Manual do usuário

Amazon Verified Permissions

Amazon Verified Permissions: Manual do usuário

Table of Contents

O que é o Amazon Verified Permissions?

O Amazon Verified Permissions é um serviço de autorização e gerenciamento de permissões escaláveis e refinadas para aplicações personalizadas criadas por você. O Verified Permissions permite que seus desenvolvedores criem aplicações seguras com mais rapidez ao externalizar a autorização e centralizar o gerenciamento e a administração de políticas. O Verified Permissions usa a linguagem de política Cedar para definir permissões refinadas para usuários de aplicações.

Tópicos

Autorização no Verified Permissions

O Verified Permissions fornece autorização verificando se uma entidade principal tem permissão para realizar uma ação em um recurso em um contexto específico de uma aplicação personalizada. O Verified Permissions presume que a entidade principal tenha sido previamente identificada e autenticada por outros meios, por exemplo, por meio do uso de protocolos como o OpenID Connect, um provedor hospedado como o Amazon Cognito ou outra solução de autenticação. O Verified Permissions não depende do local onde o usuário é gerenciado e de como ele foi autenticado.

O Verified Permissions é um serviço que permite aos clientes criar, manter e testar políticas no AWS Management Console. As permissões são expressas com a linguagem de política Cedar. A aplicação cliente chama APIs de autorização para avaliar as políticas do Cedar armazenadas no serviço e fornecer uma decisão de acesso para determinar se uma ação é permitida.

Linguagem de política Cedar

As políticas de autorização no Verified Permissions são escritas com a linguagem de política Cedar. O Cedar é uma linguagem de código aberto desenvolvida para escrever políticas de autorização

e tomar decisões de autorização com base nessas políticas. Ao criar uma aplicação, você precisa garantir que somente usuários autorizados possam acessar a aplicação e que os usuários façam somente o que estão autorizados a fazer. Usando o Cedar, você pode dissociar sua lógica de negócios da lógica de autorização. No código da aplicação, você prefacia as solicitações feitas nas operações com uma chamada para o mecanismo de autorização do Cedar, perguntando “Essa solicitação está autorizada?”. Em seguida, a aplicação poderá executar a operação solicitada se a decisão for “permitir” ou retornar uma mensagem de erro se a decisão for “negar”.

As permissões verificadas atualmente usam a versão 2.4 do Cedar.

Para obter mais informações sobre o Cedar, consulte o seguinte:

- Guia de referência da linguagem de política Cedar
- Repositório Cedar GitHub

Benefícios do Verified Permissions

Acelerar o desenvolvimento das aplicações

Acelere o desenvolvimento das aplicações ao dissociar a autorização da lógica de negócios.

Aplicações mais seguras

O Verified Permissions permite que os desenvolvedores criem aplicações mais seguras.

Recursos para o usuário final

O Verified Permissions permite que você forneça recursos mais avançados para o usuário final para fins de gerenciamento de permissões.

Serviços relacionados

- Amazon Cognito: o Amazon Cognito é uma plataforma de identidade para aplicações web e móveis. É um diretório de usuários, um servidor de autenticação e um serviço de autorização para credenciais da AWS e tokens de acesso do OAuth 2.0. Ao criar um repositório de políticas, você tem a opção de criar seus principais e grupos a partir de um grupo de usuários do Amazon Cognito. Para obter mais informações, consulte o Guia do desenvolvedor do Amazon Cognito.

- Amazon API Gateway — O Amazon API Gateway é um AWS serviço para criar, publicar, manter, monitorar e proteger REST, HTTP e WebSocket APIs em qualquer escala. Ao criar um repositório de políticas, você tem a opção de criar suas ações e recursos a partir de uma API no API Gateway. Para obter mais informações sobre o API Gateway, consulte o Guia do desenvolvedor do API Gateway.
- AWS IAM Identity Center: com o IAM Identity Center, você pode gerenciar a segurança de login das identidades da sua força de trabalho, também conhecidas como usuários da força de trabalho. O IAM Identity Center fornece um local onde você pode criar ou conectar usuários da força de trabalho e gerenciar centralmente seu acesso em todos os aplicativos Contas da AWS . Para obter mais informações, consulte o AWS IAM Identity Center Guia de usuário do .

Como acessar o Amazon Permissions

Você pode trabalhar com o Amazon Verified Permissions da seguinte maneira.

AWS Management Console

O console é uma interface baseada em navegador para gerenciar os recursos do Verified Permissions e da AWS . Para obter mais informações sobre como acessar o Verified Permissions pelo console, consulte Como fazer login na AWS no Guia do usuário do Início de Sessão da AWS .

- Console de permissões verificadas da Amazon

AWS Ferramentas de linha de comando

Você pode usar as ferramentas da linha de AWS comando para emitir comandos na linha de comando do seu sistema para realizar permissões e AWS tarefas verificadas. Usar a linha de comando pode ser mais rápido e mais conveniente do que o console. As ferramentas da linha de comando também são úteis se você quiser criar scripts que realizem tarefas da AWS .

AWS fornece dois conjuntos de ferramentas de linha de comando: o AWS Command Line Interface(AWS CLI) e AWS Tools for Windows PowerShello. Para obter informações sobre como instalar e usar o AWS CLI, consulte o Guia AWS Command Line Interface do usuário. Para obter informações sobre como instalar e usar as Ferramentas para Windows PowerShell, consulte o Guia AWS Tools for Windows PowerShell do Usuário.

- permissões verificadas na Referência de Comandos AWS CLI
- Permissões verificadas pela Amazon em AWS Tools for Windows PowerShell

AWS SDKs

AWS fornece SDKs (kits de desenvolvimento de software) que consistem em bibliotecas e exemplos de código para várias linguagens e plataformas de programação (Java, Python, Ruby, .NET, iOS, Android etc.). Os SDKs são uma maneira prática de criar acesso programático ao Verified Permissions e à AWS. Por exemplo, os SDKs processam tarefas como assinatura criptográfica de solicitações, gerenciamento de erros e novas tentativas automáticas de solicitações.

Para saber mais e baixar AWS SDKs, consulte Ferramentas para Amazon Web Services.

A seguir estão links para a documentação dos recursos de permissões verificadas em vários AWS SDKs.

- 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 Construções CDK

AWS Cloud Development Kit (AWS CDK) É uma estrutura de desenvolvimento de software de código aberto para definir a infraestrutura de nuvem em código e provisioná-la por meio dela. AWS CloudFormation Construções ou componentes de nuvem reutilizáveis podem ser usados para criar modelos. AWS CloudFormation Esses modelos podem então ser usados para implantar sua infraestrutura de nuvem.

Para saber mais e baixar AWS CDKs, consulte AWS Cloud Development Kit.

A seguir estão links para a documentação de AWS CDK recursos de permissões verificadas, como construções.

- Construção de CDK L2 de permissões verificadas pela Amazon

API do Verified Permissions

Você pode acessar as Permissões Verificadas e AWS programaticamente usando a API de Permissões Verificadas, que permite emitir solicitações HTTPS diretamente para o serviço.

Quando você usa a API do , deve incluir código para assinar digitalmente solicitações usando suas credenciais.

- Guia de referência da API de permissões verificadas da Amazon

Preços do Verified Permissions

O Verified Permissions fornece preços progressivos com base na quantidade mensal de solicitações de autorização feitas por suas aplicações ao Verified Permissions. Também há preços para ações de gerenciamento de políticas com base na quantidade mensal de solicitações de API de política de cURL (URL do cliente) feitas por suas aplicações ao Verified Permissions.

Para obter uma lista completa de tarifas e preços do Verified Permissions, consulte Preços do Amazon Verified Permissions.

Para ver sua fatura, acesse o Painel de gerenciamento de custos e faturamento no console do AWS Billing and Cost Management. Sua fatura contém links para relatórios de uso que fornecem detalhes sobre sua conta. Para saber mais sobre Conta da AWS faturamento, consulte o Guia do AWS Billing usuário.

Se você tiver dúvidas sobre AWS faturamento, contas e eventos, entre em contato com AWS Support.

Termos e conceitos do Amazon Verified Permissions

Você deve entender os seguintes conceitos para usar o Amazon Verified Permissions.

Conceitos do Verified Permissions

- Modelo de autorização
- Solicitação de autorização
- Resposta de autorização
- Políticas consideradas
- Dados de contexto
- Políticas determinantes
- Dados da entidade
- Permissões, autorização e entidades principais
- Aplicação de políticas
- Armazenamentos de políticas
- Políticas atendidas
- Diferenças entre o Verified Permissions e o Cedar

Conceitos da linguagem de política Cedar

- Autorização
- Entidade
- Grupos e hierarquias
- Namespaces
- Política
- Modelo de política
- Esquema

Modelo de autorização

O modelo de autorização descreve o escopo das solicitações de autorização feitas pela aplicação e é a base para avaliar essas solicitações. Ele é definido com base nos diferentes tipos de recursos,

das ações realizadas nesses recursos e dos tipos de entidades principais que realizam essas ações. Ele também considera o contexto em que essas ações estão sendo realizadas.

O controle de acesso baseado em função (RBAC) é uma base de avaliação na qual as funções são definidas e associadas a um conjunto de permissões. Essas funções podem, então, ser atribuídas a uma ou mais identidades. A identidade atribuída adquire as permissões associadas à função. Se as permissões associadas à função forem modificadas, a modificação afetará automaticamente qualquer identidade à qual a função tenha sido atribuída. O Cedar pode oferecer suporte às decisões do RBAC por meio do uso de grupos de entidades principais.

O controle de acesso baseado em função (RBAC) é uma base de avaliação na qual as permissões associadas a uma identidade são determinadas pelos atributos dessa identidade. O Cedar pode oferecer suporte às decisões do ABAC por meio do uso de condições de política que fazem referência aos atributos da entidade principal.

A linguagem de política Cedar permite a combinação do RBAC e do ABAC em uma única política, fazendo com que as permissões sejam definidas para um grupo de usuários, que têm condições baseadas em atributos.

Solicitação de autorização

Uma solicitação de autorização é uma solicitação feita ao Verified Permissions por uma aplicação para avaliar um conjunto de políticas, a fim de determinar se uma entidade principal pode realizar uma ação em um recurso para um determinado contexto.

Resposta de autorização

A resposta de autorização é a resposta à solicitação de autorização. Ela inclui a decisão de permitir ou negar, além de informações adicionais, como os IDs das políticas determinantes.

Políticas consideradas

As políticas consideradas são o conjunto completo de políticas selecionadas pelo Verified Permissions para inclusão ao avaliar uma solicitação de autorização.

Dados de contexto

Os dados de contexto são valores de atributos que fornecem informações adicionais a serem avaliadas.

Políticas determinantes

As políticas determinantes são aquelas que determinam a resposta da autorização. Por exemplo, se houver duas políticas atendidas, em que uma é a negação e a outra é a permissão, a política de negação será a política determinante. Se houver várias políticas de permissão atendidas e nenhuma política de proibição atendida, haverá várias políticas determinantes. Caso nenhuma política corresponda e a resposta seja uma negação, não haverá políticas determinantes.

Dados da entidade

Os dados da entidade são dados sobre a entidade principal, a ação e o recurso. Os dados de entidade relevantes para a avaliação da política são a associação ao grupo em toda a hierarquia de entidades e os valores de atributo da entidade principal e do recurso.

Permissões, autorização e entidades principais

O Verified Permissions gerencia permissões e autorizações refinadas nas aplicações personalizadas criadas por você.

A entidade principal é o usuário de uma aplicação, seja ele um ser humano ou uma máquina, que tem uma identidade vinculada a um identificador, como um nome de usuário ou um ID de máquina. O processo de autenticação determina se a entidade principal é, de fato, a identidade que afirma ser.

Associado a essa identidade está um conjunto de permissões de aplicação que determina o que essa entidade principal tem permissão para fazer nessa aplicação. A autorização é o processo de avaliação dessas permissões para determinar se uma entidade principal tem permissão para realizar uma ação específica na aplicação. Essas permissões podem ser expressas como políticas.

Aplicação de políticas

A aplicação da política é o processo de aplicar a decisão de avaliação na aplicação, fora do Verified Permissions. Se a avaliação do Verified Permissions retornar uma negação, a aplicação garantirá que a entidade principal foi impedida de acessar o recurso.

Armazenamentos de políticas

Um armazenamento de políticas é um contêiner de políticas e modelos. Cada armazenamento contém um esquema, que é usado para validar as políticas adicionadas ao armazenamento. Por

padrão, cada aplicação tem seu próprio armazenamento de políticas, mas vários aplicações podem compartilhar um único armazenamento de políticas. Quando uma aplicação faz uma solicitação de autorização, ele identifica o armazenamento de políticas usado para avaliar essa solicitação. Os armazenamentos de políticas são uma maneira de isolar um conjunto de políticas e, portanto, podem ser usados em uma aplicação multilocatária para reter os esquemas e políticas de cada locatário. Uma única aplicação pode ter armazenamentos de políticas separados para cada locatário.

Ao avaliar uma solicitação de autorização, o Verified Permissions considera apenas o subconjunto das políticas contidas no armazenamento que são relevantes para a solicitação. A relevância é determinada com base no escopo da política. O escopo identifica a entidade principal e o recurso específicos aos quais a política se aplica, bem como as ações que a entidade principal pode realizar no recurso. A definição do escopo ajuda a melhorar o desempenho ao restringir o conjunto de políticas consideradas.

Políticas atendidas

Políticas atendidas são as políticas que correspondem aos parâmetros da solicitação de autorização.

Diferenças entre o Verified Permissions e o Cedar

O Amazon Verified Permissions usa o mecanismo de linguagem de política Cedar para realizar suas tarefas de autorização. No entanto, existem algumas diferenças entre a implementação nativa do Cedar e a implementação do Cedar encontrada no Verified Permissions. Este tópico identifica essas diferenças.

Definição de namespace

A implementação do Cedar no Verified Permissions tem as seguintes diferenças em relação à implementação nativa do Cedar:

- O Verified Permissions oferece suporte somente a um namespace em um esquema definido em um armazenamento de políticas.
- O Verified Permissions não permite que você crie um namespace com os seguintes valores: `aws`, `amazon` ou `cedar`.

Suporte do modelo de política

O Verified Permissions e o Cedar permitem espaços reservados no escopo apenas para `principal` e `resource`. No entanto, o Verified Permissions também requer que nem `principal` nem `resource` sejam irrestritos.

A política a seguir é válida no Cedar, mas é rejeitada pelo Verified Permissions porque `principal` não tem restrições.

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

Os dois exemplos a seguir são válidos no Cedar e no Verified Permissions porque tanto `principal` quanto `resource` têm restrições.

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

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

Suporte ao esquema

O Verified Permissions requer que todos os nomes de chave JSON do esquema sejam strings não vazias. O Cedar permite strings vazias em alguns casos; por exemplo, nas propriedades.

Suporte ao tipo de extensão

O Verified Permissions oferece suporte aos tipos de extensão do Cedar nas políticas, mas atualmente não oferece suporte para incluí-los na definição de um esquema ou como parte do parâmetro `entities` das operações `IsAuthorized` e `IsAuthorizedWithToken`.

Os tipos de extensão incluem os tipos de dados de ponto fixo (`decimal`) e de endereço IP (`ipaddr`).

Formato JSON do Cedar para entidades

No momento, o Verified Permissions requer que você passe a lista de entidades a serem consideradas em uma solicitação de autorização usando a estrutura definida para EntitiesDefinition, que é uma matriz de elementos EntityItem. Atualmente, o Verified Permissions não oferece suporte à passagem da lista de entidades a serem consideradas em uma solicitação de autorização no formato

JSON do Cedar. Para requisitos específicos de formatação das entidades para uso no Verified Permissions, consulte Formatação de entidades no Amazon Verified Permissions.

Definição de grupos de ação

Os métodos de autorização do Cedar requerem uma lista das entidades a serem consideradas durante a avaliação de uma solicitação de autorização com base nas políticas.

Você pode definir as ações e os grupos de ação usados pela sua aplicação no esquema. No entanto, o Cedar não inclui o esquema como parte de uma solicitação de avaliação. Em vez disso, o Cedar usa o esquema somente para validar as políticas e os modelos de políticas que você envia. Como o Cedar não faz referência ao esquema durante as solicitações de avaliação, mesmo que você tenha definido grupos de ação no esquema, você também deve incluir a lista de todos os grupos de ação como parte da lista de entidades que você deve passar para as operações da API de autorização.

O Verified Permissions faz isso para você. Todos os grupos de ação definidos em seu esquema são automaticamente anexados à lista de entidades que você passa como parâmetro para as operações `IsAuthorized` ou `IsAuthorizedWithToken`.

Limites de comprimento e tamanho

O Verified Permissions oferece suporte ao armazenamento sob a forma de armazenamentos de políticas que reterão seu esquema, políticas e modelos de políticas. Esse armazenamento faz com que o Verified Permissions aplique alguns limites de comprimento e tamanho que não são relevantes para o Cedar.

Objeto	Limite do Verified Permissions (em bytes)	Limite do Cedar
Tamanho da política[1]	10.000	Nenhum
Descrição da política em linha	150	Não aplicável ao Cedar
Tamanho do modelo de política	10.000	Nenhum
Tamanho do esquema	10.000	Nenhum
Tipo de entidade	200	Nenhum

Objeto	Limite do Verified Permissions (em bytes)	Limite do Cedar
Policy ID (ID da política)	64	Nenhum
ID do modelo de política	64	Nenhum
ID da entidade	200	Nenhum
ID do armazenamento de políticas	64	Não aplicável ao Cedar

[1] No Verified Permissions, há um limite para políticas por armazenamento de políticas com base no tamanho combinado de entidades principais, ações e recursos de políticas criadas no armazenamento de políticas. [3] O tamanho total de todas as políticas relacionadas a um único recurso não pode exceder 200.000 bytes. Para políticas vinculadas a modelos, o tamanho do modelo de política é contabilizado somente uma vez, mais o tamanho de cada conjunto de parâmetros usados para instanciar cada política vinculada a modelo.

Conceitos básicos do Verified Permissions

Use este tutorial para começar a usar o Amazon Verified Permissions.

Tópicos

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções on-line.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando https://aws.amazon.com/ e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, digite sua senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Signing in as the root user (Fazer login como usuário-raiz) no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do IAM usuário.

Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

   Para obter instruções, consulte Habilitar AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Create a permission set no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Add groups no Guia do usuário do AWS IAM Identity Center .

IAM políticas para permissões verificadas

O Verified Permissions gerencia as permissões dos usuários na sua aplicação. Para que seu aplicativo chame as APIs de Permissões Verificadas ou para que AWS Management Console os usuários possam gerenciar as políticas do Cedar em um repositório de políticas de Permissões Verificadas, você deve adicionar as permissões necessárias IAM .

Políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criação de IAM políticas no Guia do IAM usuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas (listadas abaixo). Não é possível especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexado. Para saber mais sobre todos os elementos que você pode usar em uma política JSON, consulte a referência aos elementos da política IAM JSON no Guia do IAM usuário.

Ação	Descrição
CreatePolicyStore	Ação para criar um novo armazenamento de políticas.
DeletePolicyStore	Ação para excluir um armazenamento de políticas.

Ação	Descrição
ListPolicyStores	Ação para listar todos os repositórios de políticas no Conta da AWS.
CreatePolicy	Ação para criar uma política do Cedar em um armazenamento de políticas. Você pode criar uma política estática ou uma política vinculada a um modelo de política.
DeletePolicy	Ação para excluir uma política de um armazenamento de políticas.
GetPolicy	Ação para recuperar informações sobre uma política específica.
ListPolicies	Ação para listar todas as políticas de um armazenamento de políticas.
IsAuthorized	Ação para obter uma resposta de autorização com base nos parâmetros descritos na solicitação de autorização.

Exemplo IAM de política para permissão para a CreatePolicy ação:

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

Crie seu primeiro armazenamento de políticas do Verified Permissions

Ao entrar no console do Verified Permissions pela primeira vez, você pode especificar como criará seu primeiro armazenamento de políticas e a política do Cedar. Siga o procedimento de login adequado para o tipo de usuário, conforme descrito no tópico Como fazer login na AWS no Guia do usuário do AWS Sign-In. Na página inicial do console, selecione o serviço Amazon Verified Permissions. Escolha Comece a usar.

Criar um armazenamento de políticas de exemplo

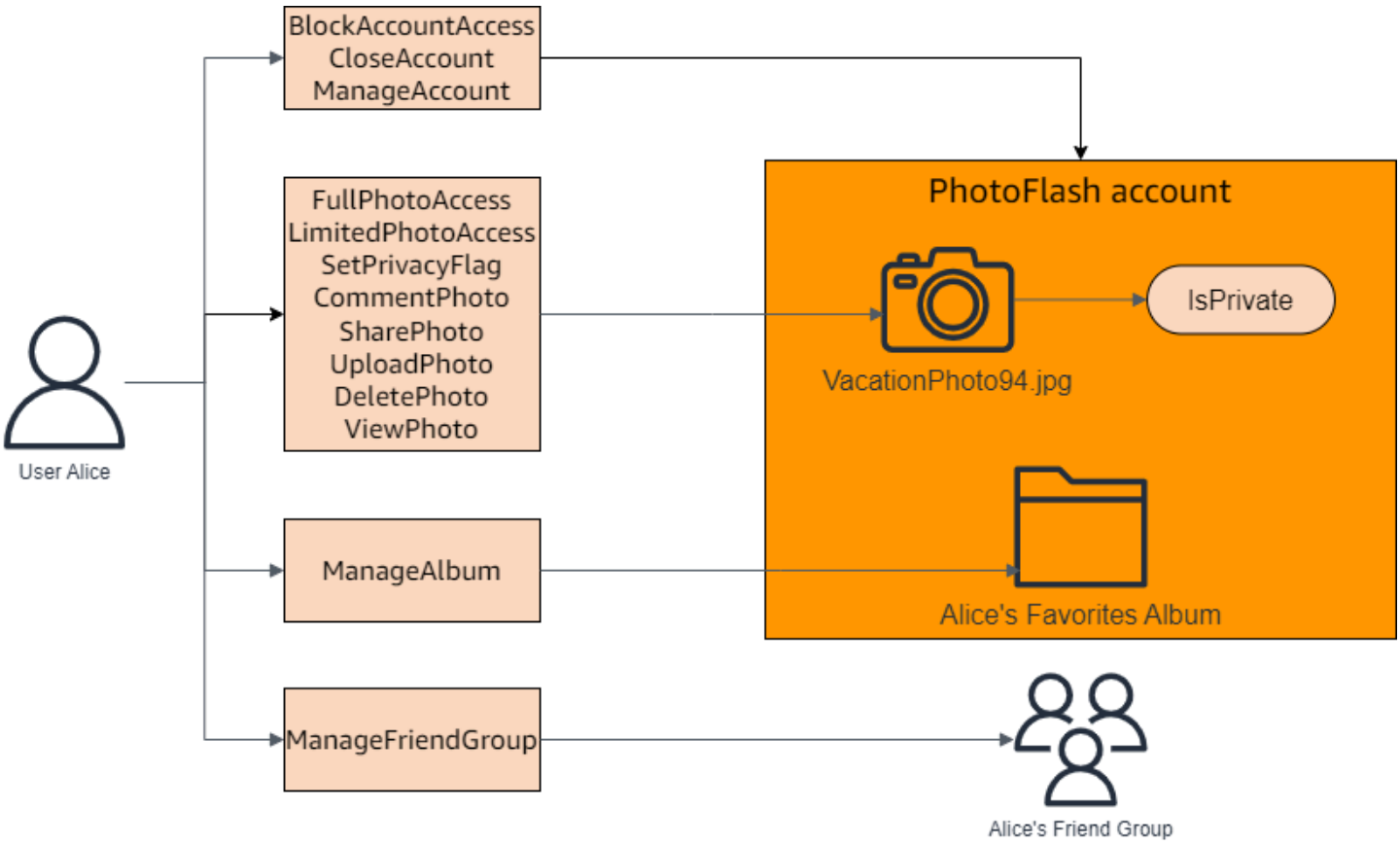
Se esta é a primeira vez que você usa o Verified Permissions, é recomendável usar um dos exemplos de armazenamentos de políticas para se familiarizar com o funcionamento do Verified Permissions. Os exemplos de armazenamentos de políticas fornecem políticas predefinidas e um esquema.

Para criar um armazenamento de políticas usando o método de configuração Exemplo de armazenamento de políticas

1. No console Permissões verificadas, selecione Criar novo repositório de políticas.
2. Na seção Opções iniciais, escolha Exemplo de armazenamento de políticas.
3. Na seção Projeto de exemplo, escolha o tipo de exemplo de aplicação do Verified Permissions a ser usado. Para este tutorial, escolha o repositório PhotoFlashde políticas.
4. Um namespace é gerado automaticamente para o esquema do exemplo de armazenamento de políticas com base no projeto de exemplo que você escolheu.
5. Escolha Criar armazenamento de políticas.

   Seu armazenamento de políticas é criado com políticas, modelos de políticas e um esquema para o exemplo de armazenamento de políticas.

O diagrama abaixo ilustra as relações entre os PhotoFlash exemplos de ações do repositório de políticas e os tipos de recursos aos quais elas se aplicam.



Criação de políticas vinculadas a modelos para um exemplo de armazenamento de políticas

O PhotoFlash exemplo de armazenamento de políticas inclui políticas, modelos de políticas e um esquema. Você pode criar políticas vinculadas a modelos com base nos modelos de políticas incluídos no exemplo de armazenamento de políticas.

Para criar políticas vinculadas a modelos para um exemplo de armazenamento de políticas

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. Escolha Criar política e, em seguida, escolha Criar política vinculada a modelo.
4. Escolha o botão de opção ao lado do modelo de política com a descrição Conceder acesso total a fotos compartilhadas não privadas e escolha Avançar.
5. Para Diretor, insira`PhotoFlash::User::"Alice"`. Em Recurso, insira`PhotoFlash::Album::"Bob-Vacation-Album"`.

6. Escolha Criar política vinculada a modelo.

   A nova política vinculada a modelo é exibida em Políticas.
7. Crie outra política vinculada ao modelo para o repositório de políticas de PhotoFlash amostra. Escolha Criar política e, em seguida, escolha Criar política vinculada a modelo.
8. Escolha o botão de opção ao lado do modelo de política com a descrição Conceder acesso limitado a fotos compartilhadas não privadas e escolha Avançar.
9. Para Diretor, insira`PhotoFlash::FriendGroup::"MySchoolFriends"`. Em Recurso, insira`PhotoFlash::Album::"Alice's favorite album"`.
10. Escolha Criar política vinculada a modelo.

    A nova política vinculada a modelo é exibida em Políticas.

Testaremos as novas políticas vinculadas a modelos na próxima seção do tutorial. Para obter mais exemplos de valores que você pode usar para criar uma política vinculada a um modelo, consulte. PhotoFlash PhotoFlashexemplos de políticas vinculadas a modelos

Teste de um exemplo de armazenamento de políticas

Após criar o exemplo de armazenamento de políticas e as políticas vinculadas a modelos, você pode testar os exemplos de políticas estáticas do Verified Permissions e suas novas políticas vinculadas a modelos executando uma solicitação de autorização simulada no banco de teste do Verified Permissions.

Dependendo de quando você criou seu repositório de políticas de amostra, seus modelos de política podem ser diferentes das referências neste procedimento. Antes de começar essa parte do tutorial, verifique se você tem cada modelo de política a seguir no seu repositório de políticas de PhotoFlash exemplo. Se sua política não estiver alinhada com essas políticas, edite as políticas existentes ou crie um novo repositório de políticas a partir da opção PhotoFlashProjeto de amostra.

Conceda acesso total a fotos compartilhadas não privadas

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

Conceda acesso limitado a fotos compartilhadas não privadas

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

Para testar exemplos de políticas de armazenamento de políticas

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Banco de testes.
3. Escolha Modo visual.
4. Na seção Principal, escolha PhotoFlash: :User entre os tipos principais em seu esquema. Digite um identificador para o usuário na caixa de texto. Por exemplo, `Alice`.
5. Não escolha Adicionar um pai para a entidade principal.
6. Para o atributo Conta: Entidade, certifique-se de que a entidade PhotoFlash: :Account esteja selecionada. Digite um identificador para a conta. Por exemplo, `Alice-account`.
7. Na seção Recurso, escolha o tipo de recurso PhotoFlash: :Photo. Digite um identificador para a foto na caixa de texto. Por exemplo, `photo.jpeg`.
8. Escolha Adicionar um pai e escolha PhotoFlash: :Conta para o tipo de entidade. Digite o mesmo identificador da conta pai para a foto que você especificou no campo Account: Entity para o usuário. Por exemplo, `Alice-account`.
9. Na seção Ação, escolha PhotoFlash: :Ação::” ViewPhoto "na lista de ações válidas.
10. Na seção Entidades adicionais, escolha Adicionar esta entidade para adicionar a entidade de conta sugerida.
11. Escolha Executar solicitação de autorização na parte superior da página para simular a solicitação de autorização para as políticas do Cedar no exemplo de armazenamento de políticas. O banco de testes exibirá a decisão de permitir a solicitação.

A tabela a seguir fornece valores adicionais para a entidade principal, o recurso e a ação que você pode testar com o banco de testes do Verified Permissions. A tabela inclui a decisão da solicitação

de autorização com base nas políticas estáticas incluídas no repositório de políticas de PhotoFlash amostra e nas políticas vinculadas ao modelo que você criou na seção anterior.

Valor da entidade principal	Valor Account: Entity da entidade principal	Valor do recurso	Valor pai do recurso	Ação	Decisão de autorização
PhotoFlash: :Usuário | Alice	PhotoFlash: :Conta | Conta Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Conta | Conta Bob	PhotoFlash: :Ação::"" ViewPhoto	Deny
PhotoFlash: :Usuário | Alice	PhotoFlash: :Conta | Conta Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Conta | Conta Alice	PhotoFlash: :Ação::"" ViewPhoto	Permitir
PhotoFlash: :Usuário | Alice	PhotoFlash: :Conta | Conta Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash: :Álbum | Bob - Álbum de férias	PhotoFlash: :Ação::"" ViewPhoto	Permitir
PhotoFlash: :Usuário | Alice	PhotoFlash: :Conta | Conta Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash: :Álbum | Bob - Álbum de férias	PhotoFlash: :Ação::"" DeletePhoto	Deny
PhotoFlash: :Usuário | Alice	PhotoFlash: :Conta | Conta Alice	PhotoFlash: :Foto | Bob-photo.jpeg, IsPrivate: Boolean | verdadeiro	PhotoFlash: :Álbum | Bob - Álbum de férias	PhotoFlash: :Ação::"" ViewPhoto	Deny
PhotoFlash: :Usuário | Jane,	PhotoFlash: :Conta |	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | O álbum	PhotoFlash: :Ação::"" ViewPhoto	Permitir

Valor da entidade principal	Valor Account: Entity da entidade principal	Valor do recurso	Valor pai do recurso	Ação	Decisão de autorização
PhotoFlash:: FriendGroup | MySchoolF riends	Conta de Jane		favorito de Alice		
PhotoFlas h: :Usuário | Jane, PhotoFlash:: FriendGroup | MySchoolF riends	PhotoFlas h: :Conta | Conta de Jane	PhotoFlas h: :Foto | photo.jpeg	PhotoFlas h: :Album | O álbum favorito de Alice	PhotoFlas h: :Ação::"" DeletePhoto	Deny

Crie um repositório de políticas com uma API conectada e um provedor de identidade

Um caso de uso comum das Permissões Verificadas da Amazon é autorizar solicitações de um cliente de aplicativo para uma API de back-end. AWS tem um serviço para autenticação de usuários do aplicativo: Amazon Cognito. AWS também tem um serviço para APIs hospedadas de forma segura: Amazon API Gateway. Ao combinar um repositório de políticas de Permissões Verificadas com esses dois Serviços da AWS, você pode conectar a autenticação do grupo de usuários e a autorização da API em seu aplicativo com um conjunto consistente e centralizado de políticas. Os repositórios de políticas de permissões verificadas têm suporte integrado para fontes de identidade do grupo de usuários do Amazon Cognito e APIs do API Gateway.

Para criar um repositório de políticas vinculado a um pool de usuários e API existentes, escolha Configurar com o Cognito e o API Gateway ao criar um novo repositório de políticas.

Um repositório de políticas vinculado à API provisiona automaticamente seu modelo de autorização e recursos para solicitações de autorização. O processo de criação do Set up with Cognito and API Gateway gera um repositório de políticas com uma fonte de identidade do grupo de usuários e

um autorizador Lambda que conecta o API Gateway às permissões verificadas. Inicialmente, você pode autorizar solicitações de API com base nas associações de grupos dos usuários. Por exemplo, as Permissões verificadas podem conceder acesso somente aos usuários que são membros do `Directors` grupo.

Conforme seu aplicativo cresce, você pode implementar uma autorização refinada com atributos de usuário e escopos do OAuth 2.0. Por exemplo, as Permissões verificadas podem conceder acesso somente a usuários que tenham um `email` atributo no domínio`mycompany.co.uk`.

Depois de automatizar o modelo de autorização para sua API, sua responsabilidade restante é autenticar usuários, gerar solicitações de API em seu aplicativo e manter seu armazenamento de políticas.

Para saber mais, consulte Armazenamentos de políticas vinculados à API.

Armazenamentos de políticas do Amazon Verified Permissions

Um armazenamento de políticas é um contêiner para políticas e modelos de políticas. Cada armazenamento de políticas contém um esquema usado para validar políticas adicionadas ao repositório de políticas. É recomendável criar um armazenamento de políticas por aplicação ou um armazenamento de políticas por locatário para aplicações multilocatárias. Você deve especificar um armazenamento de políticas ao fazer uma solicitação de autorização.

É recomendável usar namespaces para entidades do Cedar nos armazenamentos de políticas para evitar ambiguidades. Um namespace é um prefixo de string para um tipo, separado por um par de dois-pontos (`::`) como delimitador. O Verified Permissions oferece suporte a um namespace por armazenamento de políticas. Para obter mais informações, consulte Namespaces no Guia de referência da linguagem de política Cedar.

Tópicos

Criação de armazenamentos de políticas do Verified Permissions

Você pode criar um armazenamento de políticas usando um dos seguintes métodos:

- Siga uma configuração guiada — Você definirá um tipo de recurso com ações válidas e um tipo principal antes de criar sua primeira política.
- Configure com o API Gateway e uma fonte de identidade — Defina suas entidades principais com usuários que fazem login com um provedor de identidade (IdP) e suas ações e entidades de recursos a partir de uma API do Amazon API Gateway. Recomendamos essa opção se você quiser que seu aplicativo autorize solicitações de API com a associação de grupos de usuários.
- Comece com um exemplo de armazenamento de políticas — Escolha um exemplo predefinido de armazenamento de políticas de projeto. É recomendável o uso desta opção se você estiver em busca de informações sobre o Verified Permissions e quiser ver e testar exemplos de políticas.

- Crie um repositório de políticas vazio — Você mesmo definirá o esquema e todas as políticas de acesso. É recomendável o uso desta opção se você já estiver familiarizado com a configuração de um armazenamento de políticas.

Guided setup

Para criar um armazenamento de políticas por meio do método de Configuração guiada

O assistente de configuração guiada conduz você pelo processo de criação da primeira iteração do seu armazenamento de políticas. Você criará um esquema para seu primeiro tipo de recurso, descreverá as ações aplicáveis a esse tipo de recurso e o tipo de entidade principal para o qual você está concedendo permissões. Em seguida, você criará sua primeira política. Após concluir esse assistente, você poderá adicionar conteúdo ao seu armazenamento de políticas, estender o esquema para descrever outros tipos de recursos e entidades principais, além de criar políticas e modelos adicionais.

1. No console Permissões verificadas, selecione Criar novo repositório de políticas.
2. Na seção Opções iniciais, escolha Configuração guiada.
3. Insira uma descrição do repositório de políticas. Esse texto pode ser o que for adequado à sua organização como uma referência amigável à função do repositório de políticas atual, por exemplo, atualizações meteorológicas.
4. Na seção Detalhes, digite um Namespace para seu esquema.
5. Escolha Próximo.
6. Na janela Tipo de recurso, digite um nome para seu tipo de recurso.
7. (Opcional) Escolha Adicionar um atributo para adicionar atributos de recursos. Digite o Nome do atributo e escolha um Tipo de atributo para cada atributo do recurso. Especifique se cada atributo será Obrigatório. O Verified Permissions usa os valores de atributo especificados ao verificar as políticas com base no esquema. Para remover um atributo adicionado para o tipo de recurso, escolha Remover ao lado do atributo.
8. No campo Ações, digite as ações a serem autorizadas para o tipo de recurso especificado. Para adicionar ações extras para o tipo de recurso, escolha Adicionar uma ação. Para remover uma ação adicionada para o tipo de recurso, escolha Remover ao lado da ação.
9. No campo Nome do tipo de entidade principal, digite o nome para um tipo de entidade principal que usará as ações especificadas para seu tipo de recurso.
10. Escolha Próximo.

11. Na janela Tipo de entidade principal, escolha a origem de identidade para seu tipo de entidade principal.

    - Escolha Personalizado se o ID e os atributos da entidade principal forem fornecidos diretamente pelo Verified Permissions. Escolha Adicionar um atributo para adicionar atributos de entidade principal. Digite o Nome do atributo e escolha um Tipo de atributo para cada atributo da entidade principal. O Verified Permissions usa os valores de atributo especificados ao verificar as políticas com base no esquema. Para remover um atributo adicionado para o tipo de entidade principal, escolha Remover ao lado do atributo.
    - Escolha Grupo de usuários do Cognito se o ID e os atributos da entidade principal forem fornecidos a partir de um ID ou token de acesso gerado pelo Amazon Cognito. Escolha Conectar grupo de usuários. Selecione a Região da AWS e digite o ID do grupo de usuários do Amazon Cognito com o qual você se conectará. Selecione Conectar. Para obter mais informações, consulte Autorização com o Amazon Verified Permissions no Guia do desenvolvedor do Amazon Cognito.
12. Escolha Próximo.
13. Na seção Detalhes da política, digite uma Descrição da política para sua primeira política Cedar (essa descrição é opcional).
14. No campo Escopo das entidades principais, escolha as entidades principais que receberão permissões da política.

    - Escolha Entidade principal específica para aplicar a política a uma entidade principal específica. Escolha a entidade principal no campo Entidade principal que terá permissão para executar ações e digite um identificador de entidade para a entidade principal.
    - Escolha Todas as entidades principais para aplicar a política a todas as entidades principais do armazenamento de políticas.
15. No campo Escopo dos recursos, escolha em quais recursos as entidades principais especificadas serão autorizadas a atuar.

    - Escolha Recurso específico para aplicar a política a um recurso específico. Escolha o recurso no campo Recurso ao qual esta política deve ser aplicada e digite um identificador de entidade para o recurso.
    - Escolha Todos os recursos para aplicar a política a todos os recursos do armazenamento de políticas.
16. No campo Escopo dos recursos, escolha em quais recursos as entidades principais especificadas serão autorizadas a atuar.

- Escolha Conjunto específico de ações para aplicar a política a ações específicas. Marque as caixas de seleção ao lado das ações no campo Ações às quais esta política deve ser aplicada.
- Escolha Todas as ações para aplicar a política a todas as ações do armazenamento de políticas.

17. Revise a política na seção Visualização da política. Escolha Criar armazenamento de políticas.

Set up with API Gateway and an identity source

Para criar um repositório de políticas usando o método Configurar com o API Gateway e um método de configuração de fonte de identidade

A opção API Gateway protege as APIs com políticas de permissões verificadas, projetadas para tomar decisões de autorização dos grupos ou funções dos usuários. Essa opção cria um repositório de políticas para testar a autorização com grupos de origem de identidade e uma API com um autorizador Lambda.

Os usuários e seus grupos em um IdP se tornam seus principais (tokens de ID) ou seu contexto (tokens de acesso). Os métodos e caminhos em uma API do API Gateway se tornam as ações que suas políticas autorizam. Seu aplicativo se torna o recurso. Como resultado desse fluxo de trabalho, o Verified Permissions cria um repositório de políticas, uma função Lambda e um autorizador de API Lambda. Você deve atribuir o autorizador Lambda à sua API depois de concluir esse fluxo de trabalho.

1. No console Permissões verificadas, selecione Criar novo repositório de políticas.
2. Na seção Opções iniciais, escolha Configurar com o API Gateway e uma fonte de identidade e selecione Avançar.
3. Na etapa Importar recursos e ações, em API, escolha uma API que funcionará como modelo para os recursos e ações do seu repositório de políticas.

    a. Escolha um estágio de implantação entre os estágios configurados em sua API e selecione Importar API. Para obter mais informações sobre os estágios da API, consulte Como configurar um estágio para uma API REST no Guia do desenvolvedor do Amazon API Gateway.

    b. Visualize seu mapa de recursos e ações importados.

c. Para atualizar recursos ou ações, modifique seus caminhos ou métodos de API e selecione Importar API.

d. Quando estiver satisfeito com suas escolhas, escolha Avançar.

4. Em Fonte de identidade, escolha um tipo de provedor de identidade. Você pode escolher um grupo de usuários do Amazon Cognito ou um tipo de IdP do OpenID Connect (OIDC).

5. Se você escolheu o Amazon Cognito:

   a. Escolha um grupo de usuários no mesmo Região da AWS e Conta da AWS no seu repositório de políticas.

   b. Escolha o tipo de token a ser passado para a API que você deseja enviar para autorização. Qualquer um dos tipos de token contém grupos de usuários, a base desse modelo de autorização vinculado à API.

   c. Em Validação do cliente do aplicativo, você pode limitar o escopo de um armazenamento de políticas a um subconjunto dos clientes do aplicativo Amazon Cognito em um grupo de usuários multilocatários. Para exigir que o usuário se autentique com um ou mais clientes de aplicativos especificados em seu grupo de usuários, selecione Aceitar somente tokens com IDs de cliente de aplicativo esperados. Para aceitar qualquer usuário que se autentique com o grupo de usuários, selecione Não validar IDs de cliente do aplicativo.

   d. Escolha Próximo.

6. Se você escolheu o provedor OIDC:

   a. Em URL do emissor, insira a URL do seu emissor do OIDC. Esse é o endpoint do serviço que fornece o servidor de autorização, as chaves de assinatura e outras informações sobre seu provedor, por exemplo`https://auth.example.com`. Seu URL de emissor deve hospedar um documento de descoberta do OIDC em. `/.well-known/openid-configuration`

   b. Em Tipo de token, escolha o tipo de OIDC JWT que você deseja que seu aplicativo envie para autorização. Para ter mais informações, consulte Trabalhando com fontes de identidade em esquemas e políticas.

   c. Em Declarações de token, escolha como você deseja configurar os atributos do usuário em seu repositório de políticas. Esses atributos definem as declarações que suas políticas podem referenciar.

      i. Escolha uma fonte de reclamação.

A. Para fornecer um token de amostra, escolha Extrair da carga útil do JWT e cole a carga útil de um JWT do tipo de token escolhido. Os JWTs contêm um cabeçalho, uma carga útil e uma assinatura. Seu JWT de amostra deve ser decodificado e somente para carga útil. Para analisar a carga, selecione Extrair.

B. Para inserir seu próprio conjunto de atributos, escolha Inserir declarações manualmente.

ii. Insira ou confirme cada nome de solicitação de token e tipo de valor de declaração que você deseja adicionar aos atributos do usuário principal ou do contexto de ação em seu esquema.

d. Em Declarações de usuário e grupo, escolha uma reivindicação de usuário para a fonte de identidade. Normalmente`sub`, essa é uma afirmação do seu ID ou token de acesso que contém o identificador exclusivo da entidade a ser avaliada. As identidades do IdP OIDC conectado serão mapeadas para o tipo de usuário em seu repositório de políticas.

e. Em Declarações de usuário e grupo, escolha uma afirmação de grupo para a fonte de identidade. Normalmente`groups`, essa é uma reivindicação do seu ID ou token de acesso que contém uma lista dos grupos do usuário. Seu repositório de políticas autorizará solicitações com base na associação ao grupo.

f. Em Validação de público ou IDs de cliente, insira os IDs de cliente ou URLs de público que você deseja que seu repositório de políticas aceite nas solicitações de autorização, se houver. Para tokens de acesso, insira um valor de reivindicação de público, como`https://myapp.example.com`. Para tokens de ID, insira um ID de cliente como`1example23456789`.

g. Escolha Próximo.

7. Se você escolheu o Amazon Cognito, o Verified Permissions consulta seu grupo de usuários em busca de grupos. Para provedores do OIDC, insira os nomes dos grupos manualmente. A etapa Atribuir ações aos grupos cria políticas para seu repositório de políticas que permitem que os membros do grupo realizem ações.

   a. Escolha ou adicione os grupos que você deseja incluir em suas políticas.

   b. Atribua ações a cada um dos grupos que você selecionou.

   c. Escolha Próximo.

8. Em Implantar integração de aplicativos, revise as etapas que as Permissões Verificadas seguirão para criar seu repositório de políticas e o autorizador Lambda.

9. Quando você estiver pronto para criar os novos recursos, escolha Criar e implantar.

10. Mantenha a etapa de status do repositório de políticas aberta em seu navegador para monitorar o progresso da criação de recursos por meio de permissões verificadas.
11. Depois de algum tempo, normalmente cerca de uma hora, ou quando a etapa do autorizador Deploy Lambda mostrar Sucesso, configure seu autorizador.

    As permissões verificadas terão criado uma função Lambda e um autorizador Lambda em sua API. Escolha Abrir API para navegar até sua API.

    Para saber como atribuir um autorizador Lambda, consulte Usar autorizadores Lambda do API Gateway no Guia do desenvolvedor do Amazon API Gateway.

    a. Navegue até Autorizadores da sua API e anote o nome do autorizador criado pelas Permissões Verificadas.
    b. Navegue até Recursos e selecione um método de nível superior na sua API.
    c. Selecione Editar em Configurações de solicitação de método.
    d. Defina o Autorizador como o nome do autorizador que você anotou anteriormente.
    e. Expanda os cabeçalhos da solicitação HTTP, insira um Nome ou `AUTHORIZATION` e selecione Obrigatório.
    f. Implante o estágio da API.
    g. Salve suas alterações.
12. Teste seu autorizador com um token de grupo de usuários do tipo Token que você selecionou na etapa Escolher fonte de identidade. Para obter mais informações sobre o login do grupo de usuários e a recuperação de tokens, consulte Fluxo de autenticação do grupo de usuários no Guia do desenvolvedor do Amazon Cognito.
13. Teste a autenticação novamente com um token do grupo de usuários no `AUTHORIZATION` cabeçalho de uma solicitação para sua API.
14. Examine seu novo repositório de políticas. Adicione e refine políticas.

Sample policy store

Para criar um armazenamento de políticas usando o método de configuração Exemplo de armazenamento de políticas

1. Na seção Opções iniciais, escolha Exemplo de armazenamento de políticas.
2. Na seção Projeto de exemplo, escolha o tipo de exemplo de aplicação do Verified Permissions a ser usado.

- PhotoFlashé um exemplo de aplicativo web voltado para o cliente que permite aos usuários compartilhar fotos e álbuns individuais com amigos. Os usuários podem definir permissões refinadas sobre quem tem permissão para visualizar, comentar e compartilhar novamente suas fotos. Os proprietários da conta também podem criar grupos de amigos e organizar fotos em álbuns.
- DigitalPetO Store é um exemplo de aplicativo em que qualquer pessoa pode se registrar e se tornar um cliente. Os clientes podem adicionar animais de estimação à venda, pesquisar animais de estimação e fazer pedidos. Os clientes que adicionaram um animal de estimação são registrados como donos do animal. Os donos de animais de estimação podem atualizar os detalhes do animal, fazer upload de uma imagem do animal ou excluir o anúncio do animal. Os clientes que fizeram um pedido são registrados como proprietários do pedido. Os proprietários do pedido podem obter detalhes sobre o pedido ou cancelá-lo. Os gerentes de lojas de animais de estimação têm acesso administrativo.

  > **Note**
  >
  > O DigitalPetrepositório de políticas de amostra da Store não inclui modelos de política. Os repositórios TinyTododе políticas PhotoFlashe exemplos incluem modelos de políticas.

- TinyTodoé um aplicativo de exemplo que permite aos usuários criar tarefas e listas de tarefas. Os proprietários de listas podem gerenciar e compartilhar suas listas e especificar quem pode visualizar ou editar as listas.

3. Um namespace é gerado automaticamente para o esquema do exemplo de armazenamento de políticas com base no projeto de exemplo que você escolheu.
4. Escolha Criar armazenamento de políticas.

   Seu armazenamento de políticas é criado com políticas e um esquema para o exemplo de armazenamento de políticas que você escolheu. Para obter mais informações sobre políticas vinculadas a modelos que você pode criar para os exemplos de armazenamentos de políticas, consulte. Exemplo de políticas vinculadas a modelos para exemplos de armazenamentos de políticas do Verified Permissions

Empty policy store

Para criar um armazenamento de políticas usando o método de configuração Armazenamento de políticas vazio

1. Na seção Opções iniciais, escolha Armazenamento de políticas vazio.
2. Escolha Criar armazenamento de políticas.

Um armazenamento de políticas vazio é criado sem um esquema, o que significa que as políticas não são validadas. Para obter mais informações sobre a atualização do esquema do armazenamento de políticas, consulte Esquema do armazenamento de políticas do Amazon Verified Permissions.

Para obter mais informações sobre a criação de políticas para seu armazenamento de políticas, consulte Criação de políticas estáticas do Amazon Verified Permissions e Criação de políticas vinculadas a modelos.

AWS CLI

Para criar um armazenamento de políticas vazio usando a AWS CLI.

Você pode criar um armazenamento de políticas usando a operação `create-policy-store`.

> **Note**
>
> Um repositório de políticas que você cria usando o AWS CLI está vazio.
>
> - Para adicionar um esquema, consulte Esquema do armazenamento de políticas do Amazon Verified Permissions.
> - Para adicionar políticas, consulte Criação de políticas estáticas do Amazon Verified Permissions.
> - Para adicionar modelos de políticas, consulte Criação de modelos de política.

```
$ 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

Você pode criar um armazenamento de políticas usando a API `CreatePolicyStore`. Para obter mais informações, consulte CreatePolicyArmazenar no Guia de referência da API de permissões verificadas da Amazon.

Armazenamentos de políticas vinculados à API

Ao criar um novo repositório de políticas no console Amazon Verified Permissions, você pode escolher a opção Configurar com o API Gateway e uma fonte de identidade. Com essa opção, você cria um armazenamento de políticas vinculado à API, um modelo de autorização para aplicativos que se autenticam com grupos de usuários do Amazon Cognito ou com um provedor de identidade (IdP) do OIDC e obtêm dados das APIs do Amazon API Gateway. Para começar, consulte o Crie um repositório de políticas com uma API conectada e um provedor de identidade.

Tópicos

- Como as permissões verificadas autorizam solicitações de API
- Adicionando controle de acesso baseado em atributos (ABAC)
- Considerações sobre repositórios de políticas vinculados à API
- Solução de problemas de repositórios de políticas vinculados à API

> **Important**
>
> Os repositórios de políticas que você cria com a opção Configurar com o API Gateway e uma fonte de identidade no console de Permissões Verificadas não se destinam à implantação imediata na produção. Com seu repositório de políticas inicial, finalize seu modelo de autorização e exporte os recursos do repositório de políticas para o. CloudFormation Implante permissões verificadas na produção de forma programática com o AWS Cloud Development Kit (CDK). Para ter mais informações, consulte Passando para a produção com AWS CloudFormation.

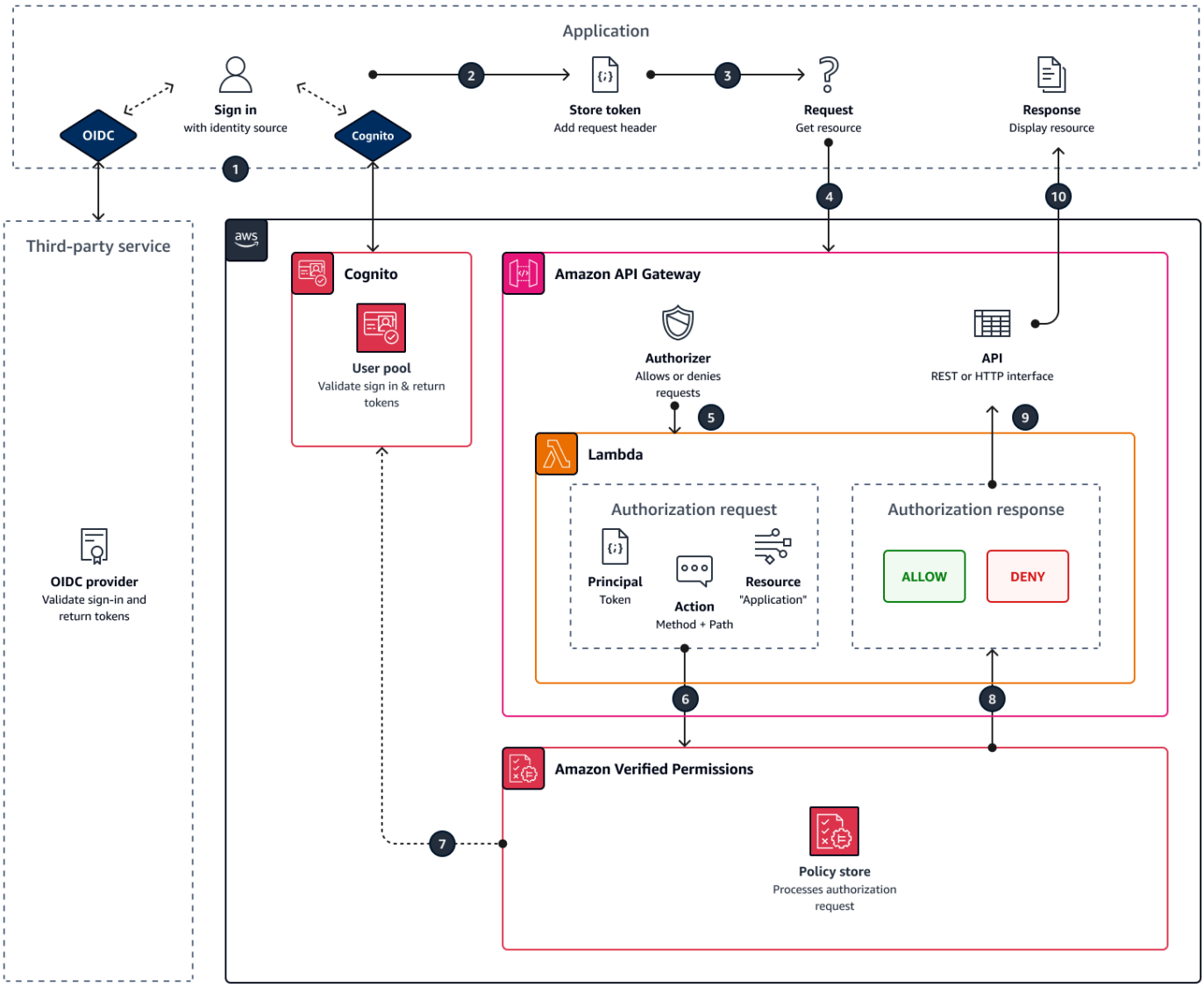
Em um repositório de políticas vinculado a uma API e a uma fonte de identidade, seu aplicativo apresenta um token de grupo de usuários em um cabeçalho de autorização ao fazer uma solicitação à API. A fonte de identidade do seu repositório de políticas fornece validação de token para permissões verificadas. O token `principal` forma as solicitações de autorização com a IsAuthorizedWithTokenAPI. As Permissões Verificadas criam políticas em torno da associação de seus usuários ao grupo, conforme apresentado em uma declaração de grupos em identidade (ID) e tokens de acesso, por exemplo`cognito:groups`, para grupos de usuários. Sua API processa o token do seu aplicativo em um autorizador Lambda e o envia à Verified Permissions para uma decisão de autorização. Quando sua API recebe a decisão de autorização do autorizador Lambda, ela passa a solicitação para sua fonte de dados ou nega a solicitação.

Componentes da fonte de identidade e autorização do API Gateway com permissões verificadas

- Um grupo de usuários do Amazon Cognito ou IdP do OIDC que autentica e agrupa usuários. Os tokens dos usuários preenchem a associação ao grupo e o principal ou contexto que as Permissões Verificadas avaliam em seu repositório de políticas.
- Uma API REST do API Gateway. As permissões verificadas definem ações de caminhos e métodos de API, por exemplo`MyAPI::Action::get /photo`.
- Uma função Lambda e um autorizador Lambda para sua API. A função Lambda recebe tokens portadores do seu grupo de usuários, solicita autorização das Permissões Verificadas e retorna uma decisão ao API Gateway. O fluxo de trabalho Configurar com o Cognito e o API Gateway cria automaticamente esse autorizador Lambda para você.
- Um repositório de políticas de permissões verificadas. A fonte de identidade do repositório de políticas é seu grupo de usuários. O esquema do repositório de políticas reflete a configuração da sua API, e as políticas vinculam os grupos de usuários às ações permitidas da API.
- Um aplicativo que autentica usuários com seu IdP e anexa tokens às solicitações da API.

Como as permissões verificadas autorizam solicitações de API

Quando você cria um novo repositório de políticas e seleciona a opção Configurar com o Cognito e o API Gateway, o Verified Permissions cria o esquema e as políticas do repositório de políticas. O esquema e as políticas refletem as ações da API e os grupos de grupos de usuários que você deseja autorizar a realizar as ações. As permissões verificadas também criam a função e o autorizador do Lambda. Você deve configurar o novo autorizador em um método na sua API.



1. Seu usuário faz login com seu aplicativo por meio do Amazon Cognito ou de outro IdP do OIDC. O IdP emite tokens de ID e acesso com as informações do usuário.
2. Seu aplicativo armazena os JWTs. Para obter mais informações, consulte Uso de tokens com grupos de usuários no Guia do Desenvolvedor do Amazon Cognito.
3. Seu usuário solicita dados que seu aplicativo deve recuperar de uma API externa.
4. Seu aplicativo solicita dados de uma API REST no API Gateway. Ele anexa um ID ou token de acesso como cabeçalho da solicitação.
5. Se sua API tiver um cache para a decisão de autorização, ela retornará a resposta anterior. Se o armazenamento em cache estiver desativado ou a API não tiver cache atual, o API Gateway passará os parâmetros da solicitação para um autorizador Lambda baseado em token.

6. A função Lambda envia uma solicitação de autorização para um repositório de políticas de permissões verificadas com a IsAuthorizedWithTokenAPI. A função Lambda transmite os elementos de uma decisão de autorização:
   a. O token do usuário como principal.
   b. O método da API combinado com o caminho da API, por exemplo`GetPhoto`, como a ação.
   c. O termo `Application` como recurso.
7. As permissões verificadas validam o token. Para obter mais informações sobre como os tokens do Amazon Cognito são validados, consulte Autorização com permissões verificadas da Amazon no Guia do desenvolvedor do Amazon Cognito.
8. O Verified Permissions avalia a solicitação de autorização em relação às políticas em seu repositório de políticas e retorna uma decisão de autorização.
9. O autorizador Lambda retorna uma `Deny` resposta `Allow` or para o API Gateway.
10. A API retorna dados ou uma `ACCESS_DENIED` resposta ao seu aplicativo. Seu aplicativo processa e exibe os resultados da solicitação da API.

Adicionando controle de acesso baseado em atributos (ABAC)

Uma sessão de autenticação típica com um IdP retorna tokens de ID e acesso. Você pode passar qualquer um desses tipos de token como um token portador em solicitações de aplicativos para sua API. Dependendo de suas escolhas ao criar seu repositório de políticas, as Permissões Verificadas esperam um dos dois tipos de tokens. Ambos os tipos contêm informações sobre a associação do usuário ao grupo. Para obter mais informações sobre os tipos de token no Amazon Cognito, consulte Uso de tokens com grupos de usuários no Guia do desenvolvedor do Amazon Cognito.

Depois de criar um repositório de políticas, você pode adicionar e estender políticas. Por exemplo, você pode adicionar novos grupos às suas políticas ao adicioná-los ao seu grupo de usuários. Como seu repositório de políticas já está ciente da forma como seu grupo de usuários apresenta grupos em tokens, você pode permitir um conjunto de ações para qualquer novo grupo com uma nova política.

Talvez você também queira estender o modelo de avaliação de políticas baseado em grupos para um modelo mais preciso baseado nas propriedades do usuário. Os tokens do grupo de usuários contêm informações adicionais do usuário que podem contribuir para as decisões de autorização.

Tokens de identificação

Os tokens de ID representam os atributos do usuário e têm o nível mais alto de controle de acesso refinado. Para avaliar endereços de e-mail, números de telefone ou atributos personalizados, como departamento e gerente, avalie o token de ID.

Tokens de acesso

Os tokens de acesso representam as permissões de um usuário com escopos do OAuth 2.0. Para adicionar uma camada de autorização ou configurar solicitações de recursos adicionais, avalie o token de acesso. Por exemplo, você pode validar se um usuário está nos grupos apropriados e tem um escopo como `PetStore.read` esse que geralmente autoriza o acesso à API. Os grupos de usuários podem adicionar escopos personalizados aos tokens com servidores de recursos e com a personalização de tokens em tempo de execução.

Veja, Trabalhando com fontes de identidade em esquemas e políticas por exemplo, políticas que processam reivindicações em tokens de ID e acesso.

Considerações sobre repositórios de políticas vinculados à API

Ao criar um repositório de políticas vinculado à API no console de permissões verificadas, você está criando um teste para uma eventual implantação de produção. Antes de passar para a produção, estabeleça uma configuração fixa para sua API e seu grupo de usuários. Considere os seguintes fatores:

O API Gateway armazena respostas em cache

Em repositórios de políticas vinculados à API, o Verified Permissions cria um autorizador Lambda com um TTL de cache de autorização de 120 segundos. Você pode ajustar esse valor ou desativar o armazenamento em cache no seu autorizador. Em um autorizador com o armazenamento em cache ativado, seu autorizador retorna a mesma resposta todas as vezes até que o TTL expire. Isso pode estender a vida útil efetiva dos tokens do grupo de usuários em uma duração igual ao TTL de cache do estágio solicitado.

Os grupos do Amazon Cognito podem ser reutilizados

As permissões verificadas da Amazon determinam a associação ao grupo de usuários do grupo de usuários a partir da `cognito:groups` reivindicação no ID do usuário ou no token de acesso. O valor dessa afirmação é uma matriz dos nomes amigáveis dos grupos de grupos de usuários aos quais o usuário pertence. Você não pode associar grupos de grupos de usuários a um identificador exclusivo.

Grupos de grupos de usuários que você exclui e recria com o mesmo nome presente no seu repositório de políticas como o mesmo grupo. Ao excluir um grupo de um grupo de usuários, exclua todas as referências ao grupo do seu repositório de políticas.

O namespace e o esquema derivados da API são point-in-time

O Verified Permissions captura sua API em um determinado momento: ele só consulta sua API quando você cria seu repositório de políticas. Quando o esquema ou o nome da sua API muda, você deve atualizar seu repositório de políticas e o autorizador Lambda ou criar um novo armazenamento de políticas vinculado à API. As permissões verificadas derivam o namespace do repositório de políticas do nome da sua API.

A função Lambda não tem configuração de VPC

A função Lambda que a Verified Permissions cria para seu autorizador de API não está conectada a uma VPC. Por padrão. As APIs que têm acesso à rede restrito a VPCs privadas não podem se comunicar com a função Lambda que autoriza solicitações de acesso com permissões verificadas.

A Verified Permissions implanta recursos do autorizador em CloudFormation

Para criar um repositório de políticas vinculado à API, você deve cadastrar um AWS principal altamente privilegiado no console de Permissões Verificadas. Esse usuário implanta uma AWS CloudFormation pilha que cria recursos em vários. Serviços da AWS Esse diretor deve ter a permissão para adicionar e modificar recursos em Permissões Verificadas IAM, Lambda e API Gateway. Como prática recomendada, não compartilhe essas credenciais com outros administradores em sua organização.

Consulte Passando para a produção com AWS CloudFormation para obter uma visão geral dos recursos criados pelas Permissões Verificadas.

Passando para a produção com AWS CloudFormation

Os armazenamentos de políticas vinculados à API são uma forma de criar rapidamente um modelo de autorização para uma API do API Gateway. Eles foram projetados para servir como um ambiente de teste para o componente de autorização do seu aplicativo. Depois de criar seu repositório de políticas de teste, passe algum tempo refinando as políticas, o esquema e o autorizador Lambda.

Você pode ajustar a arquitetura da sua API, exigindo ajustes equivalentes no esquema e nas políticas do repositório de políticas. Os repositórios de políticas vinculados à API não atualizam automaticamente seu esquema a partir da arquitetura da API. As Permissões Verificadas somente

pesquisam a API no momento em que você cria um repositório de políticas. Se sua API mudar o suficiente, talvez você precise repetir o processo com um novo repositório de políticas.

Quando seu aplicativo e modelo de autorização estiverem prontos para implantação na produção, integre o repositório de políticas vinculado à API que você desenvolveu com seus processos de automação. Como prática recomendada, recomendamos que você exporte o esquema e as políticas do repositório de políticas em um AWS CloudFormation modelo que possa ser implantado em outros Contas da AWS e. Regiões da AWS

Os resultados do processo de armazenamento de políticas vinculado à API são um armazenamento de políticas inicial e um autorizador Lambda. O autorizador Lambda tem vários recursos dependentes. O Verified Permissions implanta esses recursos em uma pilha gerada automaticamente CloudFormation . Para implantar na produção, você deve coletar o repositório de políticas e os recursos do autorizador Lambda em um modelo. Um repositório de políticas vinculado à API é composto pelos seguintes recursos:

1. AWS::VerifiedPermissions::PolicyStore: copie seu esquema para o `SchemaDefinition` objeto. Escape de " personagens como\".
2. AWS::VerifiedPermissions::IdentitySource: copie os valores da saída do seu repositório GetIdentitySource de políticas de teste e modifique-os conforme necessário.
3. Uma ou mais das AWS::VerifiedPermissions::Policyseguintes: Copie sua declaração de política para o `Definition` objeto. Escape de " personagens como\".
4. AWS: :Lambda: :Function, AWS::: :Role,IAM:AWS:: :Policy,IAM::: :Authorizer,AWS ApiGateway AWS::Lambda::Permission: Copie o modelo da guia Modelo da pilha que as Permissões Verificadas implantaram quando você criou seu armazenamento de políticas.

O modelo a seguir é um exemplo de armazenamento de políticas. Você pode acrescentar os recursos do autorizador Lambda da sua pilha existente a esse modelo.

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

Solução de problemas de repositórios de políticas vinculados à API

Use as informações aqui para ajudá-lo a diagnosticar e corrigir problemas comuns ao criar repositórios de políticas vinculados à API Amazon Verified Permissions.

Tópicos

- Eu atualizei minha política, mas a decisão de autorização não mudou
- Anexei o autorizador Lambda à minha API, mas ele não está gerando solicitações de autorização
- Recebi uma decisão de autorização inesperada e quero revisar a lógica de autorização
- Quero encontrar registros do meu autorizador Lambda
- Meu autorizador Lambda não existe
- Minha API está em uma VPC privada e não consigo invocar o autorizador
- Quero processar atributos de usuário adicionais no meu modelo de autorização
- Quero adicionar novas ações, atributos de contexto de ação ou atributos de recursos

Eu atualizei minha política, mas a decisão de autorização não mudou

Por padrão, o Verified Permissions configura o autorizador Lambda para armazenar em cache as decisões de autorização por 120 segundos. Tente novamente após dois minutos ou desative o cache no seu autorizador. Para obter mais informações, consulte Como ativar o cache da API para melhorar a capacidade de resposta no Guia do desenvolvedor do Amazon API Gateway.

Anexei o autorizador Lambda à minha API, mas ele não está gerando solicitações de autorização

Para começar a processar as solicitações, você deve implantar o estágio da API ao qual você anexou seu autorizador. Para obter mais informações, consulte Implantação de uma API REST no Guia do desenvolvedor do Amazon API Gateway.

Recebi uma decisão de autorização inesperada e quero revisar a lógica de autorização

O processo de armazenamento de políticas vinculado à API cria uma função Lambda para seu autorizador. As permissões verificadas incorporam automaticamente a lógica de suas decisões de autorização na função do autorizador. Você pode voltar depois de criar seu repositório de políticas para revisar e atualizar a lógica na função.

Para localizar sua função Lambda no AWS CloudFormation console, escolha o botão Verificar implantação na página Visão geral do seu novo repositório de políticas.

Você também pode localizar sua função no AWS Lambda console. Navegue até o console no seu repositório Região da AWS de políticas e pesquise um nome de função com o prefixo de`AVPAuthorizerLambda`. Se você criou mais de um repositório de políticas vinculado à API, use o horário da última modificação de suas funções para correlacioná-las com a criação do repositório de políticas.

Quero encontrar registros do meu autorizador Lambda

As funções Lambda coletam métricas e registram seus resultados de invocação na Amazon. CloudWatch Para revisar seus registros, localize sua função no console Lambda e escolha a guia Monitor. Selecione Exibir CloudWatch registros e revise as entradas no grupo de registros.

Para obter mais informações sobre os registros de funções do Lambda, consulte Como usar o Amazon CloudWatch Logs com AWS Lambda o Guia do AWS Lambda Desenvolvedor.

Meu autorizador Lambda não existe

Depois de concluir a configuração de um repositório de políticas vinculado à API, você deve anexar o autorizador Lambda à sua API. Se você não conseguir localizar seu autorizador no console do API Gateway, os recursos adicionais do seu repositório de políticas podem ter falhado ou ainda não terem sido implantados. Os repositórios de políticas vinculados à API implantam esses recursos em uma AWS CloudFormation pilha.

Permissões verificadas exibe um link com o rótulo Verificar implantação no final do processo de criação. Se você já saiu dessa tela, acesse o CloudFormation console e pesquise nas pilhas recentes um nome com o prefixo. `AVPAuthorizer-<policy store ID>` CloudFormation fornece informações valiosas sobre solução de problemas na saída de uma implantação de pilha.

Para obter ajuda na solução de problemas de CloudFormation pilhas, consulte Solução de problemas CloudFormation no Guia AWS CloudFormation do usuário.

Minha API está em uma VPC privada e não consigo invocar o autorizador

As permissões verificadas não oferecem suporte ao acesso aos autorizadores Lambda por meio de VPC endpoints. Você deve abrir um caminho de rede entre sua API e a função Lambda que serve como seu autorizador.

Quero processar atributos de usuário adicionais no meu modelo de autorização

O processo de armazenamento de políticas vinculado à API deriva as políticas de Permissões Verificadas da reivindicação dos grupos nos tokens dos usuários. Para atualizar seu modelo de autorização para considerar atributos adicionais do usuário, integre esses atributos às suas políticas.

Você pode mapear muitas reivindicações em tokens de ID e acesso dos grupos de usuários do Amazon Cognito para declarações de políticas de permissões verificadas. Por exemplo, a maioria dos usuários tem uma `email` reivindicação em seu token de ID. Para obter mais informações sobre como adicionar declarações de sua fonte de identidade às políticas, consulteTrabalhando com fontes de identidade em esquemas e políticas.

Quero adicionar novas ações, atributos de contexto de ação ou atributos de recursos

Um repositório de políticas vinculado à API e o autorizador Lambda que ele cria são um recurso. point-in-time Eles refletem o estado da sua API no momento da criação. O esquema do repositório de políticas não atribui nenhum atributo de contexto às ações, nem nenhum atributo ou pai ao `Application` recurso padrão.

Ao adicionar ações — caminhos e métodos — à sua API, você deve atualizar seu repositório de políticas para estar ciente das novas ações. Você também deve atualizar seu autorizador Lambda para processar solicitações de autorização para as novas ações. Você pode começar de novo com um novo repositório de políticas ou atualizar seu repositório de políticas existente.

Para atualizar seu repositório de políticas existente, localize sua função. Examine a lógica na função gerada automaticamente e atualize-a para processar as novas ações, atributos ou contexto. Em seguida, edite seu esquema para incluir as novas ações e atributos.

Alternância de armazenamentos de políticas do Verified Permissions

AWS Management Console

Para alternar entre armazenamentos de políticas ou criar armazenamentos de políticas adicionais

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha alternar ao lado do Armazenamento de políticas atuais.
3. Para alternar entre os armazenamentos de políticas existentes ou criar armazenamentos de políticas adicionais.

   - Para alternar entre os armazenamentos de políticas, escolha o ID do armazenamento de políticas para o qual você alternará.
   - Para criar um novo armazenamento de políticas, escolha Criar novo armazenamento de políticas. Siga as instruções em Criação de armazenamentos de políticas do Verified Permissions.

AWS CLI

Para alternar entre armazenamentos de políticas ou criar armazenamentos de políticas adicionais

A AWS CLI não mantém um armazenamento de políticas “padrão”. Em vez disso, a maioria dos comandos da AWS CLI usa `--policy-store-id` para especificar qual armazenamento de políticas será usado para cada comando.

Para criar um novo repositório de políticas, use o create-policy-storecomando.

Exclusão de armazenamentos de políticas do Verified Permissions

AWS Management Console

Para excluir um armazenamento de políticas

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Configurações.
3. Escolha Excluir este armazenamento de políticas.
4. Digite `delete` na caixa de texto e escolha Excluir.

AWS CLI

Para excluir um armazenamento de políticas

Você pode excluir um armazenamento de políticas usando a operação `delete-policy-store`.

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

Se for bem-sucedido, esse comando não gerará nenhum resultado.

Esquema do armazenamento de políticas do Amazon Verified Permissions

O esquema é uma declaração da estrutura dos tipos de entidade compatíveis com sua aplicação e das ações que a aplicação pode fornecer nas solicitações de autorização.

Para obter mais informações, consulte Formato do esquema do Cedar no Guia de referência da linguagem de política Cedar.

> **Note**
>
> O uso de esquemas no Verified Permissions é opcional, mas eles são altamente recomendados para software de produção. Quando você cria uma nova política, o Verified Permissions pode usar o esquema para validar as entidades e os atributos referenciados no escopo e nas condições, a fim de evitar erros de digitação e nas políticas que possam resultar em um comportamento confuso do sistema. Se você ativar a validação da política, todas as novas políticas deverão estar em conformidade com o esquema.

AWS Management Console

Para criar um esquema

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Esquema.
3. Selecione Create schema (Criar esquema).

AWS CLI

Para enviar um novo esquema ou substituir um esquema existente por meio do AWS CLI.

Você pode criar um repositório de políticas executando um AWS CLI comando semelhante ao exemplo a seguir.

Considere um esquema que contenha o seguinte conteúdo do Cedar:

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

Você deve primeiro inserir o JSON em uma string de linha única e prefaciá-lo com uma declaração do seu tipo de dados: `cedarJson`. O exemplo a seguir usa o seguinte conteúdo do arquivo `schema.json` que contém a versão de escape do esquema JSON.

> **Note**
>
> O exemplo aqui apresenta quebra de linha para facilitar a leitura. O arquivo inteiro deve aparecer em uma única linha para que o comando o aceite.

```
{"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

Você pode criar um armazenamento de políticas usando a API `PutSchema`. Para obter mais informações, consulte o Guia PutSchemade referência da API de permissões verificadas da Amazon.

Edição de esquemas no modo Visual

Quando você seleciona Esquema no console de Permissões verificadas, o modo Visual exibe os tipos de entidade e as ações que compõem seu esquema. Nessa visualização de nível superior ou a partir dos detalhes de qualquer entidade, você pode escolher Editar esquema para começar a fazer atualizações em seu esquema. O modo visual não está disponível em alguns formatos de esquema, como registros aninhados.

O editor de esquema visual começa com uma série de diagramas que ilustram as relações entre as entidades em seu esquema. Escolha Expandir para maximizar sua visualização das relações entre entidades do seu esquema.

Diagrama de ações

A exibição do diagrama de ações lista os tipos de diretores que você configurou em seu repositório de políticas, as ações que eles estão qualificados para executar e os recursos nos quais eles estão qualificados para realizar ações. As linhas entre entidades indicam sua capacidade de criar uma política que permita ao diretor realizar uma ação em um recurso. Se seu diagrama de ações não indicar uma relação entre duas entidades, você deverá criar essa relação entre elas antes de permitir ou negá-la nas políticas. Selecione uma entidade para ver uma visão geral das propriedades e faça uma busca detalhada para ver todos os detalhes. Escolha Filtrar por essa [ação | tipo de recurso | tipo principal] para ver uma entidade em uma exibição com somente suas próprias conexões.

Diagrama de tipos de entidade

O diagrama de tipos de entidades se concentra nas relações entre diretores e recursos. Quando quiser entender as complexas relações parentais aninhadas em seu esquema, revise este diagrama. Passe o mouse sobre uma entidade para detalhar os relacionamentos principais que ela tem.

Abaixo dos diagramas, há visualizações de lista dos tipos de entidades e ações em seu esquema. A exibição em lista é útil quando você deseja visualizar imediatamente os detalhes de uma ação específica ou tipo de entidade. Selecione qualquer entidade para ver os detalhes.

Edite um esquema do Verified Permissions no modo visual

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Esquema.
3. Escolha Modo visual. Revise os diagramas de relacionamento entre entidades e planeje as alterações que você deseja fazer em seu esquema. Opcionalmente, você pode filtrar por uma entidade para examinar suas conexões individuais com outras entidades.
4. Escolha Edit schema (Editar esquema).
5. Na seção Detalhes, digite um Namespace para seu esquema.
6. Na seção Tipos de entidade, escolha Adicionar novo tipo de entidade.
7. Digite o nome da entidade.
8. (Opcional) Escolha Adicionar um pai para adicionar entidades pai das quais a nova entidade é membro. Para remover um pai adicionado à entidade, escolha Remover ao lado do nome do pai.
9. Para adicionar outro atributo à entidade, escolha Adicionar um atributo. Digite o Nome do atributo e escolha um Tipo de atributo para cada atributo da entidade. O Verified Permissions usa os valores de atributo especificados ao verificar as políticas com base no esquema. Especifique se cada atributo é Obrigatório. Para remover um atributo adicionado à entidade, escolha Remover ao lado do atributo.
10. Escolha Adicionar tipo de entidade para adicionar a entidade ao esquema.
11. Na seção Ações, escolha Adicionar nova ação.
12. Digite o nome da ação.
13. (Opcional) Escolha Adicionar um recurso para adicionar tipos de recursos aos quais a ação se aplica. Para remover um tipo de recurso adicionado à ação, escolha Remover ao lado do nome do tipo de recurso.

14. (Opcional) Escolha Adicionar uma entidade principal para adicionar um tipo de entidade principal ao qual a ação se aplica. Para remover um tipo de entidade principal adicionado à ação, escolha Remover ao lado do nome do tipo de entidade principal.
15. Escolha Adicionar um atributo para adicionar atributos que podem ser adicionados ao contexto de uma ação em suas solicitações de autorização. Insira o nome do atributo e escolha o tipo de atributo para cada atributo. O Verified Permissions usa os valores de atributo especificados ao verificar as políticas com base no esquema. Especifique se cada atributo é Obrigatório. Para remover um atributo adicionado à ação, escolha Remover ao lado do atributo.
16. Selecione Adicionar ação.
17. Depois que todos os tipos de entidade e ações tiverem sido adicionados ao esquema, escolha Salvar alterações.

Edição de esquemas no modo JSON

Para editar um esquema do Verified Permissions no modo JSON

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Esquema.
3. Escolha o Modo JSON e, em seguida, escolha Editar esquema.
4. Insira o conteúdo do esquema JSON no campo Conteúdo. Você só poderá salvar atualizações em seu esquema depois que resolver todos os erros de sintaxe. Você pode escolher Formatar para formatar a sintaxe JSON do seu esquema com o espaçamento e o recuo recomendados.
5. Escolha Salvar alterações.

Excluir um esquema

AWS Management Console

Para excluir um esquema do Verified Permissions

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Esquema.
3. Escolha Excluir esquema.

AWS CLI

Para excluir um esquema do Verified Permissions

Não existe um comando delete schema. Você pode excluir o esquema em um armazenamento de políticas usando o comando `put-schema` com um esquema vazio no campo `cedarJson`. Um esquema vazio é representado por um par de chaves '{}'.

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

Modo de validação de política do Amazon Verified Permissions

Você pode definir o modo de validação de política no Verified Permissions para determinar se as alterações da política são validadas com base no esquema do armazenamento de políticas.

> **Important**
>
> Quando você ativa a validação de política, todas as tentativas de criação ou atualização de uma política ou modelo de política são validadas com base no esquema do armazenamento de políticas. O Verified Permissions rejeitará a solicitação se a validação falhar.

AWS Management Console

Para definir o modo de validação de política para um armazenamento de políticas

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. Escolha Configurações.
3. Na seção Modo de validação da política, escolha Modificar.
4. Execute um destes procedimentos:
   - Para ativar a validação da política e garantir que todas as alterações de política sejam validadas com base no seu esquema, escolha o botão de opção Estrito (recomendado).
   - Para desativar a validação de alterações de política, escolha o botão de opção Desativado. Digite `confirm` para confirmar que as atualizações das políticas não serão mais validadas com base no seu esquema.
5. Escolha Salvar alterações.

AWS CLI

Para definir o modo de validação de política de um armazenamento de políticas

Você pode alterar o modo de validação de um repositório de políticas usando a UpdatePolicyStoreoperação e especificando um valor diferente para o ValidationSettingsparâmetro.

```
$ aws verifiedpermissions update-policy-store \
    --validation-settings "mode=OFF",
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-17T18:36:10.134448+00:00",
    "lastUpdatedDate": "2023-05-17T18:36:10.134448+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "validationSettings": {
        "Mode": "OFF"
    }
}
```

Para obter mais informações, consulte Validação de política no Guia de referência da linguagem de política Cedar.

Políticas do Amazon Verified Permissions

Uma política é uma declaração que permite ou proíbe uma entidade principal de realizar uma ou mais ações em um recurso. Cada política é avaliada independentemente de qualquer outra política. Para obter mais informações sobre como as políticas do Cedar são estruturadas e avaliadas, consulte Validação da política do Cedar com base no esquema no Guia de referência da linguagem de política Cedar.

Important

Ao escrever políticas do Cedar que fazem referência a entidades principais, recursos e ações, você pode definir os identificadores exclusivos usados para cada um desses elementos. Convém seguir estas práticas recomendadas:

- Use valores como identificadores universalmente exclusivos (UUIDs) para todos os identificadores de identidades principais e de recursos.

  Por exemplo, se o usuário `jane` for desligado da empresa e você permitir que outra pessoa use o nome `jane`, esse novo usuário terá acesso automaticamente a tudo o que é concedido pelas políticas que ainda fazem referência a `User::"jane"`. O Cedar não consegue fazer a distinção entre o novo usuário e o antigo. Essa orientação se aplica tanto aos identificadores de entidades principais quanto aos identificadores de recursos. Sempre use identificadores que sejam comprovadamente exclusivos e nunca sejam reutilizados para garantir que você não conceda acesso involuntariamente devido à presença de um identificador antigo em uma política.

  Quando você usa um UUID para uma entidade, recomendamos que você o siga com o especificador de comentário // e o nome “amigável” da sua entidade. Isso torna as políticas mais fáceis de entender. Por exemplo: principal == User::"a1b2c3d4-e5f6-a1b2-c3d4-EXAMPLE11111", // alice
- Não inclua informações pessoais, confidenciais ou sigilosas como parte do identificador exclusivo de suas entidades principais ou recursos. Esses identificadores são incluídos nas entradas de registro compartilhadas nas AWS CloudTrail trilhas.

Formatação de entidades no Amazon Verified Permissions

O Amazon Verified Permissions usa a linguagem de política Cedar para criar políticas. A sintaxe das políticas e os tipos de dados compatíveis correspondem à sintaxe e aos tipos de dados descritos nos tópicos Construção de políticas básicas no Cedar e Tipos de dados compatíveis no Cedar no Guia de referência da linguagem de política Cedar. No entanto, existem diferenças entre o Verified Permissions e o Cedar na formatação de entidades ao fazer uma solicitação de autorização.

A formatação JSON de entidades no Verified Permissions difere da mesma formatação no Cedar nos seguintes aspectos:

- No Verified Permissions, um objeto JSON deve ter todos os seus pares de chave-valor encapsulados em um objeto JSON com o nome `Record`.
- Uma lista JSON no Verified Permissions deve ser encapsulado em um par de chave-valor JSON em que o nome da chave é `Set` e o valor é a lista JSON original do Cedar.
- Para os nomes de tipo `String`, `Long` e `Boolean`, cada par de chave-valor do Cedar é substituído por um objeto JSON no Verified Permissions. O nome do objeto é o nome da chave original. No objeto JSON, há um par de chave-valor em que o nome da chave é o nome de tipo do valor escalar (`String`, `Long` ou `Boolean`) e o valor é o valor da entidade do Cedar.
- A formatação de sintaxe das entidades do Cedar e das entidades do Verified Permissions difere nos seguintes aspectos:

Formato do Cedar	Formato do Verified Permissions
`uid`	`Identifier`
`type`	`EntityType`
`id`	`EntityId`
`attrs`	`Attributes`
`parents`	`Parents`

O exemplo a seguir mostra como as entidades em uma lista são formatadas por meio do Cedar.

```
[
```

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

O exemplo a seguir mostra como as mesmas entidades do exemplo anterior na lista do Cedar são formatadas no Verified Permissions.

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

O exemplo a seguir mostra como as entidades do Cedar são formatadas para avaliar uma política em uma solicitação de autorização.

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

O exemplo a seguir mostra como as mesmas entidades do exemplo anterior do Cedar são formatadas no 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": []
    }
]
```

Criação de políticas estáticas do Amazon Verified Permissions

Você pode criar uma política estática do Cedar para permitir ou negar que as entidades principais executem ações especificadas em recursos especificados da sua aplicação.

AWS Management Console

Para criar uma política estática

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. Selecione Criar política e escolha Criar política estática.
4. Na seção Efeito da política, escolha se a política permitirá ou proibirá quando uma solicitação corresponder à política.
5. No campo Escopo das entidades principais, escolha o escopo das entidades principais aos quais a política se aplicará.
   - Escolha Entidade principal específica para aplicar a política a uma entidade principal específica. Especifique o tipo de entidade e o identificador da entidade principal que receberá a permissão ou a proibição de realizar as ações especificadas na política.
   - Escolha Grupo de entidades principais para aplicar a política a um grupo de entidades principais. Digite o nome do grupo de entidades principais no campo Grupo de entidades principais.
   - Escolha Todas as entidades principais para aplicar a política a todas as entidades principais do armazenamento de políticas.
6. No campo Escopo dos recursos, escolha o escopo dos recursos aos quais a política será aplicada.
   - Escolha Recursos específicos para aplicar a política a um recurso específico. Especifique o tipo de entidade e o identificador do recurso ao qual a política será aplicada.
   - Escolha Grupo de recursos para aplicar a política a um grupo de recursos. Digite o nome do grupo de recursos no campo Grupo de recursos.
   - Escolha Todos os recursos para aplicar a política a todos os recursos do armazenamento de políticas.
7. Na seção Escopo das ações, escolha o escopo dos recursos aos quais a política será aplicada.
   - Escolha Conjunto específico de ações para aplicar a política a um conjunto de ações. Marque as caixas de seleção ao lado das ações às quais a política será aplicada.

- Escolha Todas as ações para aplicar a política a todas as ações do armazenamento de políticas.

8. Escolha Próximo.
9. Na seção Política, revise a política do Cedar. Você pode escolher Formatar para formatar a sintaxe da política com o espaçamento e o recuo recomendados. Para obter mais informações, consulte Construção de políticas básicas no Cedar no Guia de referência da linguagem de política Cedar.
10. Na seção Detalhes, digite uma descrição opcional para a política.
11. Escolha Criar política.

AWS CLI

Para criar uma política estática

Você pode criar uma política estática usando a CreatePolicyoperação. O exemplo a seguir cria uma política estática simples.

```
$ 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"
}
```

Edição de políticas estáticas do Amazon Verified Permissions

Você pode editar uma política estática do Cedar em seu armazenamento de políticas. Você pode atualizar diretamente somente políticas estáticas. Você pode alterar somente alguns elementos de uma política estática:

- O `action` referenciada pela política.
- Uma cláusula de condição, como `when` e `unless`.

Você pode alterar somente estes elementos de uma política estática:

- Transformar uma política estática em política vinculada a modelo.
- Alterar o efeito de uma política estática de `permit` ou `forbid`.
- O `principal` referenciado por uma política estática.
- O `resource` referenciado por uma política estática.

Para alterar uma política vinculada a modelo, você deve atualizar o modelo. Para ter mais informações, consulte Edição de modelos de política.

AWS Management Console

Para editar uma política estática

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. Escolha o botão de opção ao lado da política estática a ser editada e, em seguida, escolha Editar.
4. Na seção Corpo da política, atualize a `action` ou a cláusula de condição da política estática. Você não pode atualizar o efeito da política, a `principal` ou o `resource` da política.
5. Escolha Atualizar política.

> **Note**
>
> Se a validação da política estiver habilitada no armazenamento de políticas, a atualização de uma política estática fará com que o Verified Permissions valide a política com base no esquema no armazenamento de políticas. Se a política estática atualizada não passar pela validação, a operação falhará e a atualização não será salva.

AWS CLI

Para editar uma política estática

Você pode editar uma política estática usando a UpdatePolicyoperação. O exemplo a seguir edita uma política estática simples.

O exemplo usa o arquivo `definition.txt` para conter a definição da política.

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

O comando a seguir faz referência a esse arquivo.

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

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

Visualização de políticas

AWS Management Console

Para visualizar as políticas do Verified Permissions

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Policies (Políticas). Todas as políticas que você criou são exibidas.
3. Escolha a caixa de texto Pesquisar para filtrar as políticas por Entidade principal ou Recurso.
4. Escolha o botão de opção ao lado de um modelo de política para exibir os detalhes sobre a política, por exemplo, quando a política foi criada, atualizada e o conteúdo da política.
5. Você pode excluir uma política escolhendo o botão de opção ao lado de uma política e, em seguida, escolhendo Excluir. Escolha Excluir política para confirmar a exclusão da política.

AWS CLI

Para listar todas as políticas disponíveis em um armazenamento de políticas

Você pode ver a lista de políticas usando a GetPolicyoperação. O exemplo a seguir recupera uma lista contendo uma política estática e uma política vinculada a modelo.

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

```
            "resource": {
                "entityId": "publicFolder",
                "entityType": "Album"
            }
        },
        {
            "createdDate": "2023-05-22T18:57:53.298278+00:00",
            "definition": {
                "templateLinked": {
                    "policyTemplateId": "PTEXAMPLEabcdefg111111"
                }
            },
            "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",
            "policyId": "TPEXAMPLEabcdefg111111",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "TEMPLATELINKED",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
            },
            "resource": {
                "entityId": "VacationPhoto94.jpg",
                "entityType": "Photo"
            }
        }
    ]
}
```

Para visualizar os detalhes de uma política individual

Você pode recuperar os detalhes de uma política usando a GetPolicyoperação. O exemplo a seguir recupera detalhes de uma política vinculada a modelo.

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

{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEabcdefg111111/
TPEXAMPLEabcdefg111111",
    "createdDate": "2023-03-15T16:03:07.620867Z",
    "lastUpdatedDate": "2023-03-15T16:03:07.620867Z",
    "policyDefinition": {
```

```
        "templatedPolicy": {
            "policyTemplateId": "PTEXAMPLEabcdefg111111",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
            },
            "resource": {
                "entityId": "Vacation94.jpg",
                "entityType": "Photo"
            }
        }
    },
    "policyId": "TPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "TEMPLATELINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
    },
    "resource": {
        "entityId": "Vacation94.jpg",
        "entityType": "Photo"
    }
}
```

Exemplos de políticas do Amazon Verified Permissions

Os exemplos de políticas de permissões verificadas a seguir são baseados no esquema definido para o aplicativo hipotético chamado PhotoFlash descrito na seção Exemplo de esquema do Guia de referência da linguagem de políticas Cedar. Para obter mais informações sobre a sintaxe de política do Cedar, consulte Construção de políticas básicas no Cedar no Guia de referência da linguagem de política Cedar.

Exemplos de políticas

- Permite o acesso a entidades individuais
- Permite o acesso a grupos de entidades
- Permite o acesso a qualquer entidade
- Permite o acesso aos atributos de uma entidade (ABAC)
- Nega o acesso

Permite o acesso a entidades individuais

Este exemplo mostra como criar uma política que permita à usuária `alice` visualizar a foto `VacationPhoto94.jpg`.

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

Permite o acesso a grupos de entidades

Este exemplo mostra como criar uma política que permita a qualquer membro do grupo `alice_friends` visualizar a foto `VacationPhoto94.jpg`.

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

Este exemplo mostra como criar uma política que permita à usuária `alice` visualizar qualquer foto no álbum `alice_vacation`.

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

Este exemplo mostra como criar uma política que permita à usuária `alice` visualizar, editar ou excluir qualquer foto no álbum `alice_vacation`.

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

Este exemplo mostra como criar uma política que conceda permissões à usuária `alice` no álbum `alice_vacation`, em que `admin` é um grupo definido na hierarquia de esquemas que contém as permissões para visualizar, editar e excluir uma foto.

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

Este exemplo mostra como criar uma política que conceda permissões à usuária `alice` no álbum `alice_vacation`, em que `viewer` é um grupo definido na hierarquia de esquemas que contém as permissões para visualizar e comentar uma foto. A usuária `alice` também recebe a permissão `edit` pela segunda ação listada na política.

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

Permite o acesso a qualquer entidade

Este exemplo mostra como criar uma política que permita que qualquer entidade principal autenticada visualize o álbum `alice_vacation`.

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

Este exemplo mostra como criar uma política que permita à usuária `alice` listar todos os álbuns na conta de `jane`, listar as fotos de todos os álbuns e visualizar fotos na conta.

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

```
);
```

Este exemplo mostra como criar uma política que permita à usuária `alice` executar qualquer ação nos recursos do álbum `jane_vaction`.

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

Permite o acesso aos atributos de uma entidade (ABAC)

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos. O Verified Permissions permite que sejam anexados atributos a entidades principais, ações e recursos. Esses atributos podem, então, ser referenciados nas cláusulas `when` e `unless` das políticas que avaliam os atributos das entidades principais, das ações e dos recursos que compõem o contexto da solicitação.

Os exemplos a seguir usam os atributos definidos no aplicativo hipotético chamado PhotoFlash descrito na seção Exemplo de esquema do Guia de referência da linguagem de políticas Cedar.

Este exemplo mostra como criar uma política que permita a qualquer entidade principal do departamento `HardwareEngineering` com um nível de trabalho maior ou igual a 5 visualizar e listar fotos no álbum `device_prototypes`.

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

Este exemplo mostra como criar uma política que permita à usuária `alice` visualizar qualquer recurso do tipo de arquivo `JPEG`.

```
permit(
```

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

As ações têm atributos de contexto. Você deve passar esses atributos no `context` de uma solicitação de autorização. Este exemplo mostra como você pode criar uma política que permita `alice` ao usuário realizar qualquer `readOnly` ação. Você também pode definir uma `appliesTo` propriedade para ações em seu esquema. Isso especifica ações válidas para um recurso quando você deseja garantir que, por exemplo, os usuários só possam tentar autorizar `ViewPhoto` um recurso do tipo. `PhotoFlash::Photo`

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

A melhor maneira de definir as propriedades das ações em seu esquema, no entanto, é organizá-las em grupos de ações funcionais. Por exemplo, você pode criar uma ação chamada `ReadOnlyPhotoAccess` e configurada `PhotoFlash::Action::"ViewPhoto"` para ser membro de `ReadOnlyPhotoAccess` um grupo de ação. Este exemplo mostra como você pode criar uma política que conceda a Alice acesso às ações somente para leitura nesse grupo.

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

Este exemplo mostra como criar uma política que permita a todas as entidades principais executar qualquer ação nos recursos para os quais elas tenham o atributo `owner`.

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

Este exemplo mostra como criar uma política que permita a qualquer entidade principal visualizar qualquer recurso se o atributo `department` da entidade principal corresponder ao atributo `department` do recurso.

> **Note**
>
> Se uma entidade não tiver um atributo mencionado em uma condição de política, a política será ignorada quando uma decisão de autorização for tomada e a avaliação dessa política falhar para essa entidade. Por exemplo, qualquer entidade principal que não tenha um atributo `department` não poderá receber dessa política acesso a nenhum recurso.

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

Este exemplo mostra como criar uma política que permita a qualquer entidade principal executar qualquer ação em um recurso se a entidade principal for o `owner` do recurso OU se a entidade principal fizer parte do grupo `admins` do recurso.

```
permit(
  principal,
  action,
  resource,
)
when {
```

```
  principal == resource.owner |
  resource.admins.contains(principal)
};
```

Nega o acesso

Se uma política contiver `forbid` como efeito da política, ela restringirá as permissões, em vez de concedê-las.

> **Important**
>
> Durante a autorização, se as políticas `permit` e `forbid` forem aplicadas, `forbid` terá precedência.

Os exemplos a seguir usam os atributos definidos no aplicativo hipotético chamado PhotoFlash descrito na seção Exemplo de esquema do Guia de referência da linguagem de políticas Cedar.

Este exemplo mostra como criar uma política que impeça que a usuária `alice` execute todas as ações, exceto `readOnly`, em qualquer recurso.

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

Este exemplo mostra como criar uma política que negue o acesso a todos os recursos que tenham um atributo `private`, a menos que a entidade principal tenha o atributo `owner` para o recurso.

```
forbid (
  principal,
  action,
  resource
)
when {
  resource.private
```

```
}
unless {
  principal == resource.owner
};
```

Modelos de políticas do Amazon Verified Permissions

Você pode criar modelos de políticas do Cedar no Verified Permissions para definir uma regra de controle de acesso para seu sistema. Os modelos de políticas são políticas do Cedar com espaços reservados para `principal`,`resource` ou ambos. Os modelos de políticas permitem que uma política seja definida uma vez e, depois, anexada a várias entidades principais e recursos. As atualizações do modelo de política são refletidas em todas as entidades principais e recursos que usam o modelo. Para obter mais informações, consulte Modelos de políticas do Cedar no Guia de referência da linguagem de política Cedar.

É recomendável o uso de modelos de políticas para criar políticas que possam ser compartilhadas em toda a sua aplicação. Por exemplo, você pode criar um modelo de política para um editor que forneça permissões de leitura, edição e comentário para a entidade principal e o recurso que usa o modelo de política.

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

Quando uma a entidade principal é designada como editor de um recurso, sua aplicação pode instanciar uma política usando o modelo para fornecer permissões para que a entidade principal execute as ações de leitura, edição e comentário no recurso.

Criação de modelos de política

AWS Management Console

Para criar um modelo de política

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Modelos de política.
3. Escolha Criar modelo de política.
4. Na seção Detalhes, digite a Descrição do modelo de política.

5. Na seção Corpo do modelo de política, use os espaços reservados `?principal` e `?resource` para que as políticas criadas com base nesse modelo personalizem as permissões que elas concedem. Você pode escolher Formatar para formatar a sintaxe do seu modelo de política com o espaçamento e o recuo recomendados.
6. Escolha Criar modelo de política.

AWS CLI

Para criar um modelo de política

Você pode criar um modelo de política usando a CreatePolicyTemplateoperação. O exemplo a seguir cria um modelo de política com um espaço reservado para a entidade principal.

Veja a seguir o conteúdo do arquivo `template1.txt`.

```
"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"
}
```

Criação de políticas vinculadas a modelos

Você pode criar políticas vinculadas a modelos para vinculá-las a um modelo de política. As políticas vinculadas a modelos permanecem vinculadas a seus modelos de política. Se você alterar a declaração de política no modelo de política, todas as políticas vinculadas a esse modelo usarão

automaticamente a nova declaração para todas as decisões de autorização tomadas a partir desse momento.

AWS Management Console

Para criar uma política vinculada a modelo instanciando um modelo de política

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. Escolha Criar política e, em seguida, escolha Criar política vinculada a modelo.
4. Escolha o botão de opção ao lado do modelo de política a ser usado e, em seguida, escolha Próximo.
5. Digite a Entidade principal e o Recurso a serem usados nessa instância específica da política vinculada a modelo. Os valores especificados são exibidos no campo Visualização da declaração de política.

   > **Note**
   >
   > Os valores Entidade principal e Recurso devem ter a mesma formatação das políticas estáticas. Por exemplo, para especificar o grupo `AdminUsers` para a entidade principal, digite `Group::"AdminUsers"`. Se você digitar `AdminUsers`, será exibido um erro de validação.

6. Escolha Criar política vinculada a modelo.

   A nova política vinculada a modelo é exibida em Políticas.

AWS CLI

Para criar uma política vinculada a modelo instanciando um modelo de política

Você pode criar uma política vinculada a modelo que faça referência a um modelo de política existente e especifique valores para quaisquer espaços reservados usados pelo modelo.

O exemplo a seguir cria uma política vinculada a modelo que usa um modelo com a seguinte declaração:

```
permit(
```

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

Ele também usa o arquivo `definition.txt` a seguir para fornecer o valor do parâmetro `definition`:

```
{
    "templateLinked": {
        "policyTemplateId": "pt-4651be67-c128-4d22-8e67-9b068980c631",
        "principal": {
            "entityType": "User",
            "entityId": "alice"
        }
    }
}
```

A saída mostra o recurso, extraído do modelo, e a entidade principal, extraída do parâmetro de definição

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

Edição de modelos de política

AWS Management Console

Para editar os modelos de política

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Modelos de política. O console exibe todos os modelos de política que você criou no armazenamento de políticas atual.
3. Escolha o botão de opção ao lado de um modelo de política para exibir os detalhes sobre o modelo de política, por exemplo, quando o modelo de política foi criado, atualizado e o conteúdo do modelo de política.
4. Escolha Editar para editar o modelo de política. Atualize a Descrição da política e o Corpo da política conforme necessário e escolha Atualizar modelo de política.
5. Você pode excluir um modelo de política escolhendo o botão de opção ao lado de um modelo de política e, em seguida, escolhendo Excluir. Escolha OK para confirmar a exclusão do modelo de política.

AWS CLI

Para atualizar um modelo de política

Você pode criar uma política estática usando a UpdatePolicyoperação. O exemplo a seguir atualiza o modelo de política especificado substituindo o corpo da política por uma nova política definida em um arquivo.

Conteúdo do arquivo `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"
}
```

Exemplo de políticas vinculadas a modelos para exemplos de armazenamentos de políticas do Verified Permissions

Quando você cria um armazenamento de políticas no Verified Permissions usando o método Exemplo de armazenamento de políticas, seu armazenamento de políticas é criado com políticas predefinidas, modelos de política e um esquema para o exemplo de projeto escolhido. Os seguintes exemplos de políticas vinculadas a modelos do Verified Permissions podem ser usados com os exemplos de armazenamentos de políticas e suas respectivas políticas, modelos de políticas e esquemas.

PhotoFlashexemplos de políticas vinculadas a modelos

Este exemplo mostra como você pode criar uma política vinculada a um modelo que usa o modelo de política Conceder acesso limitado a fotos compartilhadas não privadas com um usuário e uma foto individuais.

> **Note**
>
> A linguagem de política Cedar considera uma entidade como `in`. Portanto, `principal in User::"Alice"` é equivalente a `principal == User::"Alice"`.

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

Este exemplo mostra como você pode criar uma política vinculada a um modelo que usa o modelo de política Conceder acesso limitado a fotos compartilhadas não privadas com um usuário e álbum individuais.

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

Este exemplo mostra como você pode criar uma política vinculada a um modelo que usa o modelo de política Conceder acesso limitado a fotos compartilhadas não privadas com um grupo de amigos e uma foto individual.

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

Este exemplo mostra como você pode criar uma política vinculada a um modelo que usa o modelo de política Conceder acesso limitado a fotos compartilhadas não privadas com um grupo de amigos e um álbum.

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

Este exemplo mostra como você pode criar uma política vinculada a um modelo que usa o modelo de política Conceder acesso total a fotos compartilhadas não privadas com um grupo de amigos e uma foto individual.

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

Este exemplo mostra como você pode criar uma política vinculada a um modelo que usa o modelo de política Bloquear usuário de uma conta.

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

DigitalPetStore

O repositório de políticas de DigitalPetStore amostra não inclui nenhum modelo de política. Você pode visualizar as políticas incluídas no repositório de políticas escolhendo Políticas no painel de navegação à esquerda após criar o DigitalPetStoreexemplo de armazenamento de políticas.

TinyToDo exemplos de políticas vinculadas a modelos

Este exemplo mostra como você pode criar uma política vinculada a modelo que usa o modelo de política que concede acesso de visualizador a um usuário individual e uma lista de tarefas.

```
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"
);
```

Este exemplo mostra como você pode criar uma política vinculada a modelo que usa o modelo de política que concede acesso de editor a um usuário individual e uma lista de tarefas.

```
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"
);
```

Uso do Amazon Verified Permissions com provedores de identidade

Uma fonte de identidade é uma representação de um provedor de identidade externo (IdP) nas Permissões Verificadas da Amazon. As fontes de identidade fornecem informações de um usuário que se autenticou com um IdP que tem uma relação de confiança com seu repositório de políticas. Quando seu aplicativo faz uma solicitação de autorização com um token de uma fonte de identidade, seu repositório de políticas pode tomar decisões de autorização a partir das propriedades do usuário e das permissões de acesso. As fontes de identidade de permissões verificadas melhoram a autorização com uma conexão direta com seu repositório central de identidades e serviço de autenticação.

Você pode usar os provedores de identidade () do OpenID Connect (OIDC) com permissões IdPs verificadas. Seu aplicativo pode gerar solicitações de autorização com identidade (ID) OIDC ou acessar tokens web JSON (JWTs). Com tokens de ID, as Permissões Verificadas lêem IDs de usuários e declarações de atributos como principais para controle de acesso baseado em atributos (ABAC). Com os tokens de acesso, as Permissões Verificadas lêem os IDs dos usuários como principais e outras declarações como contexto. Com os dois tipos de token, você pode mapear uma declaração como `groups` para um grupo principal e criar políticas que avaliem o controle de acesso baseado em funções (RBAC).

Você pode adicionar um grupo de usuários do Amazon Cognito ou um IdP personalizado do OpenID Connect (OIDC) como sua fonte de identidade.

Tópicos

Trabalhando com fontes de identidade do Amazon Cognito

As permissões verificadas trabalham em estreita colaboração com os grupos de usuários do Amazon Cognito. Os JWTs do Amazon Cognito têm uma estrutura previsível. As permissões verificadas reconhecem essa estrutura e tiram o máximo proveito das informações que ela contém. Por exemplo, você pode implementar um modelo de autorização de controle de acesso baseado em função (RBAC) com tokens de ID ou tokens de acesso.

Uma nova fonte de identidade de grupos de usuários do Amazon Cognito exige as seguintes informações:

- Região da AWS A.
- O ID do grupo de usuários.
- O tipo de entidade do usuário que você deseja associar à sua fonte de identidade, por exemplo`MyCorp::User`.
- O tipo de entidade do grupo que você deseja associar à sua fonte de identidade, por exemplo`MyCorp::UserGroup`.
- (Opcional) Os IDs de cliente do seu grupo de usuários que você deseja autorizar a fazer solicitações ao seu repositório de políticas.

Como as Permissões Verificadas só funcionam com grupos de usuários do Amazon Cognito nos mesmos Conta da AWS, você não pode especificar uma fonte de identidade em outra conta. As permissões verificadas definem o prefixo da entidade — o identificador da fonte de identidade que você deve referenciar nas políticas que atuam de acordo com os diretores do grupo de usuários — como o ID do seu grupo de usuários, por exemplo. `us-west-2_EXAMPLE`

As declarações de token do grupo de usuários podem conter atributos, escopos, grupos, IDs de clientes e dados personalizados. Os JWTs do Amazon Cognito têm a capacidade de incluir uma variedade de informações que podem contribuir para as decisões de autorização nas Permissões verificadas. Isso inclui:

1. Declarações de nome de usuário e grupo com um `cognito:` prefixo
2. Atributos de usuário personalizados com um `custom: prefix`
3. Declarações personalizadas adicionadas em tempo de execução
4. Reivindicações padrão do OIDC, como e `sub` `email`

Abordamos essas reivindicações em detalhes e como gerenciá-las nas políticas de permissões verificadas, emTrabalhando com fontes de identidade em esquemas e políticas.

> **Important**
>
> Embora você possa revogar os tokens do Amazon Cognito antes que eles expirem, os JWTs são considerados recursos sem estado independentes, com assinatura e validade. Espera-se que os serviços em conformidade com o JSON Web Token RFC 7519 validem os tokens remotamente e não precisem validá-los com o emissor. Isso significa que o Verified Permissions pode conceder acesso com base em um token revogado ou emitido para o usuário posteriormente excluído. Para mitigar esse risco, recomendamos que você crie seus tokens com o menor período de validade possível e revogue os tokens de atualização quando quiser remover a autorização para continuar a sessão de um usuário.

As políticas do Cedar para fontes de identidade de grupos de usuários em Permissões verificadas usam uma sintaxe especial para nomes de declarações que contêm caracteres diferentes de alfanuméricos e sublinhado (). _ Isso inclui declarações de prefixo do grupo de usuários que contêm um `:` `cognito:username` caractere, como e. `custom:department` Para escrever uma condição de política que faça referência à `custom:department` reivindicação `cognito:username` or, escreva-a como `principal["cognito:username"]` e`principal["custom:department"]`, respectivamente.

> **Note**
>
> Se um token contiver uma declaração com um `custom:` prefixo `cognito:` or e um nome de solicitação com o valor literal `cognito` ou`custom`, uma solicitação de autorização com IsAuthorizedWithTokenfalhará com a. `ValidationException`

Este exemplo mostra como você pode criar uma política que faça referência a algumas das reivindicações dos grupos de usuários do Amazon Cognito associadas a um principal.

```
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"
};
```

Para obter mais informações sobre o mapeamento de declarações, consulteMapeamento de tokens de ID para o esquema. Para obter mais informações sobre autorização para usuários do Amazon Cognito, consulte Autorização com permissões verificadas da Amazon no Guia do desenvolvedor do Amazon Cognito.

Trabalhando com fontes de identidade do OIDC

Você também pode configurar qualquer IdP compatível do OpenID Connect (OIDC) como fonte de identidade de um repositório de políticas. Os provedores do OIDC são semelhantes aos grupos de usuários do Amazon Cognito: eles produzem JWTs como produto da autenticação. Para adicionar um provedor OIDC, você deve fornecer uma URL do emissor

Uma nova fonte de identidade do OIDC requer as seguintes informações:

- O URL do emissor. As permissões verificadas devem ser capazes de descobrir um `.well-known/openid-configuration` endpoint nesse URL.
- O tipo de token que você deseja usar nas solicitações de autorização. Nesse caso, você escolheu o token de identidade.
- O tipo de entidade do usuário que você deseja associar à sua fonte de identidade, por exemplo`MyCorp::User`.
- O tipo de entidade do grupo que você deseja associar à sua fonte de identidade, por exemplo`MyCorp::UserGroup`.
- Um exemplo de token de ID ou uma definição das declarações no token de ID.
- O prefixo que você deseja aplicar às IDs de entidade de usuário e grupo. Na CLI e na API, você pode escolher esse prefixo. Nos repositórios de políticas que você cria com o Configurar com o API Gateway e uma fonte de identidade ou a opção Configuração guiada, as Permissões Verificadas atribuem um prefixo do nome do emissor menos`https://`, por exemplo. `MyCorp::User::"auth.example.com|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"`

A autorização com fontes de identidade do OIDC usa as mesmas operações de API que as fontes de identidade do grupo de usuários: IsAuthorizedWithTokene BatchIs AuthorizedWith o token.

Este exemplo mostra como você pode criar uma política que permita o acesso aos relatórios de fim de ano para funcionários do departamento de contabilidade, que tenham uma classificação confidencial e não estejam em um escritório satélite. As permissões verificadas derivam esses atributos das declarações no token de ID do principal.

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

Validação de clientes e públicos

Quando você adiciona uma fonte de identidade a um repositório de políticas, as Permissões Verificadas têm opções de configuração que verificam se os tokens de ID e acesso estão sendo usados conforme o esperado. Essa validação acontece no processamento `IsAuthorizedWithToken` e nas solicitações de `BatchIsAuthorizedWithToken` API. O comportamento difere entre tokens de ID e acesso e entre as fontes de identidade do Amazon Cognito e do OIDC. Com os provedores de grupos de usuários do Amazon Cognito, as Permissões Verificadas podem validar a ID do cliente em tokens de ID e de acesso. Com os provedores do OIDC, as Permissões Verificadas podem validar o ID do cliente em tokens de ID e o público em tokens de acesso.

Um ID de cliente é um identificador associado a um aplicativo OAuth ou OIDC configurado com o provedor, por exemplo. `1example23456789` Um público é um caminho de URL associado à parte confiável ou ao destino pretendido do aplicativo de destino, por exemplo`https://myapplication.example.com`. A `aud` afirmação nem sempre está associada ao público.

O Verified Permissions realiza a validação do público da fonte de identidade e do cliente da seguinte forma:

Amazon Cognito

- Os tokens de ID do Amazon Cognito têm uma `aud` declaração que contém o ID do cliente do aplicativo. Os tokens de acesso têm uma `client_id` declaração que também contém o ID do cliente do aplicativo.

Quando você insere um ou mais valores para a validação do aplicativo cliente em sua fonte de identidade, o Verified Permissions compara essa lista de IDs do cliente do aplicativo com a declaração do token de ID ou com a `aud` declaração do token `client_id` de acesso. As permissões verificadas não validam uma URL de público confiável para fontes de identidade do Amazon Cognito.

OIDC

Os tokens de ID do OIDC têm uma `aud` declaração que contém uma lista de IDs de clientes. Os tokens de acesso têm uma `aud` declaração que contém o URL do público do token. Os tokens de acesso também têm uma `client_id` declaração que contém o ID do cliente pretendido.

Você pode inserir um ou mais valores para validação do Audience com um provedor OIDC. Quando você escolhe um tipo de token de ID de token, as Permissões verificadas validam a ID do cliente, verificando se pelo menos um membro das IDs do cliente na `aud` declaração corresponde a um valor de validação de público.

As Permissões verificadas validam o público para tokens de acesso, verificando se a `aud` afirmação corresponde a um valor de validação do público. Esse valor do token de acesso vem principalmente da `aud` reivindicação, mas pode vir da `client_id` reivindicação `cid` ou se não houver nenhuma `aud` reivindicação. Verifique com seu IdP a afirmação e o formato corretos do público.

Um exemplo de valor de validação de público do token de ID é`1example23456789`.

Um exemplo de valor de validação de público do token de acesso é`https://myapplication.example.com`.

Autorização do lado do cliente para JWTs

Talvez você queira processar tokens web JSON em seu aplicativo e passar suas reivindicações para Permissões Verificadas sem usar uma fonte de identidade do repositório de políticas. Você pode extrair seus atributos de entidade de um JSON Web Token (JWT) e analisá-los em Permissões verificadas.

Este exemplo mostra como você pode chamar Permissões Verificadas de um IdC 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 = "PSEXAMPLEabcdefg111111"; // set your own policy store id

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

    return authResult;

}

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

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

  });

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

[1] Esse exemplo de código usa a biblioteca aws-jwt-verify para verificar JWTs assinados por compatíveis com OIDC. IdPs

Criação de origens de identidade do Amazon Verified Permissions

O procedimento a seguir adiciona uma fonte de identidade a um repositório de políticas existente. Depois de adicionar sua fonte de identidade, você deve adicionar atributos ao seu esquema.

Você também pode criar uma fonte de identidade ao criar um novo repositório de políticas no console de Permissões Verificadas. Nesse processo, você pode importar automaticamente as declarações em seus tokens de origem de identidade para os atributos da entidade. Escolha a configuração guiada ou a opção Configurar com o API Gateway e um provedor de identidade. Essas opções também criam políticas iniciais.

> **Note**
>
> As origens de identidade só estarão disponíveis no painel de navegação à esquerda depois que você criar um armazenamento de políticas. As origens de identidade criadas por você são associadas ao armazenamento de políticas atual.

Você pode omitir o tipo de entidade principal ao criar uma fonte de identidade com create-identity-source na AWS CLI ou Source na API de permissões CreateIdentityverificadas. No entanto, um tipo de entidade em branco cria uma fonte de identidade com um tipo de entidade de`AWS::Cognito`. Esse nome de entidade não é compatível com o esquema do repositório de políticas. Para integrar as identidades do Amazon Cognito com seu esquema de armazenamento de políticas, você deve definir o tipo de entidade principal como uma entidade de armazenamento de políticas compatível.

Tópicos

- Fonte de identidade do Amazon Cognito
- Fonte de identidade OIDC

Fonte de identidade do Amazon Cognito

AWS Management Console

Para criar uma origem de identidade de grupos de usuários do Amazon Cognito

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Origens de identidade.
3. Escolha Criar origem de identidade.
4. Em Detalhes do grupo de usuários do Cognito, selecione Região da AWS e insira o ID do grupo de usuários para sua fonte de identidade.
5. Em Configuração principal, escolha um tipo principal para a fonte de identidade. As identidades dos grupos de usuários conectados do Amazon Cognito serão mapeadas para o tipo de entidade principal selecionado.
6. Em Configuração de grupo, selecione Usar grupo Cognito se quiser mapear a declaração do grupo `cognito:groups` de usuários. Escolha um tipo de entidade que seja pai do tipo principal.
7. Em Validação do aplicativo cliente, escolha se deseja validar as IDs do aplicativo cliente.
   - Para validar IDs de aplicação cliente, escolha Aceitar somente tokens com IDs de aplicação cliente correspondentes. Escolha Adicionar novo ID de aplicação cliente para cada ID de aplicação cliente a ser validado. Para remover um ID de aplicação cliente adicionado, escolha Remover ao lado do ID de aplicação cliente.
   - Escolha Não valide os IDs da aplicação cliente se você não quiser validar IDs de aplicação cliente.
8. Escolha Criar origem de identidade.
9. Antes de fazer referência aos atributos extraídos dos tokens de identidade ou acesso nas políticas do Cedar, você deve atualizar o esquema para informar o Cedar sobre o tipo de entidade principal criado por sua origem de identidade. Essa adição ao esquema deve incluir os atributos que você deseja referenciar nas políticas do Cedar. Para obter mais informações sobre o mapeamento dos atributos de token do Amazon Cognito para os atributos de entidade principal do Cedar, consulte Trabalhando com fontes de identidade em esquemas e políticas.

Quando você cria um repositório de políticas vinculado à API, o Verified Permissions consulta seu grupo de usuários em busca de atributos de usuário e cria um esquema em que seu tipo principal é preenchido com atributos do grupo de usuários.

AWS CLI

Para criar uma origem de identidade de grupos de usuários do Amazon Cognito

Você pode criar uma fonte de identidade usando a operação CreateIdentityFonte. O exemplo a seguir cria uma origem de identidade que pode acessar identidades autenticadas de um grupo de usuários do Amazon Cognito.

O arquivo `config.txt` a seguir contém os detalhes do grupo de usuários do Amazon Cognito que será usado pelo parâmetro --configuration no 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"
}
```

Antes de fazer referência aos atributos extraídos dos tokens de identidade ou acesso nas políticas do Cedar, você deve atualizar o esquema para informar o Cedar sobre o tipo de entidade principal criado por sua origem de identidade. Essa adição ao esquema deve incluir os atributos que você deseja referenciar nas políticas do Cedar. Para obter mais informações sobre o mapeamento dos atributos de token do Amazon Cognito para os atributos de entidade principal do Cedar, consulte Trabalhando com fontes de identidade em esquemas e políticas.

Quando você cria um repositório de políticas vinculado à API, o Verified Permissions consulta seu grupo de usuários em busca de atributos de usuário e cria um esquema em que seu tipo principal é preenchido com atributos do grupo de usuários.

Para obter mais informações sobre o uso de tokens de acesso e identidade do Amazon Cognito para usuários autenticados no Verified Permissions, consulte Autorização com o Amazon Verified Permissions no Guia do desenvolvedor do Amazon Cognito.

Fonte de identidade OIDC

AWS Management Console

Para criar uma fonte de identidade do OpenID Connect (OIDC)

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Origens de identidade.
3. Escolha Criar origem de identidade.
4. Escolha o provedor externo do OIDC.
5. Em URL do emissor, insira a URL do seu emissor do OIDC. Esse é o endpoint do serviço que fornece o servidor de autorização, as chaves de assinatura e outras informações sobre seu provedor, por exemplo`https://auth.example.com`. Seu URL de emissor deve hospedar um documento de descoberta do OIDC em. `/.well-known/openid-configuration`
6. Em Tipo de token, escolha o tipo de OIDC JWT que você deseja que seu aplicativo envie para autorização. Para ter mais informações, consulte Trabalhando com fontes de identidade em esquemas e políticas.
7. Em Declarações de usuário e grupo, escolha uma entidade de usuário e uma reivindicação de usuário para a fonte de identidade. A entidade Usuário é uma entidade em seu repositório

de políticas que você deseja indicar aos usuários do seu provedor OIDC. A reivindicação do usuário é uma reivindicação`sub`, normalmente, de seu ID ou token de acesso que contém o identificador exclusivo da entidade a ser avaliada. As identidades do IdP OIDC conectado serão mapeadas para o tipo principal selecionado.

8. Em Declarações de usuário e grupo, escolha uma entidade de grupo e uma afirmação de grupo para a fonte de identidade. A entidade do Grupo é controladora da entidade Usuário. As reivindicações de grupo são mapeadas para essa entidade. A declaração de grupo é uma afirmação, normalmente`groups`, de seu ID ou token de acesso que contém uma string, JSON ou string delimitada por espaço de nomes de grupos de usuários para a entidade a ser avaliada. As identidades do IdP OIDC conectado serão mapeadas para o tipo principal selecionado.
9. Em Validação de público, insira os IDs do cliente ou os URLs do público que você deseja que seu repositório de políticas aceite nas solicitações de autorização, se houver.
10. Escolha Criar origem de identidade.
11. Atualize seu esquema para que o Cedar conheça o tipo de principal que sua fonte de identidade cria. Essa adição ao esquema deve incluir os atributos que você deseja referenciar nas políticas do Cedar. Para obter mais informações sobre o mapeamento dos atributos de token do Amazon Cognito para os atributos de entidade principal do Cedar, consulte Trabalhando com fontes de identidade em esquemas e políticas.

    Quando você cria um repositório de políticas vinculado à API, o Verified Permissions consulta seu grupo de usuários em busca de atributos de usuário e cria um esquema em que seu tipo principal é preenchido com atributos do grupo de usuários.

AWS CLI

Para criar uma fonte de identidade OIDC

Você pode criar uma fonte de identidade usando a operação CreateIdentityFonte. O exemplo a seguir cria uma origem de identidade que pode acessar identidades autenticadas de um grupo de usuários do Amazon Cognito.

O `config.txt` arquivo a seguir contém os detalhes de um IdP do OIDC para uso pelo `--configuration` parâmetro do comando. `create-identity-source` Este exemplo cria uma fonte de identidade OIDC para tokens de ID.

```
{
```

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

O `config.txt` arquivo a seguir contém os detalhes de um IdP do OIDC para uso pelo `--configuration` parâmetro do comando. `create-identity-source` Este exemplo cria uma fonte de identidade OIDC para tokens de acesso.

```
{
    "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": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Antes de fazer referência aos atributos extraídos dos tokens de identidade ou acesso nas políticas do Cedar, você deve atualizar o esquema para informar o Cedar sobre o tipo de entidade principal criado por sua origem de identidade. Essa adição ao esquema deve incluir os atributos que você deseja referenciar nas políticas do Cedar. Para obter mais informações sobre o mapeamento dos atributos de token do Amazon Cognito para os atributos de entidade principal do Cedar, consulte Trabalhando com fontes de identidade em esquemas e políticas.

Quando você cria um repositório de políticas vinculado à API, o Verified Permissions consulta seu grupo de usuários em busca de atributos de usuário e cria um esquema em que seu tipo principal é preenchido com atributos do grupo de usuários.

Edição de origens de identidade do Amazon Verified Permissions

Você pode editar alguns parâmetros da sua fonte de identidade depois de criá-la. Se o esquema do repositório de políticas corresponder aos atributos da fonte de identidade, observe que você deve atualizar o esquema separadamente para refletir as alterações feitas na fonte de identidade.

Tópicos

- Fonte de identidade dos grupos de usuários do Amazon Cognito
- Fonte de identidade do OpenID Connect (OIDC)

Fonte de identidade dos grupos de usuários do Amazon Cognito

AWS Management Console

Para atualizar uma origem de identidade de grupos de usuários do Amazon Cognito

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Origens de identidade.

3. Escolha o ID da origem de identidade a ser editada.
4. Selecione a opção Editar.
5. Em Detalhes do grupo de usuários do Cognito, selecione Região da AWS e digite o ID do grupo de usuários para sua fonte de identidade.
6. Em Detalhes do principal, você pode atualizar o tipo de principal para a fonte de identidade. As identidades dos grupos de usuários conectados do Amazon Cognito serão mapeadas para o tipo de entidade principal selecionado.
7. Em Configuração de grupo, selecione Usar grupo Cognito se quiser mapear a declaração do grupo `cognito:groups` de usuários. Escolha um tipo de entidade que seja pai do tipo principal.
8. Em Validação do aplicativo cliente, escolha se deseja validar as IDs do aplicativo cliente.
   - Para validar IDs de aplicação cliente, escolha Aceitar somente tokens com IDs de aplicação cliente correspondentes. Escolha Adicionar novo ID de aplicação cliente para cada ID de aplicação cliente a ser validado. Para remover um ID de aplicação cliente adicionado, escolha Remover ao lado do ID de aplicação cliente.
   - Escolha Não valide os IDs da aplicação cliente se você não quiser validar IDs de aplicação cliente.
9. Escolha Salvar alterações.
10. Se você alterou o tipo de entidade principal da origem de identidade, deverá atualizar seu esquema para refletir corretamente o tipo de entidade principal atualizado.

Você pode excluir uma origem de identidade escolhendo o botão de opção ao lado de uma origem de identidade e, em seguida, escolhendo Excluir origem de identidade. Digite `delete` na caixa de texto e escolha Excluir origem de identidade para confirmar a exclusão da origem de identidade.

AWS CLI

Para atualizar uma origem de identidade de grupos de usuários do Amazon Cognito

Você pode atualizar uma fonte de identidade usando a operação UpdateIdentityFonte. O exemplo a seguir atualiza a origem de identidade especificada para usar um grupo de usuários diferente do Amazon Cognito.

O arquivo `config.txt` a seguir contém os detalhes do grupo de usuários do Amazon Cognito que será usado pelo parâmetro --configuration no 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 você alterar o tipo de entidade principal da origem de identidade, será necessário atualizar o esquema para refletir corretamente o tipo de entidade principal atualizado.

Fonte de identidade do OpenID Connect (OIDC)

AWS Management Console

Para atualizar uma fonte de identidade do OIDC

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Origens de identidade.
3. Escolha o ID da origem de identidade a ser editada.
4. Selecione a opção Editar.
5. Nos detalhes do provedor OIDC, altere a URL do emissor conforme necessário.

6. Em Mapear declarações de token para atributos de esquema, altere as associações entre declarações de usuário e grupo e os tipos de entidade de armazenamento de políticas, conforme necessário. Depois de alterar os tipos de entidade, você deve atualizar suas políticas e atributos do esquema para aplicar aos novos tipos de entidade.
7. Na validação de público, adicione ou remova valores de público que você deseja aplicar.
8. Escolha Salvar alterações.

Você pode excluir uma origem de identidade escolhendo o botão de opção ao lado de uma origem de identidade e, em seguida, escolhendo Excluir origem de identidade. Digite `delete` na caixa de texto e escolha Excluir origem de identidade para confirmar a exclusão da origem de identidade.

AWS CLI

Para atualizar uma fonte de identidade do OIDC

Você pode atualizar uma fonte de identidade usando a operação UpdateIdentityFonte. O exemplo a seguir atualiza a fonte de identidade especificada para usar um provedor OIDC diferente.

O arquivo `config.txt` a seguir contém os detalhes do grupo de usuários do Amazon Cognito que será usado pelo parâmetro --configuration no 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 você alterar o tipo de entidade principal da origem de identidade, será necessário atualizar o esquema para refletir corretamente o tipo de entidade principal atualizado.

Trabalhando com fontes de identidade em esquemas e políticas

Talvez você queira adicionar uma fonte de identidade a um repositório de políticas e mapear reivindicações do provedor ao seu esquema de armazenamento de políticas. Você pode automatizar esse processo ou atualizar seu esquema manualmente. Esta seção do guia do usuário tem as seguintes informações:

- Quando você pode preencher automaticamente os atributos de um esquema de armazenamento de políticas
- Como usar as declarações de token do Amazon Cognito e do OIDC em suas políticas de permissões verificadas
- Como criar manualmente um esquema para uma fonte de identidade

Os repositórios de políticas vinculados à API e os repositórios de políticas com uma fonte de identidade por meio da configuração guiada não exigem o mapeamento manual dos atributos do token de identidade (ID) para o esquema. Você pode fornecer permissões verificadas com os atributos em seu grupo de usuários ou tokens OIDC e criar um esquema preenchido com atributos de usuário. Na autorização do token de ID, as Permissões verificadas mapeiam as reivindicações aos atributos de uma entidade principal. Talvez seja necessário mapear manualmente os tokens do Amazon Cognito para o seu esquema nas seguintes condições:

- Você criou um repositório de políticas em branco ou um repositório de políticas a partir de uma amostra.

- Você deseja estender o uso de tokens de acesso além do controle de acesso baseado em funções (RBAC).
- Você cria repositórios de políticas com a API REST de permissões verificadas, um AWS SDK ou o. AWS CDK

Para usar o Amazon Cognito ou um provedor de identidade (IdP) do OIDC como fonte de identidade em seu repositório de políticas de permissões verificadas, você deve ter atributos de provedor em seu esquema. Se você criou seu repositório de políticas de uma forma que preenche automaticamente seu esquema a partir das informações do provedor em um token de ID, você está pronto para escrever políticas. Se você criar um repositório de políticas sem um esquema para sua fonte de identidade, deverá adicionar atributos do provedor ao esquema. Seu esquema deve corresponder às entidades que os tokens do provedor criam IsAuthorizedWithTokenou às solicitações da BatchIsAuthorizedWithToken API. Em seguida, você pode escrever políticas usando atributos do token do provedor.

Para obter mais informações sobre o uso do Amazon Cognito ID e tokens de acesso para usuários autenticados em Permissões verificadas, consulte Autorização com permissões verificadas da Amazon no Guia do desenvolvedor do Amazon Cognito.

Tópicos

Coisas que você deve saber sobre mapeamento de esquemas

O mapeamento de atributos difere entre os tipos de token

Na autorização do token de acesso, as permissões verificadas mapeiam as reivindicações de acordo com o contexto. Na autorização do token de ID, as Permissões verificadas mapeiam as reivindicações para os atributos principais. Para repositórios de políticas que você cria no console de Permissões Verificadas, somente repositórios de políticas vazios e de amostra deixam você sem fonte de identidade e exigem que você preencha seu esquema com atributos do grupo de usuários para autorização do token de ID. A autorização do token de acesso é baseada no controle

de acesso baseado em funções (RBAC) com declarações de associação a grupos e não mapeia automaticamente outras reivindicações para o esquema do repositório de políticas.

Os atributos da fonte de identidade não são obrigatórios

Quando você cria uma fonte de identidade no console de Permissões verificadas, nenhum atributo é marcado como obrigatório. Isso evita que declarações perdidas causem erros de validação nas solicitações de autorização. Você pode definir atributos como obrigatórios conforme necessário, mas eles devem estar presentes em todas as solicitações de autorização.

O RBAC não exige atributos no esquema

Os esquemas para fontes de identidade dependem das associações de entidades que você faz ao adicionar sua fonte de identidade. Uma fonte de identidade mapeia uma afirmação para um tipo de entidade de usuário e uma afirmação para um tipo de entidade de grupo. Esses mapeamentos de entidades são o núcleo de uma configuração de origem de identidade. Com essas informações mínimas, você pode criar políticas que executem ações de autorização para usuários específicos e grupos específicos dos quais os usuários possam ser membros, em um modelo de controle de acesso baseado em função (RBAC). A adição de declarações de token ao esquema amplia o escopo de autorização do seu repositório de políticas. Os atributos de usuário dos tokens de ID têm informações sobre usuários que podem contribuir para a autorização do controle de acesso baseado em atributos (ABAC). Os atributos de contexto dos tokens de acesso têm informações como escopos do OAuth 2.0 que podem contribuir com informações adicionais de controle de acesso do seu provedor, mas exigem modificações adicionais no esquema.

As opções Configurar com o API Gateway e uma fonte de identidade e Configuração guiada no console de permissões verificadas atribuem reivindicações de token de ID ao esquema. Esse não é o caso das reivindicações de token de acesso. Para adicionar declarações de token de acesso que não sejam de grupo ao seu esquema, você deve editá-lo no modo JSON e adicionar atributos commonTypes. Para ter mais informações, consulte Mapeamento de tokens de acesso.

Os grupos do OIDC afirmam que suporta vários formatos

Ao adicionar um provedor OIDC, você pode escolher o nome da reivindicação do grupo em ID ou tokens de acesso que deseja mapear para a associação de um usuário ao grupo em seu repositório de políticas. As permissões verificadas reconhecem as reivindicações de grupos nos seguintes formatos:

1. Cadeia de caracteres sem espaços: `"groups": "MyGroup"`

2. Lista delimitada por espaço:. `"groups": "MyGroup1 MyGroup2 MyGroup3"` Cada string é um grupo.
3. Lista JSON (delimitada por vírgula): `"groups": ["MyGroup1", "MyGroup2", "MyGroup3"]`

> **Note**
>
> As Permissões verificadas interpretam cada string em uma declaração de grupos separados por espaço como um grupo separado. Para interpretar um nome de grupo com um caractere de espaço como um único grupo, substitua ou remova o espaço na declaração. Por exemplo, formate um grupo chamado `My Group` como`MyGroup`.

Escolha um tipo de token

A forma como seu repositório de políticas funciona com sua fonte de identidade depende de uma decisão importante na configuração da fonte de identidade: se você processará tokens de ID ou de acesso. Com um provedor de identidade do Amazon Cognito, você pode escolher o tipo de token ao criar um armazenamento de políticas vinculado à API. Ao criar um repositório de políticas vinculado à API, você deve escolher se deseja configurar a autorização para tokens de ID ou de acesso. Essas informações afetam os atributos do esquema que as Permissões Verificadas aplicam ao seu armazenamento de políticas e a sintaxe do autorizador Lambda para sua API do API Gateway. Com um provedor OIDC, você deve escolher um tipo de token ao adicionar a fonte de identidade. Você pode escolher ID ou token de acesso, e sua escolha exclui que o tipo de token não escolhido seja processado em seu repositório de políticas. Especialmente se você quiser se beneficiar do mapeamento automático de declarações de token de ID para atributos no console de permissões verificadas, decida com antecedência sobre o tipo de token que você deseja processar antes de criar sua fonte de identidade. Alterar o tipo de token exige um esforço significativo para refatorar suas políticas e seu esquema. Os tópicos a seguir descrevem o uso de tokens de ID e acesso com repositórios de políticas.

O analisador Cedar requer colchetes para alguns caracteres

As políticas geralmente fazem referência aos atributos do esquema em um formato como`principal.username`. No caso da maioria dos caracteres não alfanuméricos`:`, como, ou`.`, `/` que podem aparecer nos nomes de reivindicações de token, as Permissões Verificadas não podem analisar um valor de condição como `principal.cognito:groups` ou. `context.ip-address` Em vez disso, você deve formatar essas condições com a notação de colchetes no

formato `principal["cognito:username"]` ou`context["ip-address"]`, respectivamente. O caractere sublinhado _ é um caractere válido nos nomes das reivindicações e a única exceção não alfanumérica a esse requisito.

Um exemplo parcial de esquema para um atributo principal desse tipo tem a seguinte aparência:

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

Um exemplo parcial de esquema para um atributo de contexto desse tipo tem a seguinte aparência:

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

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

Um exemplo de política para atributos que serão validados em relação a esse esquema é semelhante ao seguinte:

```
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.*"
};
```

Mapeamento de tokens de ID para o esquema

As permissões verificadas processam as reivindicações de token de ID como atributos do usuário: seus nomes e títulos, sua associação ao grupo, suas informações de contato. Os tokens de ID são mais úteis em um modelo de autorização de controle de acesso baseado em atributos (ABAC). Quando você quiser que as Permissões Verificadas analisem o acesso aos recursos com base em quem está fazendo a solicitação, escolha tokens de ID para sua fonte de identidade.

Tokens de ID do Amazon Cognito

Os tokens de ID do Amazon Cognito funcionam com a maioria das bibliotecas confiáveis do OIDC. Eles ampliam os recursos do OIDC com reivindicações adicionais. Seu aplicativo pode autenticar o usuário com as operações da API de autenticação de grupos de usuários do Amazon Cognito ou com a interface de usuário hospedada do grupo de usuários. Para obter mais informações, consulte Como usar a API e os endpoints no Guia do Desenvolvedor do Amazon Cognito.

Declarações úteis em tokens de ID do Amazon Cognito

`cognito:username` e *`preferred_username`*

Variantes do nome de usuário do usuário.

sub

O identificador de usuário exclusivo (UUID) do usuário

Reivindicações com um *custom:* prefixo

Um prefixo para atributos personalizados do grupo de usuários, como`custom:employmentStoreCode`.

Reivindicações padrão

Afirmações padrão do OIDC, como e. `email` `phone_number` Para obter mais informações, consulte Declarações padrão no OpenID Connect Core 1.0 incorporando o conjunto de erratas 2.

cognito:groups

As associações de um usuário ao grupo. Em um modelo de autorização baseado no controle de acesso baseado em funções (RBAC), essa declaração apresenta as funções que você pode avaliar em suas políticas.

Reivindicações transitórias

Declarações que não são propriedade do usuário, mas são adicionadas em tempo de execução por um acionador Lambda de pré-geração de tokens do grupo de usuários. As reivindicações transitórias se assemelham às reivindicações padrão, mas estão fora do padrão, por exemplo `tenant` ou. `department`

Nas políticas que fazem referência aos atributos do Amazon Cognito que têm um `:` separador, faça referência aos atributos no formato. `principal["`*`cognito:username`*`"]` A reivindicação de funções `cognito:groups` é uma exceção a essa regra. As permissões verificadas mapeiam o conteúdo dessa declaração para as entidades principais da entidade do usuário.

Para obter mais informações sobre a estrutura dos tokens de ID dos grupos de usuários do Amazon Cognito, consulte Uso do token de ID no Guia do Desenvolvedor do Amazon Cognito.

O exemplo de token de ID a seguir tem cada um dos quatro tipos de atributos. Ele inclui a reivindicação específica do Amazon Cognito `cognito:username`, a reivindicação personalizada `custom:employmentStoreCode`, a reivindicação padrão `email` e a reivindicação temporária `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"
}
```

Ao criar uma fonte de identidade com seu grupo de usuários do Amazon Cognito, você especifica o tipo de entidade principal com a qual o Verified Permissions gera nas solicitações de autorização. `IsAuthorizedWithToken` Suas políticas poderão, então, testar os atributos dessa entidade principal como parte da avaliação dessa solicitação. Seu esquema define o tipo e os atributos principais de uma fonte de identidade e, em seguida, você pode referenciá-los nas políticas do Cedar.

Você também especifica o tipo de entidade de grupo que deseja derivar da declaração do grupo de tokens de ID. Nas solicitações de autorização, as Permissões verificadas mapeiam cada membro da reivindicação do grupo para esse tipo de entidade do grupo. Nas políticas, você pode referenciar essa entidade do grupo como principal.

O exemplo a seguir mostra como refletir os atributos do exemplo de token de identidade no esquema do Verified Permissions. Para obter mais informações sobre a edição do esquema, consulte Edição de esquemas no modo JSON. Se a configuração da origem de identidade especificar o tipo de entidade principal `User`, você poderá incluir algo semelhante ao exemplo a seguir para disponibilizar esses atributos ao 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
         }
      }
   }
}
```

Após atualizar o esquema para refletir os atributos do Amazon Cognito, você poderá criar políticas que façam referência aos atributos.

```
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"
};
```

Tokens de ID OIDC

Trabalhar com tokens de ID de um provedor OIDC é praticamente o mesmo que trabalhar com tokens de ID do Amazon Cognito. A diferença está nas reivindicações. Seu IdP pode apresentar atributos padrão do OIDC ou ter um esquema personalizado. Ao criar um novo repositório de políticas no console de Permissões verificadas, você pode adicionar uma fonte de identidade do

OIDC com um exemplo de token de ID ou mapear manualmente as declarações de token para os atributos do usuário. Como as Permissões Verificadas não conhecem o esquema de atributos do seu IdP, você deve fornecer essas informações.

Para ter mais informações, consulte Criação de armazenamentos de políticas do Verified Permissions.

Veja a seguir um exemplo de esquema para um repositório de políticas com uma fonte de identidade OIDC.

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

A política a seguir se aplica aos membros de um grupo em seu provedor de OIDC.

```
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*"
};
```

Mapeamento de tokens de acesso

As permissões verificadas processam declarações de token de acesso diferentes das reivindicações do grupo como atributos da ação ou atributos de contexto. Além da associação ao grupo, os tokens de acesso do seu IdP podem conter informações sobre o acesso à API. Os tokens de acesso são úteis em modelos de autorização que usam controle de acesso baseado em funções (RBAC). Os modelos de autorização que dependem de declarações de token de acesso que não sejam a associação ao grupo exigem um esforço adicional na configuração do esquema.

Mapeamento de tokens de acesso do Amazon Cognito

Os tokens de acesso do Amazon Cognito têm reivindicações que podem ser usadas para autorização:

Declarações úteis em tokens de acesso do Amazon Cognito

`client_id`

O ID do aplicativo cliente de uma parte confiável do OIDC. Com a ID do cliente, as Permissões Verificadas podem verificar se a solicitação de autorização vem de um cliente permitido para o repositório de políticas. Na autorização machine-to-machine (M2M), o sistema solicitante autoriza uma solicitação com um segredo do cliente e fornece o ID e os escopos do cliente como evidência da autorização.

`scope`

Os escopos do OAuth 2.0 que representam as permissões de acesso do portador do token.

`cognito:groups`

As associações de um usuário ao grupo. Em um modelo de autorização baseado no controle de acesso baseado em funções (RBAC), essa declaração apresenta as funções que você pode avaliar em suas políticas.

Reivindicações transitórias

Declarações que não são uma permissão de acesso, mas são adicionadas em tempo de execução por um acionador Lambda de pré-geração de tokens do grupo de usuários. As reivindicações transitórias se assemelham às reivindicações padrão, mas estão fora do padrão,

por exemplo `tenant` ou. `department` A personalização dos tokens de acesso adiciona custo à sua AWS fatura.

Para obter mais informações sobre a estrutura dos tokens de acesso dos grupos de usuários do Amazon Cognito, consulte Uso do token de acesso no Guia do Desenvolvedor do Amazon Cognito.

Um token de acesso do Amazon Cognito é mapeado para um objeto de contexto quando transmitido para o Verified Permissions. Os atributos do token de acesso podem ser referenciados por meio de `context.token.`*`attribute_name`*. O exemplo de token de acesso a seguir inclui as reivindicações `client_id` e `scope`.

```
{
    "sub": "91eb4550-9091-708c-a7a6-9758ef8b6b1e",
    "cognito:groups": [
        "Store-Owner-Role",
        "Customer"
    ],
    "iss": "https://cognito-idp.us-east-2.amazonaws.com/us-east-2_EXAMPLE",
    "client_id": "1example23456789",
    "origin_jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "event_id": "bda909cb-3e29-4bb8-83e3-ce6808f49011",
    "token_use": "access",
    "scope": "MyAPI/mydata.write",
    "auth_time": 1688092966,
    "exp": 1688096566,
    "iat": 1688092966,
    "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN2222222",
    "username": "alice"
}
```

O exemplo a seguir mostra como refletir os atributos do exemplo de token de acesso no esquema do Verified Permissions. Para obter mais informações sobre a edição do esquema, consulte Edição de esquemas no modo 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"
          }
       }
    }
}
```

Após atualizar o esquema para refletir os atributos do Amazon Cognito, você poderá criar políticas que façam referência aos atributos.

```
permit(principal, action in [MyApplication::Action::"Read",
 MyApplication::Action::"GetStoreInventory"], resource)
when {
    context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
```

```
    context.token.scope.contains("MyAPI/mydata.write")
};
```

Mapeando tokens de acesso do OIDC

A maioria dos tokens de acesso de provedores externos do OIDC se alinha estreitamente aos tokens de acesso do Amazon Cognito. Um token de acesso OIDC é mapeado para um objeto de contexto quando passado para Permissões verificadas. Os atributos do token de acesso podem ser referenciados por meio de `context.token.`*`attribute_name`*. O exemplo de token de acesso OIDC a seguir inclui exemplos de declarações básicas.

```
{
    "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"
}
```

O exemplo a seguir mostra como refletir os atributos do exemplo de token de acesso no esquema do Verified Permissions. Para obter mais informações sobre a edição do esquema, consulte Edição de esquemas no modo 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"
          }
       }
    }
}
```

Depois de atualizar seu esquema para refletir os atributos do IdP, você pode criar políticas que façam referência aos atributos.

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

```
    context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
    context.token.scope.contains("MyAPI-read")
};
```

Notação alternativa para declarações delimitadas por dois pontos do Amazon Cognito

No momento em que as Permissões verificadas foram lançadas, o esquema recomendado para o token do Amazon Cognito afirma ser `cognito:groups` semelhante `custom:store` e converteu essas cadeias de caracteres delimitadas por dois pontos para usar o caractere como delimitador de hierarquia. `.` Esse formato é chamado de notação de pontos. Por exemplo, uma referência a `cognito:groups` tornou-se `principal.cognito.groups` em suas políticas. Embora você possa continuar usando esse formato, recomendamos que você crie seu esquema e suas políticas com a notação de colchetes. Nesse formato, uma referência a se `cognito:groups` torna `principal["cognito:groups"]` em suas políticas. Os esquemas gerados automaticamente para tokens de ID do grupo de usuários do console de permissões verificadas usam a notação de colchetes.

Você pode continuar usando a notação de pontos em esquemas e políticas criados manualmente para fontes de identidade do Amazon Cognito. Você não pode usar a notação de pontos com `:` ou quaisquer outros caracteres não alfanuméricos no esquema ou nas políticas para qualquer outro tipo de IdP do OIDC.

Um esquema para notação de pontos agrupa cada instância de um `:` caractere como filha da frase `custom` inicial `cognito` ou da frase, conforme mostrado no exemplo a seguir:

```
"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
         }
      }
   }
}
```

Com um esquema nesse formato, você pode criar uma política com notação de pontos, como no exemplo a seguir:

```
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"
};
```

Criação de um modelo de autorização para sua aplicação

Ao se preparar para usar o serviço Amazon Verified Permissions em um software, possivelmente será difícil começar a escrever declarações de política imediatamente. Isso seria semelhante a iniciar o desenvolvimento de outras partes de uma aplicação escrevendo instruções SQL ou especificações de API antes de decidir completamente o que a aplicação deveria fazer. Em vez disso, você deve começar com uma experiência de usuário, buscando compreender claramente o que os usuários finais verão ao gerenciar permissões na interface do usuário da aplicação. Em seguida, trabalhe com base nessa experiência para chegar a uma abordagem de implementação.

Ao realizar esse trabalho, você se perguntará, por exemplo:

- Quais são meus recursos? Eles se relacionam entre si? Por exemplo, os arquivos residem em uma pasta?
- Quais ações as entidades principais podem realizar em cada recurso?
- Como as entidades principais adquirem essas permissões?
- Você quer que seus usuários finais escolham entre permissões predefinidas, como “Administrador”, “Operador” ou “”, ou eles deveriam criar ReadOnly declarações de política ad-hoc? Ou ambos?
- As permissões devem ser herdadas entre recursos, como arquivos que herdam permissões de uma pasta pai?
- Quais tipos de consultas são necessárias para renderizar a experiência do usuário? Por exemplo, você precisa listar todos os recursos que uma entidade principal pode acessar para renderizar a página inicial desse usuário?
- Os usuários podem se privar acidentalmente de seus próprios recursos? Isso precisa ser evitado?

O resultado final desse exercício é chamado de modelo de autorização; ele define as entidades principais, os recursos, as ações e como eles se relacionam entre si. A produção desse modelo não requer conhecimento exclusivo do Cedar ou do serviço Verified Permissions. Ela é antes de mais nada um exercício de design de experiência do usuário, como qualquer outro, e pode se manifestar em artefatos como modelos de interface, diagramas lógicos e uma descrição geral de como as permissões influenciam o que os usuários veem no produto. O Cedar foi projetado para ser flexível o suficiente para atender aos clientes em um modelo, em vez de forçar o modelo a se curvar de forma anormal para estar em conformidade com a implementação do Cedar. Como resultado, obter

uma compreensão precisa da experiência desejada do usuário é a melhor maneira de chegar a um modelo ideal.

Esta seção fornece orientação geral sobre como abordar o exercício de design, pontos a serem observados e uma série de práticas recomendadas para usar o Verified Permissions com sucesso.

Além das diretrizes apresentadas aqui, lembre-se de considerar as práticas recomendadas no Guia de referência da linguagem de política Cedar.

Tópicos

Não existe um modelo canônico “correto”

Quando você cria um modelo de autorização, não há uma resposta única e exclusivamente correta. Aplicações diferentes podem usar diferentes modelos de autorização para conceitos semelhantes, e está tudo certo. Por exemplo, considere a representação do sistema de arquivos de um computador. Quando você cria um arquivo em um sistema operacional semelhante ao UNIX, ele não herda automaticamente as permissões da pasta principal. Por outro lado, em muitos outros sistemas operacionais e na maioria dos serviços de compartilhamento de arquivos on-line, os arquivos herdam as permissões de sua pasta pai. Ambas as opções são válidas, dependendo das circunstâncias da otimização da aplicação.

A exatidão de uma solução de autorização não é absoluta, mas o que deve ser avaliado é se ela oferece a experiência que seus clientes desejam e protege os recursos da maneira esperada. Se o seu modelo de autorização cumprir isso, ele será bem-sucedido.

É por isso que começar seu design com a experiência de usuário desejada é o pré-requisito mais importante para a criação de um modelo de autorização eficaz.

Concentre-se em seus recursos além das operações de API

Na maioria das aplicações voltadas para o consumidor, as permissões são modeladas com base nos recursos compatíveis com a aplicação. Por exemplo, uma aplicação de compartilhamento de arquivos pode representar permissões como ações que podem ser executadas em um arquivo ou uma pasta. Esse é um modelo bom e simples, que abstrai a implementação subjacente e as operações de API de back-end.

Por outro lado, outros tipos de aplicações, especialmente os serviços web, geralmente criam permissões com base nas próprias operações de API. Por exemplo, se um serviço web fornece uma API chamada `createThing()`, o modelo de autorização pode definir uma permissão correspondente ou uma `action` no Cedar chamada `createThing`. Isso funciona em muitas situações e facilita a compreensão das permissões. Para invocar a operação `createThing`, você precisa da permissão de ação `createThing`. Parece simples, não é?

Você descobrirá que o processo de introdução no console de Permissões verificadas inclui a opção de criar seus recursos e ações diretamente de uma API. Essa é uma linha de base útil: um mapeamento direto entre seu repositório de políticas e a API que ele autoriza.

No entanto, essa abordagem com foco na API pode não ser ideal, pois as APIs são apenas um proxy para o que seus clientes estão tentando proteger: os dados e os recursos subjacentes. Se várias APIs controlam o acesso aos mesmos recursos, pode ser difícil para os administradores determinar os caminhos que levam a esses recursos e gerenciar o acesso adequadamente.

Por exemplo, considere um diretório de usuários que contém os membros de uma organização. Os usuários podem ser organizados em grupos, e uma das metas de segurança é proibir a detecção de associações a grupos por partes não autorizadas. O serviço que gerencia esse diretório de usuários fornece duas operações de API:

- `listMembersOfGroup`
- `listGroupMembershipsForUser`

Os clientes podem usar qualquer uma dessas operações para detectar a associação a grupos. Portanto, o administrador de permissões deve se lembrar de coordenar o acesso às duas operações.

Isso será ainda mais complicado se você optar posteriormente por adicionar uma nova operação de API para tratar de outros casos de uso, como os seguintes.

- `isUserInGroups` (uma nova API para testar rapidamente se um usuário pertence a um ou mais grupos)

Do ponto de vista da segurança, essa API abre um terceiro caminho para a detecção de associações a grupos, interrompendo as permissões cuidadosamente elaboradas pelo administrador.

Recomendamos que você ignore a semântica da API e, em vez disso, se concentre nos dados e recursos subjacentes e em suas operações de associação. A aplicação dessa abordagem ao exemplo de associação a grupos resultaria em uma permissão abstrata, como `viewGroupMembership`, que cada uma das três operações de API deve consultar.

Nome da API	Permissões	
`listMembersOfGroup`	requer a permissão `viewGroupMembership`	no grupo
`listGroupMembershipsForUser`	requer a permissão `viewGroupMembership`	no usuário
`isUserInGroups`	requer a permissão `viewGroupMembership`	no usuário

Ao definir essa permissão, o administrador controlará perpetuamente o acesso à detecção de associações a grupos. Em contrapartida, agora, cada operação de API deve documentar as possíveis várias permissões necessárias, e o administrador deve consultar essa documentação ao criar as permissões. Essa pode ser uma compensação válida quando necessário para atender aos seus requisitos de segurança.

A autorização composta é normal

A autorização composta ocorre quando uma única atividade de usuário, como clicar em um botão na interface da aplicação, requer várias consultas de autorização individuais para determinar se essa atividade é permitida. Por exemplo, mover um arquivo para um novo diretório em um sistema de arquivos pode exigir três permissões diferentes: a capacidade de excluir um arquivo do diretório de origem, a capacidade de adicionar um arquivo ao diretório de destino e, possivelmente, a capacidade de tocar no próprio arquivo (dependendo da aplicação).

Se você é iniciante na criação de um modelo de autorização, talvez pense que cada decisão de autorização deve ser resolvida em uma única consulta de autorização. Mas isso pode resultar em modelos excessivamente complexos e declarações de política complicadas. Na prática, o uso de autorizações compostas pode ser útil para ajudar você a produzir um modelo de autorização mais simples. Uma das medidas de um modelo de autorização bem projetado é que, quando você tem ações individuais suficientemente decompostas, suas operações compostas, como mover um arquivo, podem ser representadas por uma agregação intuitiva de primitivas.

Outra situação em que a autorização composta ocorre é quando várias partes estão envolvidas no processo de concessão de uma permissão. Considere um diretório organizacional em que os usuários possam ser membros de grupos. Uma abordagem simples é dar permissão ao proprietário do grupo para adicionar qualquer pessoa. Mas, e se você quiser que seus usuários primeiro deem consentimento antes serem adicionados? Isso introduz um acordo de handshake no qual tanto o usuário quanto o grupo devem consentir com a associação. Para fazer isso, você pode introduzir outra permissão vinculada ao usuário e especificar se o usuário pode ser adicionado a qualquer grupo ou a um grupo específico. Quando um chamador tentar adicionar membros a um grupo posteriormente, a aplicação deverá aplicar os dois lados das permissões: que o chamador tenha permissão para adicionar membros ao grupo especificado e que o usuário individual que está sendo adicionado tenha as permissões para ser adicionado. Quando existem handshakes de N direções, é comum observar N consultas de autorização composta para aplicar cada parte do contrato.

Se você se deparar com um desafio de design em que há vários recursos envolvidos e não estiver claro como modelar as permissões, isso poderá ser indício de que você tem um cenário de autorização composta. Nesse caso, a solução pode ser a decomposição da operação em várias verificações de autorização individuais.

Considerações sobre multilocação

Talvez você queira desenvolver aplicativos para uso por vários clientes — empresas que consomem seu aplicativo ou locatários — e integrá-los às Permissões Verificadas da Amazon. Antes de desenvolver seu modelo de autorização, desenvolva uma estratégia multilocatária. Você pode gerenciar as políticas de seus clientes em um repositório de políticas compartilhado ou atribuir a cada um um repositório de políticas por inquilino.

1. Um repositório de políticas compartilhado

   Todos os inquilinos compartilham um único repositório de apólices. O aplicativo envia todas as solicitações de autorização para o repositório de políticas compartilhadas.

2. Armazenamento de políticas por inquilino

   Cada inquilino tem um repositório de políticas dedicado. O aplicativo consultará diferentes repositórios de políticas para obter uma decisão de autorização, dependendo do inquilino que fizer a solicitação.

Nenhuma das estratégias cria um volume relativamente maior de solicitações de autorização que podem ter um impacto na sua fatura. AWS Então, como você deve projetar sua abordagem? Veja a seguir condições comuns que podem contribuir para sua estratégia de autorização de multilocação de Permissões Verificadas.

Isolamento das políticas do inquilino

O isolamento das políticas de cada inquilino das demais é importante para proteger os dados do inquilino. Quando cada inquilino tem seu próprio repositório de políticas, cada um tem seu próprio conjunto isolado de políticas.

Fluxo de autorização

Você pode identificar um inquilino fazendo uma solicitação de autorização com um ID do repositório de políticas na solicitação, com repositórios de políticas por inquilino. Com um repositório de políticas compartilhado, todas as solicitações usam o mesmo ID do repositório de políticas.

Gerenciamento de modelos e esquemas

Seus modelos de políticas e um esquema de armazenamento de políticas adicionam um nível de sobrecarga de design e manutenção em cada repositório de políticas.

Gerenciamento de políticas globais

Talvez você queira aplicar algumas políticas globais a cada inquilino. O nível de sobrecarga do gerenciamento de políticas globais varia entre os modelos de armazenamento de políticas compartilhados e por inquilino.

Desembarque do inquilino

Alguns inquilinos contribuirão com elementos para seu esquema e políticas que são específicos para o caso deles. Quando um inquilino não está mais ativo na sua organização e você deseja remover seus dados, o nível de esforço varia de acordo com o nível de isolamento de outros inquilinos.

Cotas de recursos de serviço

O Verified Permissions tem cotas de recursos e taxas de solicitação que podem influenciar sua decisão de multilocação. Para obter mais informações sobre cotas, consulte Cotas para recursos.

Comparando repositórios de políticas compartilhados e repositórios de políticas por inquilino

Cada consideração exige seu próprio nível de comprometimento de tempo e recursos em modelos de repositório de políticas compartilhados e por inquilino.

Consideração	Nível de esforço em um repositório de políticas compartilhado	Nível de esforço em repositórios de políticas por inquilino
Isolamento das políticas do inquilino	Médio. Must include tenant identifiers in policies and authorization requests.	Baixo. Isolation is default behavior. Tenant-specific policies are inaccessible to other tenants.
Fluxo de autorização	Baixo. All queries target one policy store.	Médio. Must maintain mappings between each tenant and their policy store ID.
Gerenciamento de modelos e esquemas	Baixo. Must make one schema work for all tenants.	Alta. Schemas and templates might be less complex individually, but changes require more coordination and complexity.
Gerenciamento de políticas globais	Baixo. All policies are global and can be centrally updated.	Alta. You must add global policies to each policy store in onboarding. Replicate global policy updates between many policy stores.

Desembarque do inquilino	Médio. Must identify and delete only tenant-specific policies.	Baixo. Delete the policy store.
Cotas de recursos de serviço	Alta. Tenants share resource quotas that affect policy stores like schema size, policy size per resource, and identity sources per policy store.	Baixo. Each tenant has dedicated resource quotas.

Como escolher

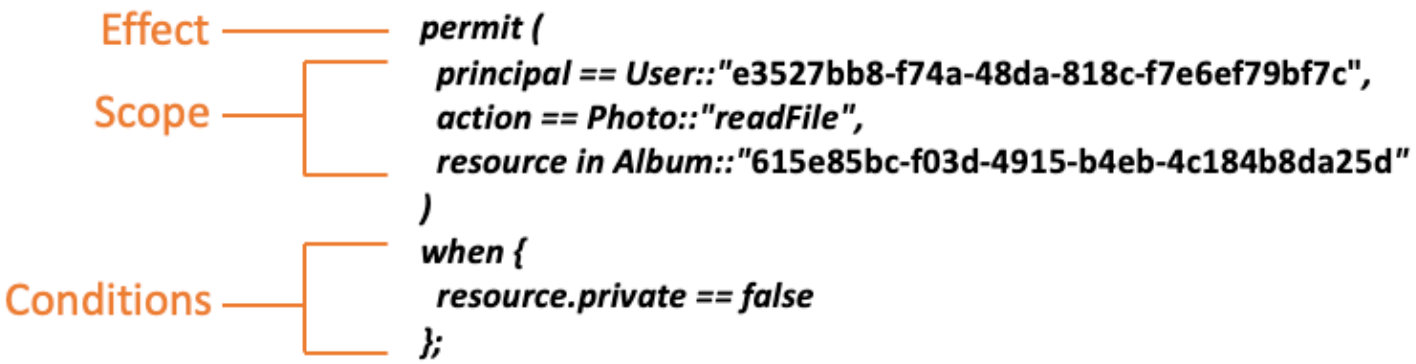
Cada aplicativo multilocatário é diferente. Compare cuidadosamente as duas abordagens e suas considerações antes de tomar uma decisão arquitetônica.

Se seu aplicativo não exigir políticas específicas para inquilinos e usar uma única fonte de identidade, um repositório de políticas compartilhado para todos os locatários provavelmente será a solução mais eficaz. Isso resulta em um fluxo de autorização mais simples e no gerenciamento global de políticas. Excluir um inquilino usando um repositório de políticas compartilhadas exige menos esforço porque o aplicativo não precisa excluir políticas específicas do inquilino.

Mas se seu aplicativo exigir muitas políticas específicas para inquilinos ou usar várias fontes de identidade, é provável que os armazenamentos de políticas por locatário sejam mais eficazes. Você pode controlar o acesso às políticas de inquilino com IAM políticas que concedem permissões por inquilino a cada repositório de políticas. Excluir um inquilino envolve a exclusão de seu repositório de políticas; em um shared-policy-store ambiente, você deve encontrar e excluir políticas específicas do inquilino.

Quando possível, preencha o escopo da política

O escopo da política é a parte de uma declaração de política do Cedar após as palavras-chave `permit` ou `forbid` e entre os parênteses de abertura.



É recomendável que você preencha os valores para `principal` e `resource` sempre que possível. Isso permite que o Verified Permissions indexe as políticas para uma recuperação mais eficiente e, portanto, melhore o desempenho. Se você precisar conceder as mesmas permissões a várias entidades principais ou recursos diferentes, recomendamos que você utilize um modelo de política e o anexe a cada par de entidade principal/recurso.

Evite criar uma política grande que contenha listas de entidades principais e recursos em uma cláusula `when`. Isso provavelmente fará com que você se depare com limites de escalabilidade ou desafios operacionais. Por exemplo, para adicionar ou remover um único usuário de uma lista grande em uma política, é necessário ler toda a política, editar a lista, escrever a nova política na íntegra e lidar com erros de simultaneidade se um administrador substituir as alterações de outro. Por outro lado, com o uso de várias permissões refinadas, adicionar ou remover um usuário é tão simples quanto adicionar ou remover a política que se aplica a elas.

Todos os recursos residem em um contêiner

Quando você cria um modelo de autorização, cada ação deve estar associada a um recurso específico. Com uma ação como `viewFile`, o recurso ao qual você pode aplicá-la é intuitivo: um arquivo individual ou talvez uma coleção de arquivos em uma pasta. No entanto, uma operação como `createFile` é menos intuitiva. Ao modelar a capacidade de criar um arquivo, a qual recurso ele se aplica? Não pode ser ao arquivo em si, pois o arquivo ainda não existe.

Esse é um exemplo do problema generalizado da criação de recursos. A criação de recursos é um problema de bootstrapping. Deve haver alguma maneira de conceder permissão para criar recursos mesmo quando esses recursos ainda não existirem. A solução é reconhecer que cada recurso deve existir em algum contêiner, e é o próprio contêiner que atua como ponto de ancoragem para as permissões. Por exemplo, se uma pasta já existe no sistema, a capacidade de criar um arquivo pode

ser modelada como uma permissão nessa pasta, pois esse é o local em que as permissões precisam estar para instanciar o novo recurso.

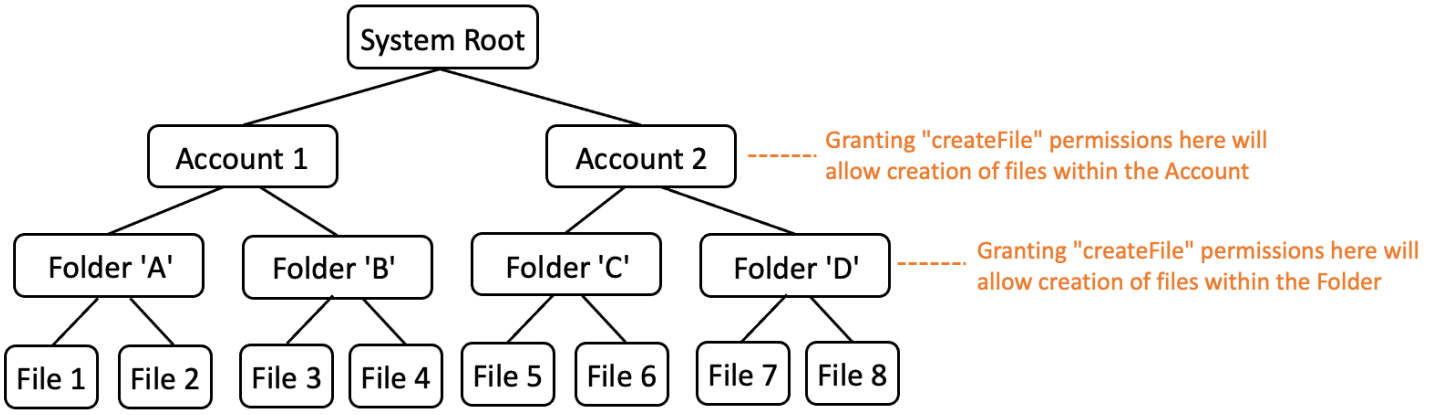
```
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    action == Action::"createFile",
    resource == Folder::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

Mas e se não houver nenhuma pasta? Talvez se trate de uma nova conta de cliente em uma aplicação na qual ainda não existam recursos. Nesse caso, ainda há um contexto que pode ser entendido intuitivamente por meio da pergunta: onde o cliente pode criar novos arquivos? Você não quer que eles criem arquivos em uma conta de cliente aleatória. Em vez disso, há um contexto implícito: o limite da conta do próprio cliente. Portanto, a própria conta representa o contêiner para criação de recursos, e isso pode ser explicitamente modelado em uma política semelhante ao exemplo a seguir.

```
// 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"
);
```

Mas, e se não houver contas? Você pode optar por criar o fluxo de trabalho de inscrição do cliente para que ele crie novas contas no sistema. Nesse caso, você precisará de um contêiner para manter o limite externo no qual o processo pode criar as contas. Esse contêiner de nível raiz representa o sistema como um todo e pode ser chamado de “raiz do sistema”. No entanto, decidir se isso será necessário ou não e como nomeá-lo é de sua responsabilidade, o proprietário da aplicação.

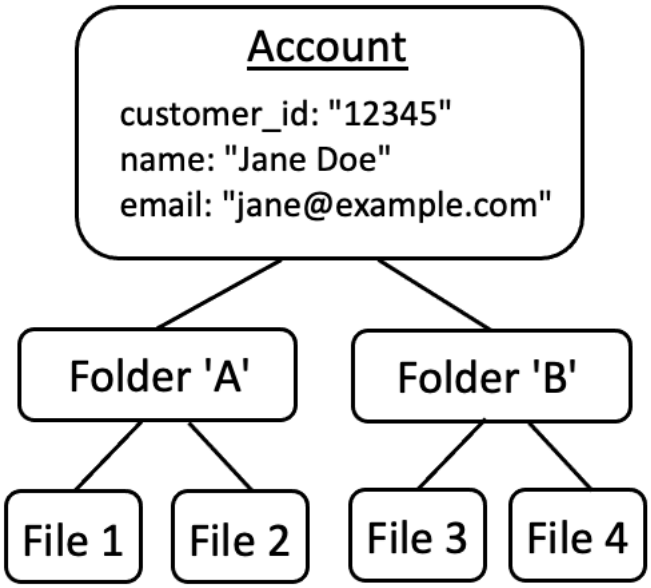
Nesse exemplo de aplicação, a hierarquia de contêineres resultante teria, portanto, a seguinte aparência:



Esse é um exemplo de hierarquia. Outros exemplos também são válidos. É importante lembrar que a criação de recursos sempre acontece no contexto de um contêiner de recursos. Esses contêineres podem estar implícitos, como um limite de conta, e podem ser facilmente ignorados. Ao projetar seu modelo de autorização, não deixe de observar essas suposições implícitas, para que elas possam ser formalmente documentadas e representadas no modelo de autorização.

Separe as entidades principais dos contêineres de recursos

Quando você está projetando uma hierarquia de recursos, uma das inclinações comuns, especialmente para aplicações voltadas para o consumidor, é usar a identidade do usuário do cliente como contêiner de recursos em uma conta de cliente.

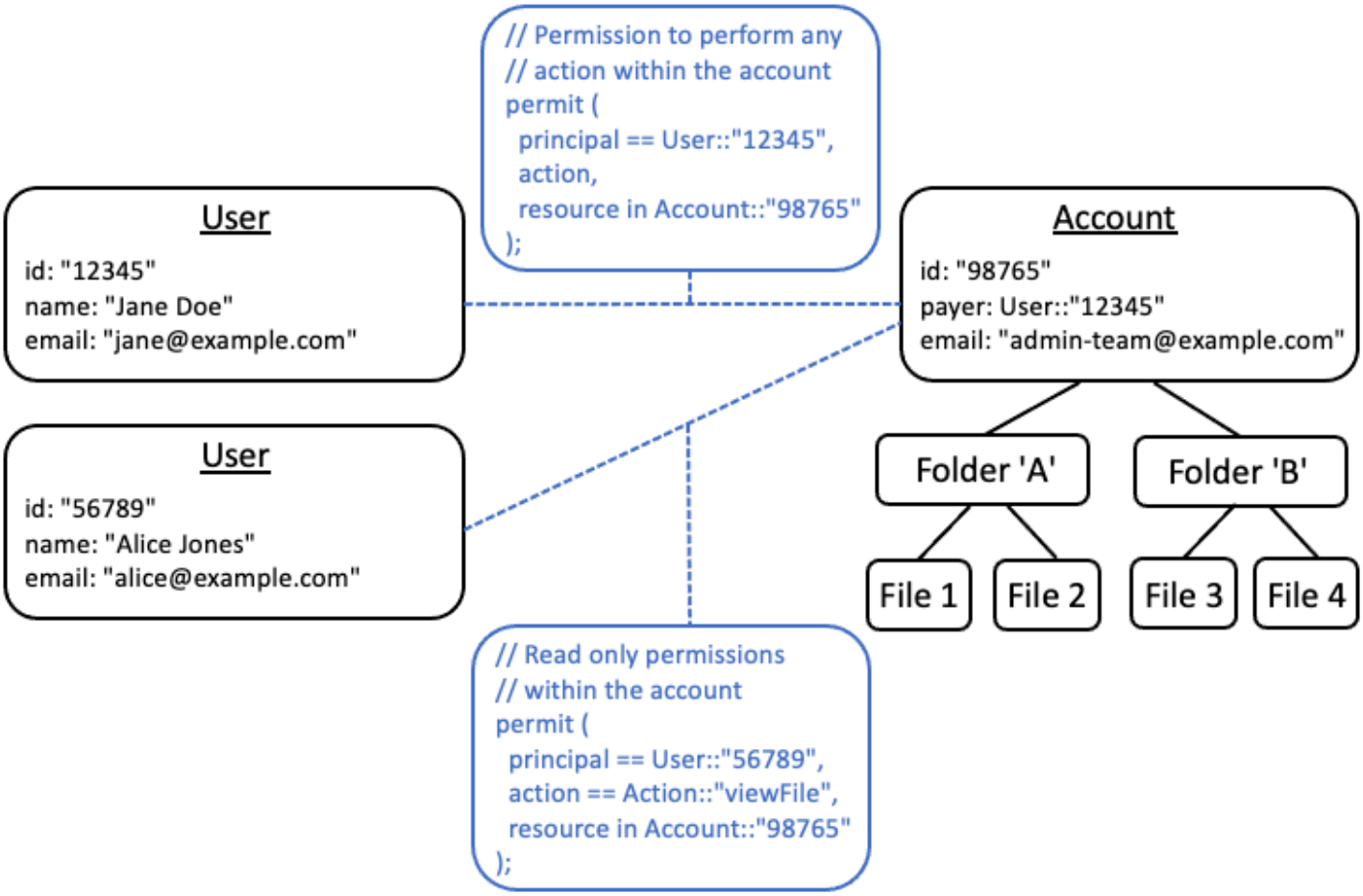


É recomendável que você trate essa estratégia como um antipadrão. Isso ocorre porque há uma tendência natural nas aplicações mais avançadas de delegar acesso a usuários adicionais. Por

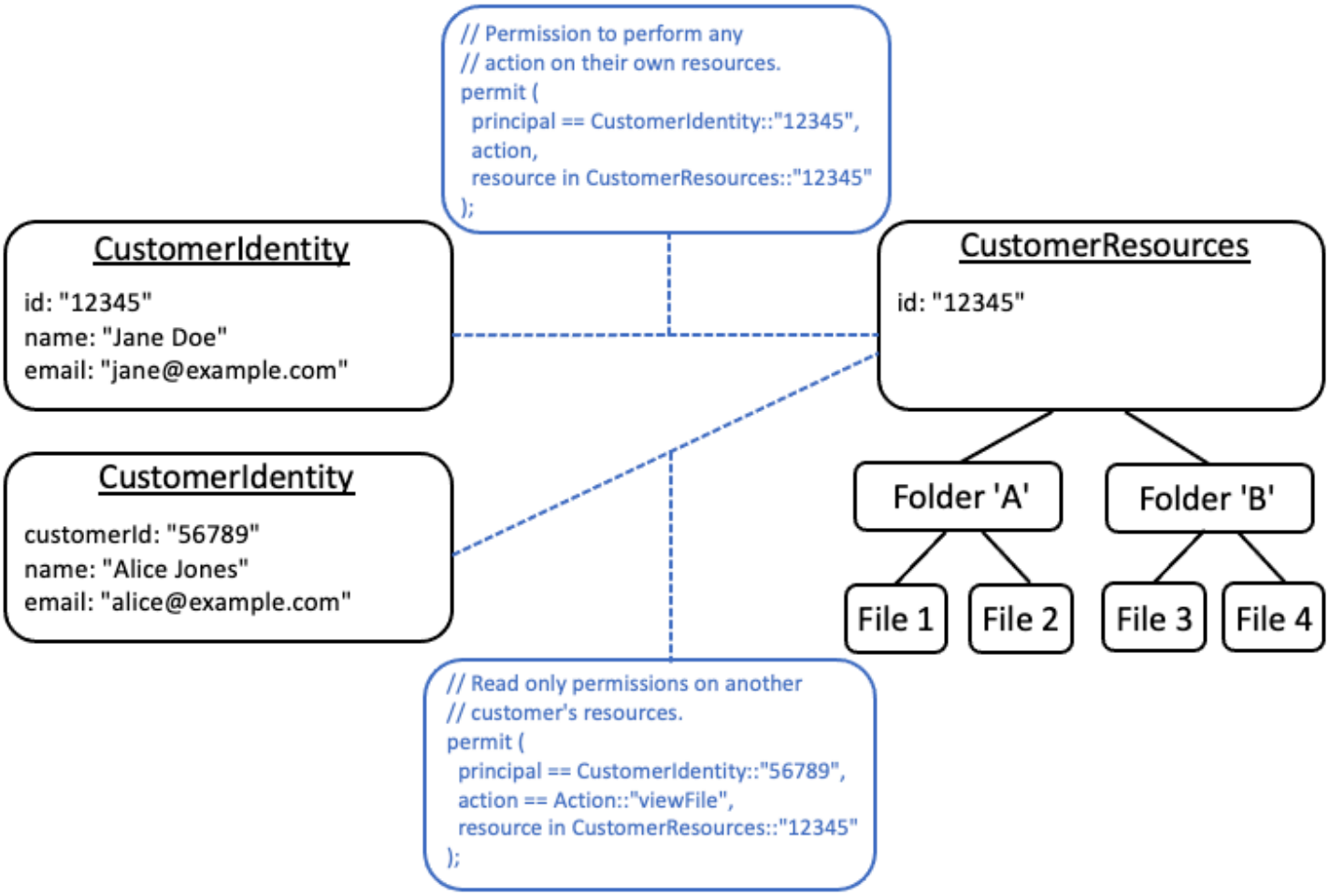
exemplo, você pode optar por introduzir contas “familiares”, nas quais outros usuários podem compartilhar recursos de conta. Da mesma forma, às vezes, os clientes corporativos desejam designar vários membros da força de trabalho como operadores para partes da conta. Talvez você também precise transferir a propriedade de uma conta para outro usuário ou mesclar os recursos de várias contas.

Quando uma identidade de usuário é usada como contêiner de recursos para uma conta, torna-se mais difícil obter os cenários anteriores. O mais alarmante é que, se outras pessoas tiverem acesso ao contêiner da conta nessa abordagem, elas poderão inadvertidamente receber acesso para modificar a própria identidade do usuário, como, por exemplo, alterar as credenciais de e-mail ou login de Jane.

Portanto, quando possível, uma abordagem mais resiliente é separar as entidades principais dos contêineres de recursos e modelar a conexão entre eles usando conceitos como “permissões de administrador” ou “propriedade”.



Quando há uma aplicação que não consegue seguir esse modelo desacoplado, recomendamos que você o simule o máximo possível ao criar um modelo de autorização. Por exemplo, uma aplicação que tenha um único conceito chamado `Customer`, que encapsula a identidade do usuário, as credenciais de login e os recursos que elas possuem, pode mapear isso para um modelo de autorização que contenha uma única entidade lógica para `Customer Identity` (contendo nome, e-mail etc.) e uma entidade lógica separada para `Customer Resources` ou `Customer Account`, atuando como nó pai de todos os recursos que elas possuem. Ambas as entidades podem compartilhar o mesmo `Id`, mas com um `Type` diferente.



Não incorpore permissões nos atributos

A melhor forma de utilizar os atributos são como entrada na decisão de autorização. Não use atributos para representar as permissões em si, por exemplo, declarando um atributo chamado “permittedFolders” em um Usuário:

```
// ANTI-PATTERN: comingling permissions into user attributes
```

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

E, posteriormente, usando o atributo em uma política:

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

Essa abordagem transforma o que seria um modelo de autorização simples, onde uma entidade principal específica tem acesso a uma pasta específica, em um modelo de controle de acesso por atributo (ABAC) com as desvantagens inerentes. Uma dessas desvantagens é que fica mais difícil determinar rapidamente quem tem permissão para um recurso. No exemplo anterior, para determinar quem tem acesso a uma pasta específica, é necessário iterar cada usuário para verificar se essa pasta está listada em seus atributos, tendo sempre em mente de que existe uma política que concede acesso em caso afirmativo.

Outro risco dessa abordagem são os fatores de escalabilidade quando as permissões são agrupadas em um único registro `User`. Se o usuário tiver acesso a muitos dados, o tamanho cumulativo do registro `User` aumentará e talvez se aproxime do limite máximo de qualquer sistema que esteja armazenando os dados.

Em vez disso, recomendamos que você represente esse cenário usando várias políticas individuais, talvez usando modelos de políticas para minimizar a repetição.

```
//BETTER PATTERN
permit (
```

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

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

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

O Verified Permissions pode lidar com eficiência com várias políticas individuais e refinadas durante a avaliação da autorização. Essa modelagem melhora a capacidade de gerenciamento e auditoria ao longo do tempo.

Prefira permissões refinadas no modelo e permissões agregadas na interface do usuário

Uma estratégia da qual os designers geralmente se arrependem é criar um modelo de autorização com ações muito amplas, como `Read` e `Write`, e perceber depois que são necessárias ações mais refinadas. A necessidade de maior granularidade pode ser motivada pelo feedback dos clientes sobre controles de acesso mais granulares ou por auditores de conformidade e segurança que incentivam permissões com privilégios mínimos.

Se as permissões refinadas não forem definidas antecipadamente, possivelmente será necessária uma conversão complicada para transformar o código da aplicação e as declarações de política em permissões de usuário mais refinadas. Por exemplo, o código de aplicação previamente autorizado com base em uma ação granulada precisará ser modificado para usar as ações refinadas. Além disso, as políticas precisarão ser atualizadas para refletir a migração:

```
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"
);
```

Para evitar essa migração dispendiosa, é melhor definir permissões refinadas com antecedência. No entanto, isso pode resultar em uma desvantagem se os usuários finais forem posteriormente forçados a entender um número maior de permissões refinadas, especialmente se a maioria dos clientes estiver satisfeita com controles granulados, como `Read` e `Write`. Para obter o melhor dos dois mundos, você pode agrupar permissões refinadas em coleções predefinidas, como `Read` e `Write`, usando mecanismos como modelos de políticas ou grupos de ação. Ao usar essa abordagem, os clientes veem somente as permissões granuladas. Mas, nos bastidores, você preparou sua aplicação para o futuro ao modelar as permissões granuladas como uma coleção de ações refinadas. Quando clientes ou auditores solicitam, as permissões refinadas podem ser expostas.

Considere outros motivos para consultar uma autorização

Normalmente, associamos as verificações de autorização às solicitações dos usuários. A verificação é uma forma de determinar se o usuário tem permissão para realizar essa solicitação. No entanto, você também pode usar dados de autorização para influenciar o design da interface da aplicação. Por exemplo, talvez você queira exibir uma tela inicial que mostre uma lista somente dos recursos que o usuário final pode acessar. Ao visualizar os detalhes de um recurso, talvez você queira que a interface mostre somente as operações que o usuário pode realizar nesse recurso.

Essas situações podem gerar desvantagens no modelo de autorização. Por exemplo, a forte dependência de políticas de controle de acesso por atributo (ABAC) pode dificultar a resposta rápida à pergunta “quem tem acesso a quê?” Isso ocorre porque, para responder a essa pergunta, é necessário examinar cada regra com base em cada entidade principal e recurso, a fim de determinar se há uma correspondência. Como resultado, um produto que precisa ser otimizado para listar somente os recursos acessíveis pelo usuário pode optar por usar um modelo de controle de acesso baseado em função (RBAC). Ao usar o RBAC, pode ser mais fácil iterar todas as políticas anexadas a um usuário para determinar o acesso aos recursos.

Banco de testes

O banco de testes do Verified Permissions permite testar e solucionar problemas de políticas do Verified Permissions executando solicitações de autorização com base nessas políticas. O banco de testes usa os parâmetros que você especifica para determinar se as políticas do Cedar em seu armazenamento de políticas autorizariam a solicitação. Você pode alternar entre o Modo Visual e o Modo JSON enquanto testa as solicitações de autorização. Para obter mais informações sobre como as políticas do Cedar são estruturadas e avaliadas, consulte Construção de políticas básicas no Cedar no Guia de referência da linguagem de política Cedar.

> **Note**
>
> Ao fazer uma solicitação de autorização usando o Verified Permissions, você pode fornecer a lista de entidades principais e recursos como parte da solicitação na seção Entidades adicionais. No entanto, não é possível incluir os detalhes sobre as ações. Eles devem ser especificados no esquema ou inferidos a partir da solicitação. Você não pode colocar uma ação na seção Entidades adicionais.

Para uma visão geral visual e uma demonstração da bancada de testes, veja este vídeo.

Visual mode

> **Note**
>
> Você deve ter um esquema definido em seu armazenamento de políticas para usar o Modo Visual do banco de testes.

Para testar políticas no modo Visual

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Banco de testes.
3. Escolha Modo visual.
4. Na seção Entidade principal, escolha a Entidade principal agindo entre os tipos de entidade principal do esquema. Digite um identificador para a entidade principal na caixa de texto.

5. (Opcional) Escolha Adicionar um pai para adicionar entidades pai para a entidade principal especificada. Para remover um pai adicionado à entidade principal, escolha Remover ao lado do nome do pai.
6. Especifique o Valor do atributo para cada atributo da entidade principal especificada. O banco de testes usa os valores de atributo especificados na solicitação de autorização simulada.
7. Na seção Recurso, escolha o Recurso no qual a entidade principal está executando uma ação. Digite um identificador para o recurso na caixa de texto.
8. (Opcional) Escolha Adicionar um pai para adicionar entidades pai para o recurso especificado. Para remover um pai adicionado ao recurso, escolha Remover ao lado do nome do pai.
9. Especifique o Valor do atributo para cada atributo do recurso especificado. O banco de testes usa os valores de atributo especificados na solicitação de autorização simulada.
10. Na seção Ação, escolha a Ação que a entidade principal está executando na lista de ações válidas para a entidade principal e o recurso especificados.
11. Especifique o Valor do atributo para cada atributo da ação especificada. O banco de testes usa os valores de atributo especificados na solicitação de autorização simulada.
12. (Opcional) Na seção Entidades adicionais, escolha Adicionar entidade para adicionar entidades a serem avaliadas na decisão de autorização.
13. Escolha o Identificador de entidade na lista suspensa e digite o identificador da entidade.
14. (Opcional) Escolha Adicionar um pai para adicionar entidades pai para a entidade especificada. Para remover um pai adicionado à entidade, escolha Remover ao lado do nome do pai.
15. Especifique o Valor do atributo para cada atributo da entidade especificada. O banco de testes usa os valores de atributo especificados na solicitação de autorização simulada.
16. Escolha Confirmar para adicionar a entidade ao banco de testes.
17. Escolha Executar solicitação de autorização para simular a solicitação de autorização para as políticas do Cedar no armazenamento de políticas. O banco de testes exibe a decisão de permitir ou negar a solicitação juntamente com informações sobre as políticas atendidas ou os erros encontrados durante a avaliação.

JSON mode

Para testar políticas no modo JSON

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Banco de testes.
3. Escolha Modo JSON.
4. Na seção Detalhes da solicitação, se você tiver um esquema definido, escolha a Entidade principal agindo entre os tipos de entidade principal do esquema. Digite um identificador para a entidade principal na caixa de texto.

   Se você não tiver um esquema definido, digite a entidade principal na caixa de texto Entidade principal agindo.
5. Se você tiver um esquema definido, escolha o Recurso entre os tipos de recurso do esquema. Digite um identificador para o recurso na caixa de texto.

   Se você não tiver um esquema definido, digite o recurso na caixa de texto Recurso.
6. Se você tiver um esquema definido, escolha a Ação na lista de ações válidas para a entidade principal e o recurso especificados.

   Se você não tiver um esquema definido, digite a ação na caixa de texto Ação.
7. Insira o contexto da solicitação a ser simulada no campo Contexto. O contexto da solicitação é uma informação adicional que pode ser usada nas decisões de autorização.
8. No campo Entidades, insira a hierarquia das entidades e respectivos atributos a serem avaliados na decisão de autorização.
9. Escolha Executar solicitação de autorização para simular a solicitação de autorização para as políticas do Cedar no armazenamento de políticas. O banco de testes exibe a decisão de permitir ou negar a solicitação juntamente com informações sobre as políticas atendidas ou os erros encontrados durante a avaliação.

Implementando a autorização nas permissões verificadas da Amazon

Depois de criar seu repositório de políticas, políticas, modelos, esquema e modelo de autorização, você estará pronto para começar a autorizar solicitações usando as Permissões Verificadas da Amazon. Para implementar a autorização de permissões verificadas, você deve combinar a configuração das políticas AWS com a integração em um aplicativo. Para integrar as Permissões Verificadas ao seu aplicativo, adicione um AWS SDK e implemente os métodos que invocam a API de Permissões Verificadas e geram decisões de autorização em relação ao seu repositório de políticas.

A autorização com permissões verificadas é útil para permissões de UX e permissões de API em seus aplicativos.

Permissões de UX

> Controle o acesso do usuário à UX do seu aplicativo. Você pode permitir que um usuário visualize somente os formulários, botões, gráficos e outros recursos exatos que ele precisa acessar. Por exemplo, quando um usuário faz login, talvez você queira determinar se o botão “Transferir fundos” está visível na conta dele. Você também pode controlar as ações que um usuário pode realizar. Por exemplo, no mesmo aplicativo bancário, talvez você queira determinar se seu usuário tem permissão para alterar a categoria de uma transação.

Permissões de API

> Controle o acesso do usuário aos dados. Os aplicativos geralmente fazem parte de um sistema distribuído e trazem informações de APIs externas. No exemplo do aplicativo bancário em que as Permissões Verificadas permitiram a exibição do botão “Transferir fundos”, uma decisão de autorização mais complexa deve ser tomada quando o usuário inicia uma transferência. As permissões verificadas podem autorizar a solicitação de API que lista as contas de destino que são alvos de transferência elegíveis e, em seguida, a solicitação para enviar a transferência para a outra conta.

Os exemplos que ilustram esse conteúdo vêm de um exemplo de armazenamento de políticas. Para acompanhar, crie o DigitalPetrepositório de políticas de amostra da Store em seu ambiente de teste.

Para um exemplo completo de aplicativo que implementa permissões de UX usando autorização em lote, consulte Use Amazon Verified Permissions para obter autorizações detalhadas em grande escala no Security Blog.AWS

Operações de API para autorização

A API de permissões verificadas tem as seguintes operações de autorização.

IsAuthorized

A operação `IsAuthorized` da API é o ponto de entrada para solicitações de autorização com permissões verificadas. Você deve enviar elementos principais, de ação, de recursos, de contexto e de entidades. As permissões verificadas validam as entidades em sua solicitação em relação ao seu esquema de armazenamento de políticas. Em seguida, as Permissões verificadas avaliam sua solicitação em relação a todas as políticas no repositório de políticas solicitado que se aplicam às entidades na solicitação.

IsAuthorizedWithToken

A `IsAuthorizedWithToken` operação gera uma solicitação de autorização dos dados do usuário nos tokens web JSON (JWTs) do Amazon Cognito. As permissões verificadas funcionam diretamente com o Amazon Cognito como uma fonte de identidade em seu repositório de políticas. As permissões verificadas preenchem todos os atributos do principal em sua solicitação a partir das declarações no ID do usuário ou nos tokens de acesso. Você pode autorizar ações e recursos a partir de atributos de usuário ou associação a grupos em um grupo de usuários do Amazon Cognito.

Você não pode incluir informações sobre os principais tipos de grupos ou usuários em uma `IsAuthorizedWithToken` solicitação. Você deve preencher todos os dados principais do JWT que você fornece.

BatchIsAutorizado

A `BatchIsAuthorized` operação processa várias decisões de autorização para um único principal ou recurso em uma única solicitação de API. Essa operação agrupa as solicitações em uma única operação em lote que minimiza o uso da cota e retorna as decisões de autorização para cada uma das até 30 ações aninhadas complexas. Com a autorização em lote para um único recurso, você pode filtrar as ações que um usuário pode realizar em um recurso. Com a autorização em lote para um único principal, você pode filtrar os recursos sobre os quais um usuário pode agir.

BatchIsAuthorizedWithSímbolo

A `BatchIsAuthorizedWithToken` operação processa várias decisões de autorização para um único principal em uma solicitação de API. O principal é fornecido pela fonte de identidade do seu repositório de políticas em um ID ou token de acesso. Essa operação agrupa as solicitações em uma única operação em lote que minimiza o uso da cota e retorna as decisões de autorização para cada uma das até 30 solicitações de ações e recursos. Em suas políticas, você pode autorizar o acesso deles a partir de seus atributos ou de sua associação a um grupo de usuários do Amazon Cognito.

Por exemplo`IsAuthorizedWithToken`, você não pode incluir informações sobre os principais tipos de grupos ou usuários em uma `BatchIsAuthorizedWithToken` solicitação. Você deve preencher todos os dados principais do JWT que você fornece.

Testando seu modelo de autorização

Para entender o efeito da decisão de autorização de Permissões Verificadas ao implantar seu aplicativo, você pode avaliar suas políticas à medida que as desenvolve com Banco de testes e com as solicitações da API REST HTTPS para Permissões Verificadas. A bancada de testes é uma ferramenta AWS Management Console para avaliar solicitações e respostas de autorização em seu repositório de políticas.

A API REST de permissões verificadas é a próxima etapa em seu desenvolvimento à medida que você passa da compreensão conceitual para o design do aplicativo. A API de permissões verificadas aceita solicitações de autorização com IsAuthorizedIsAuthorizedWithToken, e solicitações de AWS API BatchIsautorizadas como assinadas para endpoints de serviços regionais. Para testar seu modelo de autorização, você pode gerar solicitações com qualquer cliente de API e verificar se suas políticas estão retornando as decisões de autorização conforme o esperado.

Por exemplo, você pode testar `IsAuthorized` em um repositório de políticas de amostra com o procedimento a seguir.

Test bench

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Crie um repositório de políticas a partir do repositório de políticas de amostra com o nome DigitalPetArmazenamento.
2. Selecione Test bench em seu novo repositório de políticas.

3. Preencha sua solicitação de bancada de testes IsAuthorizedna referência da API de permissões verificadas. Os detalhes a seguir replicam as condições no Exemplo 4 que faz referência à amostra DigitalPetStore.

   a. Defina Alice como diretora. Em Principal tomando medidas, escolha `DigitalPetStore::User` e insira`Alice`.

   b. Defina o papel de Alice como cliente. Escolha Adicionar um pai`DigitalPetStore::Role`, escolha e insira Cliente.

   c. Defina o recurso como pedido “1234”. Em Recurso sobre o qual o diretor está atuando, escolha `DigitalPetStore::Order` e insira`1234`.

   d. O `DigitalPetStore::Order` recurso requer um `owner` atributo. Defina Alice como proprietária do pedido. Escolha `DigitalPetStore::User` e entre `Alice`

   e. Alice pediu para ver o pedido. Para Ação que o diretor está tomando, escolha`DigitalPetStore::Action::"GetOrder"`.

4. Escolha Executar solicitação de autorização. Em um repositório de políticas não modificado, essa solicitação resulta em uma `ALLOW` decisão. Observe a política Satisfied que retornou a decisão.

5. Escolha Políticas na barra de navegação à esquerda. Revise a política estática com a descrição Customer Role - Get Order.

6. Observe que as Permissões Verificadas permitiram a solicitação porque o diretor estava na função de cliente e era o proprietário do recurso.

REST API

1. Abra o console do Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/. Crie um repositório de políticas a partir do repositório de políticas de amostra com o nome DigitalPetArmazenamento.

2. Anote o ID do repositório de políticas do seu novo repositório de políticas.

3. IsAuthorizedNa referência da API de permissões verificadas, copie o corpo da solicitação do Exemplo 4 que faz referência à amostra da DigitalPetStore.

4. Abra seu cliente de API e crie uma solicitação para o endpoint de serviço regional para seu repositório de políticas. Preencha os cabeçalhos conforme mostrado no exemplo.

5. Cole o corpo da solicitação de amostra e altere o valor `policyStoreId` para o ID do repositório de políticas que você anotou anteriormente.

6. Envie a solicitação e analise os resultados. Em um DigitalPetrepositório de políticas de armazenamento padrão, essa solicitação retorna uma `ALLOW` decisão.

Você pode fazer alterações nas políticas, no esquema e nas solicitações em seu ambiente de teste para alterar os resultados e produzir decisões mais complexas.

1. Altere a solicitação de uma forma que altere a decisão das Permissões verificadas. Por exemplo, altere o papel de Alice para `Employee` ou altere o `owner` atributo da ordem 1234 para. `Bob`
2. Altere as políticas de forma que afetem as decisões de autorização. Por exemplo, modifique a política com a descrição Customer Role - Get Order para remover a condição de que ele `User` deve ser o proprietário do `Resource` e modifique a solicitação para que `Bob` ele queira visualizar o pedido.
3. Altere o esquema para permitir que as políticas tomem uma decisão mais complexa. Atualize as entidades solicitadas para que Alice possa atender aos novos requisitos. Por exemplo, edite o esquema para `User` permitir que você seja membro de `ActiveUsers` ou`InactiveUsers`. Atualize a política para que somente usuários ativos possam ver seus próprios pedidos. Atualize as entidades da solicitação para que Alice seja uma usuária ativa ou inativa.

Integração com aplicativos e AWS SDKs

Para implementar as Permissões Verificadas da Amazon em seu aplicativo, você deve definir as políticas e o esquema que deseja que seu aplicativo aplique. Com seu modelo de autorização estabelecido e testado, sua próxima etapa é começar a gerar solicitações de API a partir do ponto de fiscalização. Para fazer isso, você deve configurar a lógica do aplicativo para coletar dados do usuário e preenchê-los para solicitações de autorização.

Como um aplicativo autoriza solicitações com permissões verificadas

1. Reúna informações sobre o usuário atual. Normalmente, os detalhes de um usuário são fornecidos nos detalhes de uma sessão autenticada, como um JWT ou um cookie de sessão da web. Esses dados do usuário podem ser originários de uma fonte de identidade do Amazon Cognito vinculada ao seu repositório de políticas ou de outro provedor do OpenID Connect (OIDC).
2. Reúna informações sobre o recurso que um usuário deseja acessar. Normalmente, seu aplicativo receberá informações sobre o recurso quando um usuário fizer uma seleção que exija que seu aplicativo carregue um novo ativo.

3. Determine a ação que seu usuário deseja realizar.
4. Gere uma solicitação de autorização para Permissões Verificadas com o principal, a ação, o recurso e as entidades para a tentativa de operação do usuário. As permissões verificadas avaliam a solicitação em relação às políticas em seu repositório de políticas e retornam uma decisão de autorização.
5. Seu aplicativo lê a resposta de permissão ou negação das Permissões verificadas e aplica a decisão à solicitação do usuário.

As operações da API de permissões verificadas são incorporadas aos AWS SDKs. Para incluir permissões verificadas em um aplicativo, integre o AWS SDK do idioma escolhido ao pacote do aplicativo.

Para saber mais e baixar AWS SDKs, consulte Ferramentas para Amazon Web Services.

A seguir estão links para a documentação dos recursos de permissões verificadas em vários AWS SDKs.

- 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

O AWS SDK for JavaScript exemplo a seguir `IsAuthorized` se origina da autorização refinada Simplifique com as Permissões Verificadas da Amazon e o 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"
                }
            }
         }
       ]
    }
});
```

Mais recursos para desenvolvedores

- Workshop de permissões verificadas da Amazon
- Permissões verificadas pela Amazon - Recursos
- Implemente um provedor de política de autorização personalizado para aplicativos ASP.NET Core usando Amazon Verified Permissions
- Crie um serviço de qualificação para aplicativos de negócios usando Amazon Verified Permissions
- Simplifique a autorização refinada com as Permissões Verificadas da Amazon e o Amazon Cognito

Adicionando contexto

O contexto é a informação relevante para as decisões políticas, mas não faz parte da identidade de seu diretor, ação ou recurso. Talvez você queira permitir uma ação somente de um conjunto de endereços IP de origem ou somente se o usuário tiver feito login com o MFA. Seu aplicativo tem acesso a esses dados contextuais da sessão e deve preenchê-los para solicitações de autorização. Os dados de contexto em uma solicitação de autorização de permissões verificadas devem ser formatados em JSON em um elemento. `contextMap`

Os exemplos que ilustram esse conteúdo vêm de um exemplo de armazenamento de políticas. Para acompanhar, crie o repositório DigitalPetStorede políticas de amostra em seu ambiente de teste.

O objeto de contexto a seguir declara um de cada tipo de dados do Cedar para um aplicativo com base no exemplo de armazenamento de DigitalPetStorepolíticas.

```
"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"
         }
      }
   }
}
```

Tipos de dados no contexto de autorização

Booleano

Um binário `true` ou `false` valor. No exemplo, o valor booleano de `true` for `MfaAuthenticated` indica que o cliente realizou a autenticação multifatorial antes de solicitar a visualização do pedido.

Defina

Uma coleção de elementos de contexto. Os membros do conjunto podem ser todos do mesmo tipo, como neste exemplo, ou de tipos diferentes, incluindo um conjunto aninhado. No exemplo, o cliente está associado a três contas diferentes.

String

Uma sequência de letras, números ou símbolos, entre '' caracteres. No exemplo, a `UserAgent` string representa o navegador que o cliente usou para solicitar a visualização do pedido.

Longo

Um valor inteiro. No exemplo, `RequestedOrderCount` indica que essa solicitação faz parte de um lote que resultou da solicitação do cliente para visualizar quatro de seus pedidos anteriores.

Registro

Uma coleção de atributos. Você deve declarar esses atributos no contexto da solicitação. Um repositório de políticas com um esquema deve incluir essa entidade e os atributos da entidade no esquema. No exemplo, o `NetworkInfo` registro contém informações sobre o IP de origem do usuário, a geolocalização desse IP conforme determinado pelo cliente e a criptografia em trânsito.

EntityIdentifier

Uma referência a uma entidade e atributos declarados no `entities` elemento da solicitação. No exemplo, o pedido do usuário foi aprovado pelo funcionárioBob.

Para testar esse contexto de exemplo no DigitalPetStoreaplicativo de exemplo, você deve atualizar sua solicitação`entities`, seu esquema de armazenamento de políticas e a política estática com a descrição Customer Role - Get Order.

Modificando DigitalPetStore para aceitar o contexto de autorização

Inicialmente, não DigitalPetStoreé um repositório de políticas muito complexo. Ele não inclui nenhuma política ou atributo de contexto pré-configurado para dar suporte ao contexto que apresentamos. Para avaliar um exemplo de solicitação de autorização com essas informações de contexto, faça as seguintes modificações em seu repositório de políticas e em sua solicitação de autorização.

Schema

Aplique as seguintes atualizações ao esquema do repositório de políticas para oferecer suporte aos novos atributos de contexto. Atualize `GetOrder` da `actions` seguinte forma.

```
"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"
    ]
  }
}
```

Para referenciar o tipo de `record` dados nomeado `NetworkInfo` em seu contexto de solicitação, crie uma construção CommonType em seu esquema da seguinte maneira. Uma `commonType` construção é um conjunto compartilhado de atributos que você pode aplicar a diferentes entidades.

> **Note**
>
> Atualmente, o editor de esquema visual de Permissões Verificadas não oferece suporte a `commonType` construções. Ao adicioná-los ao seu esquema, você não pode mais visualizá-lo no modo Visual.

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

Policy

A política a seguir configura condições que devem ser atendidas por cada um dos elementos de contexto fornecidos. Ele se baseia na política estática existente com a descrição Customer Role - Get Order. Inicialmente, essa política exige apenas que o principal que faz a solicitação seja o proprietário do recurso.

```
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"
};
```

Agora exigimos que a solicitação para recuperar um pedido atenda às condições de contexto adicionais que adicionamos à solicitação.

1. O usuário deve ter feito login com o MFA.
2. O navegador da web do usuário `User-Agent` deve conter a string`My UserAgent`.
3. O usuário deve ter solicitado a visualização de 4 ou menos pedidos.
4. Um dos códigos de conta do usuário deve ser`111122223333`.
5. O endereço IP do usuário deve ser originário dos Estados Unidos, ele deve estar em uma sessão criptografada e seu endereço IP deve começar com`192.0.2.`.
6. Um funcionário deve ter aprovado seu pedido. No `entities` elemento da solicitação de autorização, declararemos um usuário `Bob` que tem a função de`Employee`.

Request body

Depois de configurar seu repositório de políticas com o esquema e a política apropriados, você pode apresentar essa solicitação de autorização à operação IsAuthorizedda API de permissões verificadas. Observe que o `entities` segmento contém uma definição de`Bob`, um usuário com uma função de`Employee`.

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

```
      },
      "RequestedOrderCount":{
       "long": 4
      },
      "AccountCodes": {
       "set": [
        {"long": 111122223333},
        {"long": 444455556666},
        {"long": 123456789012}
        ]
      },
      "NetworkInfo": {
        "record": {
          "IPAddress": {"string": "192.0.2.178"},
          "Country": {"string": "United States of America"},
          "SSL": {"boolean": true}
        }
      },
      "approvedBy": {
        "entityIdentifier": {
         "entityId": "Bob",
         "entityType": "DigitalPetStore::User"
        }
      }
     }
   },
   "entities": {
     "entityList": [
       {
         "identifier": {
            "entityType": "DigitalPetStore::User",
            "entityId": "Alice"
         },
         "attributes": {
            "memberId": {
              "string": "801b87f2-1a5c-40b3-b580-eacad506d4e6"
              }
         },
         "parents": [
            {
            "entityType": "DigitalPetStore::Role",
            "entityId": "Customer"
            }
         ]
```

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

Segurança no Amazon Verified Permissions

A segurança para com a nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de datacenters e arquiteturas de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve isso como a segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa produtos da AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos Programas de conformidade da AWS. Para conhecer os programas de conformidade que se aplicam ao Amazon Verified Permissions, consulte Serviços da AWS no escopo por programa de conformidade.
- Segurança na nuvem: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Verified Permissions. Os tópicos a seguir mostram como configurar o Verified Permissions de acordo com seus objetivos de segurança e conformidade. Saiba também como usar outros serviços da AWS para monitorar e proteger os recursos do Verified Permissions.

Tópicos

Proteção de dados no Amazon Verified Permissions

O AWS modelo de responsabilidade compartilhada se aplica à proteção de dados no Amazon Verified Permissions. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle

sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que você usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and GDPR no Blog de segurança da AWS.

- Para fins de proteção de dados, é recomendável que você proteja as credenciais da Conta da AWS e configure os usuários individuais com o AWS IAM Identity Center ou o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho.
- É recomendável também que você proteja seus dados da seguinte maneira:
  - Use uma autenticação multifator (MFA) com cada conta.
  - Use SSL/TLS para se comunicar com os recursos da AWS. O TLS 1.2 é obrigatório.
  - Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail.
  - Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão dos Serviços da AWS.
  - Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
  - Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.
- É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso é válido também quando você trabalha com o Verified Permissions ou outros Serviços da AWS usando o console, a API, a AWS CLI ou os SDKs da AWS. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendemos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.
- Os nomes de ação não devem incluir nenhuma informação confidencial.
- Também é altamente recomendável que você sempre use identificadores exclusivos, não mutáveis e não reutilizáveis para suas entidades (recursos e entidades principais). Em um ambiente de teste, você pode optar por usar identificadores de entidade simples, como `jane` ou `bob` para o nome de uma entidade do tipo `User`. No entanto, em um sistema de produção, é fundamental,

por motivos de segurança, que você use valores exclusivos que não possam ser reutilizados. Recomendamos que você use valores como identificadores universalmente exclusivos (UUIDs). Por exemplo, considere o usuário `jane`, que é desligado da empresa. Mais tarde, você deixa outra pessoa usar o nome `jane`. Esse novo usuário te, automaticamente acesso a tudo o que é concedido pelas políticas que ainda fazem referência a `User::"jane"`. O Verified Permissions e o Cedar não conseguem distinguir entre o novo usuário e o usuário anterior.

Essa orientação se aplica tanto aos identificadores de entidades principais quanto aos identificadores de recursos. Sempre use identificadores que sejam comprovadamente exclusivos e nunca sejam reutilizados para garantir que você não conceda acesso involuntariamente devido à presença de um identificador antigo em uma política.

- Certifique-se de que as strings fornecidas para definir os valores `Long` e `Decimal` estejam no intervalo válido de cada tipo. Além disso, certifique-se de que o uso de qualquer operador aritmético não resulte em um valor fora do intervalo válido. Se o intervalo for excedido, a operação resultará em uma exceção de estouro. Uma política que retorna um erro é ignorada, o que significa que uma política Permit pode inesperadamente não permitir o acesso ou uma política Forbid pode inesperadamente não bloquear o acesso.

Criptografia de dados

O Amazon Verified Permissions criptografa automaticamente todos os dados do cliente, como políticas com uma Chave gerenciada pela AWS. Portanto, o uso de uma chave gerenciada pelo cliente não é necessário nem compatível.

Gerenciamento de identidades e acesso para Amazon Verified Permissions

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAM os administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar os recursos de Permissões Verificadas. IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- Autenticando com identidades
- Gerenciando acesso usando políticas
- Como o Amazon Verified Permissions funciona com IAM
- Exemplos de políticas baseadas em identidade para Amazon Verified Permissions
- Solução de problemas de identidade e acesso do Amazon Verified Permissions

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz nas Permissões verificadas.

Usuário do serviço: se você usar o serviço do Verified Permissions para realizar o trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que você utilizar mais recursos do Verified Permissions para realizar o trabalho, possivelmente precisará de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no Verified Permissions, consulte Solução de problemas de identidade e acesso do Amazon Verified Permissions.

Administrador do serviço: se você for o responsável pelos recursos do Verified Permissions na empresa, provavelmente terá acesso total ao Verified Permissions. Cabe a você determinar quais funcionalidades e recursos do Verified Permissions os usuários do serviço devem acessar. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar IAM com permissões verificadas, consulteComo o Amazon Verified Permissions funciona com IAM.

IAM administrador — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso às Permissões Verificadas. Para ver exemplos de políticas baseadas em identidade de Permissões Verificadas que você pode usar IAM, consulte. Exemplos de políticas baseadas em identidade para Amazon Verified Permissions

Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como usuário do IAM ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Assinatura de solicitações de AWS API no Guia IAM do usuário.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia do usuário do AWS IAM Identity Center e Usar a autenticação multifator (MFA) na AWS no Guia do usuário do IAM .

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa de tarefas que requerem login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do usuário do IAM .

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte "O que é o Centro de Identidade do IAM?" no Guia do usuário AWS IAM Identity Center .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo no Guia do usuário do IAM .

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado Administradores do IAM e atribuir a esse grupo permissões para administrar recursos do IAM .

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de uma função) no Guia do usuário do IAM .

IAM funções

Uma IAM função é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma IAM função no AWS Management Console trocando de funções.

Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos de uso de funções, consulte Usando IAM funções no Guia IAM do usuário.

IAM funções com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre funções para federação, consulte Criar um perfil para um provedor de identidades de terceiros no Guia do usuário do IAM . Se você usar o Centro de identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões com uma função no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de permissões no Guia do usuário AWS IAM Identity Center .
- Permissões temporárias de usuário do IAM — Um usuário ou função do IAM pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar uma função do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte Como as IAM funções diferem das políticas baseadas em recursos no Guia do usuário.IAM
- Aplicativos em execução Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e fazendo solicitações AWS CLI de AWS API. É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém o perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para obter mais informações, consulte Como usar uma IAM função para conceder permissões a aplicativos executados em Amazon EC2 instâncias no Guia IAM do usuário.

Para saber se usar IAM funções ou usuários do IAM, consulte Quando criar uma IAM função (em vez de um usuário) no Guia do IAM usuário.

Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM .

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAM as políticas definem permissões para uma ação, independentemente do método usado para realizar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criação de IAM políticas no Guia do IAM usuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM .

Políticas baseadas em recursos

Políticas baseadas em atributos são documentos de políticas JSON que você anexa a um atributo. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes as políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões — Um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma IAM entidade (usuário ou função do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para entidades do IAM no Guia do usuário do IAM .

- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em. AWS Organizations AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte How SCPs work (Como os SCPs funcionam) no Guia do usuário do AWS Organizations .
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para ter mais informações, consulte Políticas de sessão no Guia do usuário do IAM .

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte Lógica de avaliação de políticas no Guia IAM do usuário.

Como o Amazon Verified Permissions funciona com IAM

Antes de usar IAM para gerenciar o acesso às Permissões Verificadas, saiba quais IAM recursos estão disponíveis para uso com as Permissões Verificadas.

IAM recursos que você pode usar com as permissões verificadas da Amazon

IAM recurso	Suporte do Verified Permissions
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações das políticas	Sim

IAM recurso	Suporte do Verified Permissions
Atributos de políticas	Sim
Chaves de condição de políticas	Não
ACLs	Não
ABAC (tags em políticas)	Não
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Não
Funções vinculadas ao serviço	Não

Para ter uma visão geral de como as permissões verificadas e outros AWS serviços funcionam com a maioria dos IAM recursos, consulte AWS os serviços que funcionam com IAM no Guia do IAM usuário.

Políticas baseadas em identidade para Verified Permissions

Suporta políticas baseadas em identidade	Sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criação de IAM políticas no Guia do IAM usuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM .

Exemplos de políticas baseadas em identidade para Verified Permissions

Para visualizar exemplos de políticas baseadas em identidade do Verified Permissions, consulte Exemplos de políticas baseadas em identidade para Amazon Verified Permissions.

Políticas baseadas em recursos no Verified Permissions

Oferece compatibilidade com políticas baseadas em recursos	Não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

Ações de política para Verified Permissions

Oferece compatibilidade com ações de políticas	Sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações do Verified Permissions, consulte Ações definidas pelo Amazon Verified Permissions na Referência de autorização do serviço.

As ações de políticas no Verified Permissions usam o seguinte prefixo antes da ação:

```
verifiedpermissions
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
      "verifiedpermissions:action1",
      "verifiedpermissions:action2"
         ]
```

Você também pode especificar várias ações usando caracteres-curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Get`, inclua a seguinte ação:

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

Para visualizar exemplos de políticas baseadas em identidade do Verified Permissions, consulte Exemplos de políticas baseadas em identidade para Amazon Verified Permissions.

Recursos de política para Verified Permissions

Oferece compatibilidade com recursos de políticas	Sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista de tipos de recursos do Verified Permissions e seus ARNs, consulte Tipos de recursos definidos pelo Amazon Verified Permissions na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon Verified Permissions.

Chaves de condição de política para Verified Permissions

Suporta chaves de condição de política específicas de serviço	Não

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM : variáveis e tags no Guia do usuário do IAM .

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia IAM do usuário.

ACLs no Verified Permissions

Oferece compatibilidade com ACLs	Não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes as políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com o Verified Permissions

Oferece compatibilidade com ABAC (tags em políticas)	Não

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a vários AWS recursos. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações onde o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte O que é ABAC? no Guia do usuário do IAM . Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM .

Uso de credenciais temporárias com o Verified Permissions

Oferece compatibilidade com credenciais temporárias	Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS esse trabalho IAM no Guia do IAM usuário.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte Alternar para uma função (console) no Guia do usuário do IAM .

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte Credenciais de segurança temporárias no IAM.

Permissões de entidade principal entre serviços para Verified Permissions

Suporta permissões de entidades principais	Sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos

serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

Perfis de serviço para Verified Permissions

Oferece suporte a perfis de serviço	Não

Um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do usuário do IAM .

Funções vinculadas a serviço para Verified Permissions

Oferece suporte a perfis vinculados ao serviço	Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas a serviços, consulte AWS serviços que funcionam com. IAM Encontre um serviço na tabela que inclua um `Yes` na coluna Função vinculada ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para Amazon Verified Permissions

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos do Verified Permissions. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Um IAM administrador deve criar IAM políticas que concedam aos usuários e funções permissão para realizar ações nos recursos de que

precisam. Por isso, o administrador precisa anexar essas políticas para os usuários que precisem delas.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de política JSON, consulte Criação de IAM políticas no Guia doIAM usuário.

Para obter detalhes sobre ações e tipos de recursos definidos pelo Verified Permissions, incluindo o formato dos ARNs para cada tipo de recurso, consulte Ações, recursos e chaves de condição para Amazon Verified Permissions na Referência de autorização do serviço.

Tópicos

- Melhores práticas de política
- Uso do console do Verified Permissions
- Permitir que usuários visualizem suas próprias permissões

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Acesso Verificado em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas da AWS ou Políticas gerenciadas da AWS para funções de trabalho no Guia do usuário do IAM .
- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre IAM como usar para aplicar permissões, consulte Políticas e permissões IAM no Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode gravar uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se

elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos de política JSON do IAM : condição no Guia do usuário do IAM .

- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais. O IAM Access Analyzer valida políticas novas e existentes para que elas sigam a linguagem de IAM política (JSON) e as melhores práticas do IAM. O IAM Access Analyzer oferece mais de 100 verificações de política e recomendações acionáveis para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do usuário do IAM .
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do usuário do IAM .

Para obter mais informações sobre as melhores práticas em IAM, consulte as melhores práticas de segurança IAM no Guia IAM do usuário.

Uso do console do Verified Permissions

Para acessar o console do Amazon Verified Permissions, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos de Permissões Verificadas em seu Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam a operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o console de Permissões Verificadas, anexe também as Permissões Verificadas `ConsoleAccess` ou a política `ReadOnly` AWS gerenciada às entidades. Para obter mais informações, consulte Como adicionar permissões a um usuário no Guia do usuário do IAM .

Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Solução de problemas de identidade e acesso do Amazon Verified Permissions

Use as seguintes informações para diagnosticar e corrigir problemas comuns que possam ser encontrados durante o trabalho com o Verified Permissions e o IAM.

Tópicos

Não tenho autorização para executar uma ação no Verified Permissions

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `verifiedpermissions:`*`GetWidget`* fictícias.

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

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `verifiedpermissions:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não tem autorização para executar a ação `iam:PassRole`, suas políticas deverão ser atualizadas para permitir a passagem de uma função para o Verified Permissions.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro de exemplo a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no Verified Permissions. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha acessem meus Conta da AWS recursos de Permissões Verificadas

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Verified Permissions oferece suporte a esses recursos, consulte Como o Amazon Verified Permissions funciona com IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do IAM usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Fornecer Contas da AWS acesso a terceiros no Guia do IAM usuário.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM .
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

Validação de conformidade do Amazon Verified Permissions

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- Arquitetura para segurança e conformidade com a HIPAA em Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

  > **Note**
  >
  > Nem todos Serviços da AWS são elegíveis para a HIPAA. Para obter mais informações, consulte a Referência dos serviços qualificados pela HIPAA.

- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Guias de conformidade do cliente — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.

- AWS Security Hub— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os atributos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.
- Amazon GuardDuty — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- AWS Audit Manager— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência no Amazon Verified Permissions

A infraestrutura global da AWS se baseia em Regiões da AWS e zonas de disponibilidade. A Regiões da AWS oferece várias zonas de disponibilidade separadas e isoladas fisicamente que são conectadas com baixa latência, throughputs elevadas e em redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Quando você cria um armazenamento de políticas do Verified Permissions, ele é criado em uma Região da AWS individual e é automaticamente replicado nos data centers que compõem as zonas de disponibilidade dessa região. No momento, o Verified Permissions não oferece suporte a replicações entre regiões.

Para obter mais informações sobre Regiões da AWS e zonas de disponibilidade, consulte Infraestrutura global da AWS.

Monitoramento do Amazon Verified Permissions

O monitoramento é importante para manter a confiabilidade, a disponibilidade e o desempenho do Amazon Verified Permissions e das outras soluções da AWS. A AWS fornece as ferramentas de monitoramento a seguir para observar o Verified Permissions, informar quando algo está errado e realizar ações automaticamente quando apropriado:

- O AWS CloudTrail captura chamadas de API e eventos relacionados feitos por sua conta da AWS ou em nome dela e entrega os arquivos de log a um bucket do Amazon S3 especificado por você. Você pode identificar quais usuários e contas chamaram a AWS, o endereço IP de origem no qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

Registro de chamadas de API do Amazon Verified Permissions usando o AWS CloudTrail

O Amazon Verified Permissions é integrado comAWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em Permissões verificadas. CloudTrail captura todas as chamadas de API para permissões verificadas como eventos. As chamadas capturadas incluem chamadas do console do Verified Permissions e chamadas de código para as operações de API do Verified Permissions. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para permissões verificadas. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação feita às Permissões Verificadas, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

Informações de permissões verificadas em CloudTrail

CloudTrail é ativado no seu Conta da AWS quando você cria a conta. Quando a atividade ocorre em Permissões verificadas, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode exibir, pesquisar e baixar eventos recentes em sua Conta da AWS. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para obter um registro contínuo de eventos na sua Conta da AWS, incluindo eventos do Verified Permissions, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na partição da AWS e entrega os arquivos de log no bucket do Amazon S3 que você especificou Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail serviços e integrações suportados
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e Recebendo arquivos de CloudTrail log de várias contas

Todas as ações de permissões verificadas são registradas CloudTrail e documentadas no Guia de referência da API de permissões verificadas da Amazon. Por exemplo, chamadas para as `ListPolicyStores` ações `CreateIdentitySourceDeletePolicy`, e geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário da raiz ou do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Eventos de dados como IsAuthorizede não IsAuthorizedWithTokensão registrados por padrão quando você cria um armazenamento de dados de trilhas ou eventos. Para registrar eventos de CloudTrail dados, você deve adicionar explicitamente os recursos suportados ou os tipos de recursos para os quais deseja coletar atividades. Para obter mais informações, consulte Eventos de dados, no Guia do usuário do AWS CloudTrail.

Noções básicas sobre entradas de arquivo de log do Verified Permissions

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

Tópicos

> **Note**
>
> Alguns campos foram ocultados nos exemplos por questão de privacidade dos dados.

IsAuthorized

```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
  "type": "AssumedRole",
  "principalId": "EXAMPLE_PRINCIPAL_ID",
  "arn": "arn:aws:iam::123456789012:role/ExampleRole",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    "eventTime": "2023-11-20T22:55:03Z",
    "eventSource": "verifiedpermissions.amazonaws.com",
    "eventName": "IsAuthorized",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
    "requestParameters": {
        "principal": {
            "entityType": "PhotoFlash::User",
            "entityId": "alice"
        },
        "action": {
            "actionType": "PhotoFlash::Action",
            "actionId": "ViewPhoto"
        },
        "resource": {
            "entityType": "PhotoFlash::Photo",
            "entityId": "VacationPhoto94.jpg"
        },
        "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    "responseElements": null,
    "additionalEventData": {
        "decision": "ALLOW"
    },
    "requestID": "346c4b6a-d12f-46b6-bc06-6c857bd3b28e",
    "eventID": "8a4fed32-9605-45dd-a09a-5ebbf0715bbc",
    "readOnly": true,
    "resources": [
        {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
}
```

BatchIsAuthorized

```
{
    "eventVersion": "1.08",
    "userIdentity": {
  "type": "AssumedRole",
  "principalId": "EXAMPLE_PRINCIPAL_ID",
  "arn": "arn:aws:iam::123456789012:role/ExampleRole",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    "eventTime": "2023-11-20T23:02:33Z",
    "eventSource": "verifiedpermissions.amazonaws.com",
    "eventName": "BatchIsAuthorized",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
    "requestParameters": {
        "requests": [
            {
                "principal": {
                    "entityType": "PhotoFlash::User",
                    "entityId": "alice"
                },
                "action": {
                    "actionType": "PhotoFlash::Action",
                    "actionId": "ViewPhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            },
            {
                "principal": {
```

```
                    "entityType": "PhotoFlash::User",
                    "entityId": "annalisa"
                },
                "action": {
                    "actionType": "PhotoFlash::Action",
                    "actionId": "DeletePhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            }
        ],
        "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    "responseElements": null,
    "additionalEventData": {
        "results": [
            {
                "request": {
                    "principal": {
                        "entityType": "PhotoFlash::User",
                        "entityId": "alice"
                    },
                    "action": {
                        "actionType": "PhotoFlash::Action",
                        "actionId": "ViewPhoto"
                    },
                    "resource": {
                        "entityType": "PhotoFlash::Photo",
                        "entityId": "VacationPhoto94.jpg"
                    }
                },
                "decision": "ALLOW"
            },
            {
                "request": {
                    "principal": {
                        "entityType": "PhotoFlash::User",
                        "entityId": "annalisa"
                    },
                    "action": {
                        "actionType": "PhotoFlash::Action",
                        "actionId": "DeletePhoto"
```

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

CreatePolicyStore

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
```

```
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "validationSettings": {
      "mode": "OFF"
    }
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-22T07:43:33.962794Z",
    "lastUpdatedDate": "2023-05-22T07:43:33.962794Z"
  },
  "requestID": "1dd9360e-e2dc-4554-ab65-b46d2cf45c29",
  "eventID": "b6edaeee-3584-4b4e-a48e-311de46d7532",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

ListPolicyStores

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListPolicyStores",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "maxResults": 10
  },
```

```
  "responseElements": null,
  "requestID": "5ef238db-9f87-4f37-ab7b-6cf0ba5df891",
  "eventID": "b0430fb0-12c3-4cca-8d05-84c37f99c51f",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

DeletePolicyStore

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

```
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

PutSchema

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

```
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

GetSchema

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

CreatePolicyTemplate


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

DeletePolicyTemplate


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


CreatePolicy


```
{
  "eventVersion": "1.08",
```

```
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:42:30Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityType": "PhotoApp::Role",
        "entityId": "PhotoJudge"
    },
    "resource": {
        "entityType": "PhotoApp::Application",
        "entityId": "PhotoApp"
    },
    "lastUpdatedDate": "2023-05-22T07:42:30.70852Z",
    "createdDate": "2023-05-22T07:42:30.70852Z"
  },
  "requestID": "93ffa151-3841-4960-9af6-30a7f817ef93",
  "eventID": "30ab405f-3dff-43ff-8af9-f513829e8bde",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
```

```
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

GetPolicy

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

CreateIdentitySource


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-19T01:27:44Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreateIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "configuration": {
      "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:000011112222:us-east-1:userpool/us-
east-1_aaaaaaaaaa"
      }
    },
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "principalEntityType": "User"
  },
  "responseElements": {
    "createdDate": "2023-07-14T15:05:01.599534Z",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-07-14T15:05:01.599534Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
  "requestID": "afcc1e67-d5a4-4a9b-a74c-cdc2f719391c",
  "eventID": "f13a41dc-4496-4517-aeb8-a389eb379860",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
```

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

GetIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:31Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "7a6ecf79-c489-4516-bb57-9ded970279c9",
  "eventID": "fa158e6c-f705-4a15-a731-2cdb4bd9a427",
  "readOnly": true,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
```

```
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

ListIdentitySources

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T20:05:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListIdentitySources",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "95d2a7bc-7e9a-4efe-918e-97e558aacaf7",
  "eventID": "d3dc53f6-1432-40c8-9d1d-b9eeb75c6193",
  "readOnly": true,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

DeleteIdentitySource


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeleteIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "d554d964-0957-4834-a421-c417bd293086",
  "eventID": "fe4d867c-88ee-4e5d-8d30-2fbc208c9260",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

Criação de recursos do Amazon Verified Permissions com o AWS CloudFormation

O Amazon Verified Permissions está integrado com AWS CloudFormation um serviço que ajuda você a modelar e configurar seus AWS recursos para que você possa gastar menos tempo criando e gerenciando seus recursos e infraestrutura. Você cria um modelo que descreve todos os AWS recursos que você deseja (como repositórios de políticas) e AWS CloudFormation provisiona e configura esses recursos para você.

Ao usar AWS CloudFormation, você pode reutilizar seu modelo para configurar seus recursos de Permissões Verificadas de forma consistente e repetida. Descreva seus recursos uma vez e, em seguida, provisione os mesmos recursos repetidamente em várias Contas da AWS regiões.

> **Important**
>
> O Amazon Cognito Identity não está disponível da mesma forma que as Permissões Verificadas da Regiões da AWS Amazon. Se você receber um erro AWS CloudFormation relacionado à Identidade do Amazon Cognito, por exemplo`Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient`, recomendamos que você crie o grupo de usuários e o cliente do Amazon Cognito no local geograficamente mais próximo de Região da AWS onde o Amazon Cognito Identity está disponível. Use esse grupo de usuários recém-criado ao criar a origem de identidade do Verified Permissions.

Permissões e AWS CloudFormation modelos verificados

Para provisionar e configurar recursos para o Verified Permissions e serviços relacionados, você deve entender os modelos do AWS CloudFormation. Os modelos são arquivos de texto formatados em JSON ou YAML. Esses modelos descrevem os recursos que você deseja provisionar em suas AWS CloudFormation pilhas. Se você não estiver familiarizado com JSON ou YAML, você pode usar o AWS CloudFormation Designer para ajudá-lo a começar a usar modelos. AWS CloudFormation Para obter mais informações, consulte O que é AWS CloudFormation Designer? no Guia do AWS CloudFormation usuário.

As permissões verificadas oferecem suporte à criação de fontes de identidade, políticas, repositórios de políticas e modelos de políticas no AWS CloudFormation. Para obter mais informações, incluindo

exemplos de modelos JSON e YAML para os recursos do Verified Permissions, consulte Referência de tipo de recurso do Amazon Verified Permissions no Guia do usuário do AWS CloudFormation .

AWS Construções CDK

AWS Cloud Development Kit (AWS CDK) É uma estrutura de desenvolvimento de software de código aberto para definir a infraestrutura de nuvem em código e provisioná-la por meio dela. AWS CloudFormation Construções ou componentes de nuvem reutilizáveis podem ser usados para criar modelos. AWS CloudFormation Esses modelos podem então ser usados para implantar sua infraestrutura de nuvem.

Para saber mais e baixar AWS os CDKs, consulte o AWS Cloud Development Kit.

A seguir estão links para a documentação de AWS CDK recursos de permissões verificadas, como construções.

- Construção de CDK L2 de permissões verificadas pela Amazon

Saiba mais sobre AWS CloudFormation

Para saber mais sobre isso AWS CloudFormation, consulte os seguintes recursos:

- AWS CloudFormation
- AWS CloudFormation Guia do usuário
- AWS CloudFormation API Reference
- AWS CloudFormation Guia do usuário da interface de linha de comando

Acesse o Amazon Verified Permissions usando um endpoint de interface (AWS PrivateLink)

Você pode usar o AWS PrivateLink para criar uma conexão privada entre sua VPC e o Amazon Verified Permissions. Você pode acessar o Verified Permissions como se estivesse em sua VPC, sem usar um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect. As instâncias na VPC não precisam de endereços IP públicos para acessar o Verified Permissions.

Você estabelece essa conexão privada criando um endpoint de interface, alimentado pelo AWS PrivateLink. Criaremos uma interface de rede de endpoint em cada sub-rede que você habilitar para o endpoint de interface. Essas são interfaces de rede gerenciadas pelo solicitante que servem como ponto de entrada para o tráfego destinado ao Verified Permissions.

Para obter mais informações, consulte Acessar os Serviços da AWS pelo AWS PrivateLink no Guia do AWS PrivateLink.

Considerações sobre o Verified Permissions

Antes de configurar um endpoint de interface para Verified Permissions, leia a seção Considerações no Guia do AWS PrivateLink.

O Verified Permissions oferece suporte a chamadas para todas as suas ações de API via endpoint da interface.

As políticas de endpoint da VPC não são compatíveis com o Verified Permissions. Por padrão, o acesso total ao Verified Permissions é permitido no endpoint da interface. Se desejar, você pode associar um grupo de segurança às interfaces de rede de endpoint para controlar o tráfego para o Verified Permissions por meio do endpoint da interface.

Criar um endpoint de interface para o Verified Permissions

Você pode criar um endpoint de interface para o Verified Permissions usando o console da Amazon VPC ou a AWS Command Line Interface (AWS CLI). Para mais informações, consulte Criar um endpoint de interface no Guia do usuário do AWS PrivateLink.

Crie um endpoint de interface para o Verified Permissions usando o seguinte nome de serviço:

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

Se você habilitar o DNS privado para o endpoint de interface, poderá fazer solicitações de API para o Verified Permissions usando seu nome DNS regional padrão. Por exemplo, `verifiedpermissions.us-east-1.amazonaws.com`.

Cotas do Amazon Verified Permissions

Você Conta da AWS tem cotas padrão, anteriormente chamadas de limites, para cada AWS serviço. A menos que especificado de outra forma, cada cota é específica da região . Você pode solicitar o aumento de algumas cotas, porém, algumas delas não podem ser aumentadas.

Para visualizar as cotas do Verified Permissions, abra o console do Service Quotas. No painel de navegação, escolha Serviços da AWS e selecione Verified Permissions.

Para solicitar o aumento da cota, consulte Solicitar um aumento de cota no Guia do usuário do Service Quotas. Se a cota ainda não estiver disponível no Service Quotas, use o formulário de aumento de limite.

Você Conta da AWS tem as seguintes cotas relacionadas às permissões verificadas.

Tópicos

Cotas para recursos

Nome	Padrão	Ajustá	Descrição
Armazenamentos de políticas por região por conta	Cada região compatível: 1.000	Sim	Número máximo de armazenamentos de políticas.
Modelos de política por repositório de políticas	Cada região compatível: 40	Sim	Número máximo de modelos de políticas em um armazenamento de políticas.
Origens de identidade por armazenam ento de políticas	1	Não	Número máximo de origens de identidade que você pode definir para

Nome	Padrão	Ajustá	Descrição
			um armazenamento de políticas.
Tamanho da solicitação de autorização[1]	1 MB	Não	Tamanho máximo de uma solicitação de autorização.
Tamanho da política	10,000 bytes	Não	Tamanho máximo de uma política individual.
Tamanho do esquema	100,000 bytes	Não	O tamanho máximo do esquema de um repositór io de políticas.
Tamanho da política por recurso	200.000 bytes[2]	Não	Tamanho máximo de todas as políticas que fazem referência a um recurso específico.

[1] A cota para uma solicitação de autorização é a mesma para IsAuthorizede. IsAuthorizedWithToken

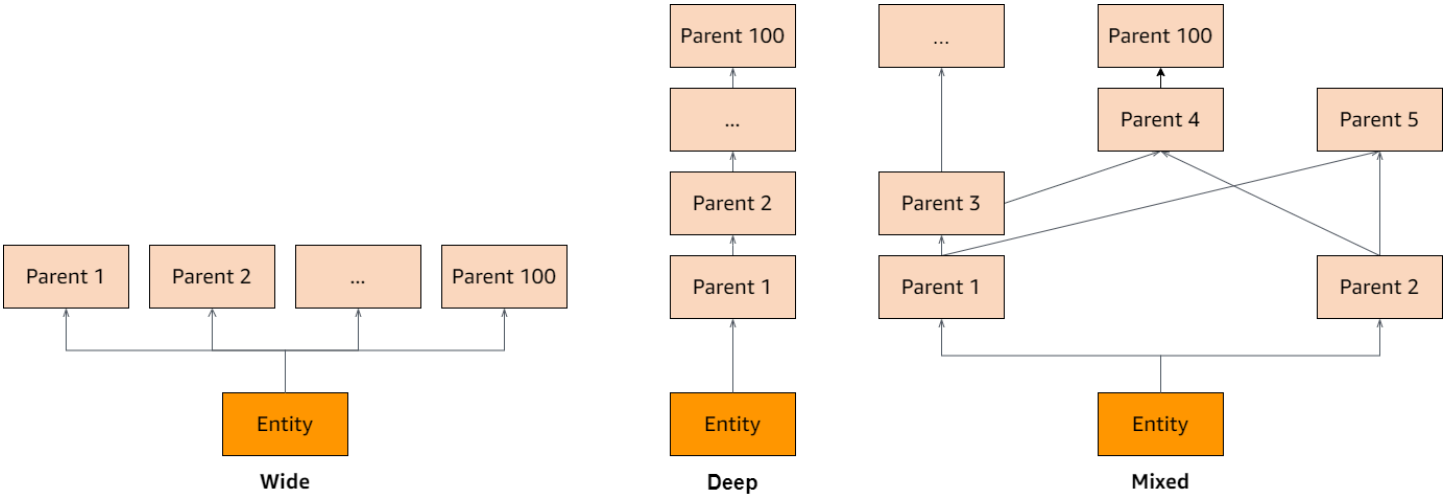
[2] O tamanho total de todas as políticas relacionadas a um único recurso não pode exceder 200.000 bytes. Para políticas vinculadas a modelos, o tamanho do modelo de política é contabilizado somente uma vez, mais o tamanho de cada conjunto de parâmetros usados para instanciar cada política vinculada a modelo.

Cotas para hierarquias

Nome	Padrão	Ajustá	Descrição
Pais transitivos por entidade principal	100	Não	Número máximo de pais transitivos para cada entidade principal.

Nome	Padrão	Ajustá	Descrição
Pais transitivos por ação	100	Não	Número máximo de pais transitivos para cada ação.
Pais transitivos por recurso	100	Não	Número máximo de pais transitivos para cada recurso.

O diagrama abaixo ilustra como pais transitivos podem ser definidos para uma entidade (entidade principal, ação ou recurso).



Cotas para operações por segundo

As permissões verificadas limitam as solicitações aos endpoints de serviço Região da AWS quando as solicitações do aplicativo excedem a cota de uma operação de API. As permissões verificadas podem retornar uma exceção quando você excede a cota de solicitações por segundo ou tenta operações de gravação simultâneas. Você pode ver suas cotas atuais do RPS em Service Quotas. Para evitar que os aplicativos excedam a cota de uma operação, você deve otimizá-los para novas tentativas e recuos exponenciais. Para obter mais informações, consulte Tentar novamente com o padrão de recuo e Gerenciar e monitorar a limitação da API em suas cargas de trabalho.

Nome	Padrão	Ajustá	Descrição
BatchIsAuthorized solicitações por segundo por região por conta	Cada região compatível: 30	Sim	O número máximo de BatchIsAuthorized solicitações por segundo.
BatchIsAuthorizedWithToken solicitações por segundo por região por conta	Cada região compatível: 30	Sim	O número máximo de BatchIsAuthorizedWithToken solicitações por segundo.
CreatePolicy solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de CreatePolicy solicitações por segundo.
CreatePolicyStore solicitações por segundo por região por conta	Cada região compatível: 1	Não	O número máximo de CreatePolicyStore solicitações por segundo.
CreatePolicyTemplate solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de CreatePolicyTemplate solicitações por segundo.
DeletePolicy solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de DeletePolicy solicitações por segundo.
DeletePolicyStore solicitações por segundo por região por conta	Cada região compatível: 1	Não	O número máximo de DeletePolicyStore solicitações por segundo.
DeletePolicyTemplate solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de DeletePolicyTemplate solicitações por segundo.
GetPolicy solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de GetPolicy solicitações por segundo.

Nome	Padrão	Ajustá	Descrição
GetPolicyTemplate solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de GetPolicyTemplate solicitações por segundo.
GetSchema solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de GetSchema solicitações por segundo.
IsAuthorized solicitações por segundo por região por conta	Cada região compatível: 200	Sim	O número máximo de IsAuthorized solicitações por segundo.
IsAuthorizedWithToken solicitações por segundo por região por conta	Cada região compatível: 200	Sim	O número máximo de IsAuthorizedWithToken solicitações por segundo.
ListPolicies solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de ListPolicies solicitações por segundo.
ListPolicyStores solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de ListPolicyStores solicitações por segundo.
ListPolicyTemplates solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de ListPolicyTemplates solicitações por segundo.
PutSchema solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de PutSchema solicitações por segundo.
UpdatePolicy solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de UpdatePolicy solicitações por segundo.

Nome	Padrão	Ajustá	Descrição
UpdatePolicyTemplate solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de UpdatePolicyTemplate solicitações por segundo.

Histórico de documento do Guia do usuário do Amazon Verified Permissions

A tabela a seguir descreve as versões de documentação do Verified Permissions.

Alteração	Descrição	Data
Fontes de identidade do OIDC	Agora você pode autorizar usuários dos provedores de identidade do OpenID Connect (OIDC).	8 de junho de 2024
Autorização em lote com tokens de origem de identidade	Agora você pode autorizar usuários de um grupo de usuários do Amazon Cognito em uma `BatchIsAuthorizedWithToken` única solicitação de API.	5 de abril de 2024
Criação de um repositório de políticas com o API Gateway	Agora você pode criar um repositório de políticas a partir de uma API existente e de um grupo de usuários do Amazon Cognito.	1º de abril de 2024
Conceitos de contexto e exemplo	Foram adicionadas informações sobre o contexto nas solicitações de autorização com permissões verificadas.	1 de fevereiro de 2024
Conceitos e exemplos de autorização	Foram adicionadas informações sobre solicitações de autorização com permissões verificadas.	1 de fevereiro de 2024
AWS CloudFormation integração	As permissões verificadas oferecem suporte à criação de	30 de junho de 2023

	fontes de identidade, políticas , repositórios de políticas e modelos de políticas no AWS CloudFormation.	
Lançamento inicial	Versão inicial do Guia do usuário do Amazon Verified Permissions	13 de junho de 2023

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.