioni del modello di policy	10.000	Nessuno

Oggetto	Limite di autorizzazioni verificate (in byte)	Limite Cedar
Dimensioni dello schema	10.000	Nessuno
Tipo di entità	200	Nessuno
ID Policy	64	Nessuno
ID del modello di policy	64	Nessuno
ID entità	200	Nessuno
ID dell'archivio delle politiche	64	Non applicabile a Cedar

¹ Esiste un limite di policy per policy store in Verified Permissions in base alla dimensione combinata dei principali, delle azioni e delle risorse dei criteri creati nell'archivio delle politiche. La dimensione totale di tutte le policy relative a una singola risorsa non può superare i 200.000 byte. Per le policy collegate al modello, la dimensione del modello di policy viene conteggiata una sola volta, più la dimensione di ogni set di parametri utilizzato per creare un'istanza di ogni policy collegata al modello.

Cronologia dei documenti per la Amazon Verified Permissions User Guide

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

Modifica	Descrizione	Data
Fonti di identità OIDC	Ora puoi autorizzare gli utenti dai provider di identità OpenID Connect (OIDC).	8 giugno 2024
Autorizzazione in batch con token di origine dell'identità	Ora puoi autorizzare gli utenti di un pool di utenti Amazon Cognito in un'API <code>BatchIsAuthorizedWithToken</code> unica richiesta API.	5 aprile 2024
Creazione di un archivio di policy con API Gateway	Ora puoi creare un archivio di policy da un'API esistente e da un pool di utenti Amazon Cognito.	1 aprile 2024
Concetti ed esempi relativi al contesto	Sono state aggiunte informazioni sul contesto nelle richieste di autorizzazione con autorizzazioni verificate.	1 febbraio 2024
Concetti ed esempi di autorizzazione	Sono state aggiunte informazioni sulle richieste di autorizzazione con autorizzazioni verificate.	1 febbraio 2024
AWS CloudFormation integrati	Verified Permissions supporta la creazione di fonti di identità, policy, archivi di policy e	30 giugno 2023

modelli di policy in. AWS
CloudFormation

[Versione iniziale](#)

Versione iniziale della Amazon Verified Permissions User Guide 13 giugno 2023

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.