Manual do usuário

Amazon Verified Permissions

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Verified Permissions: Manual do usuário

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o Amazon Verified Permissions? 1
Autorização no Verified Permissions 1
Linguagem de política Cedar 2
Benefícios do Verified Permissions 2
Acelerar o desenvolvimento das aplicações 2
Aplicações mais seguras 2
Recursos para o usuário final 3
Serviços relacionados 3
Como acessar o Amazon Permissions 3
Preços do Verified Permissions 5
Introdução aos repositórios de políticas 6
Pré-requisitos 7
Inscreva-se para um Conta da AWS 7
Etapa 1: criar um repositório PhotoFlash de políticas 7
Etapa 2: criar uma política 8
Etapa 3: Testando um repositório de políticas 9
Etapa 4: Limpar os recursos 10
Criação de um modelo de autorização 11
Não há um único modelo correto 12
Retornar erros 13
Foco nos recursos 13
Considere a multilocação 14
Comparando repositórios de políticas compartilhados e repositórios de políticas por inquilino 16
Como escolher 17
Armazenamentos de políticas 19
Criação de armazenamentos de políticas 20
Criação de um repositório de políticas usando Rust 29
Armazenamentos de políticas vinculados à API 34
Como funciona 36
Considerações 38
Adicionando ABAC 39
Passando para a produção 40
Solução de problemas 43

Exclusão de armazenamentos de políticas ... 46
Aliases do repositório de políticas ... 48
Propriedades dos aliases do repositório de políticas ... 48
Criação de aliases de armazenamento de políticas ... 51
Recuperando aliases do repositório de políticas ... 53
Obtendo um alias de armazenamento de políticas ... 53
Listando aliases do repositório de políticas ... 53
Excluindo aliases do repositório de políticas ... 55
Exclusão reversível de um alias do repositório de políticas ... 56
Exclusão definitiva de um alias do repositório de políticas ... 56
Usando aliases do repositório de políticas ... 57
Usando aliases do Policy Store em operações ... 58
Usando aliases do Policy Store Across Regiões da AWS ... 58
Os aliases do repositório de políticas não são um mecanismo de controle de tráfego ... 59
Realizando alterações no repositório de políticas sem tempo de inatividade ... 60
Controlar o acesso ... 62
permissões verificadas: CreatePolicyStoreAlias ... 62
permissões verificadas: GetPolicyStoreAlias ... 63
permissões verificadas: ListPolicyStoreAliases ... 63
permissões verificadas: DeletePolicyStoreAlias ... 64
Limitando as permissões do alias do repositório de políticas ... 64
Esquema do armazenamento de políticas ... 67
Esquema de edição ... 69
Modo de validação de política ... 72
Políticas ... 74
Criação de políticas estáticas ... 75
Edição de políticas estáticas ... 78
... 81
Avalie o contexto de exemplo ... 83
Políticas de teste ... 89
Tamanho da política por recurso ... 91
Como os tamanhos das apólices são calculados ... 92
Permanecendo dentro da cota ... 92
Template-linked exemplo de tamanho de política ... 93
Exemplo de política ... 95
Usa a notação de colchetes para referenciar atributos de token ... 96

Usa notação de pontos para referenciar atributos 96
Reflete Amazon Cognito Atributos do token de ID 97
Reflete os atributos do token de ID OIDC 97
Reflete Amazon Cognito atributos do token de acesso 98
Reflete os atributos do token de acesso OIDC 98
Modelos de políticas e políticas vinculadas a modelos 99
Criação de modelos de política 100
Criação de políticas vinculadas a modelos 102
Edição de modelos de política 104
Exemplo de políticas vinculadas a modelos 106
PhotoFlash exemplos 106
DigitalPetStore exemplos 108
TinyToDo exemplos 108
Fontes de identidade 110
Escolhendo o provedor de identidade certo 111
Trabalhando com fontes de Amazon Cognito identidade 111
Criação de origens de identidade 114
Edição de origens de identidade 118
Mapeamento de tokens para o esquema 120
Validação de clientes e públicos 131
Trabalhando com fontes de identidade do OIDC 134
Criação de origens de identidade 136
Edição de origens de identidade 139
Mapeamento de tokens para o esquema 141
Validação de clientes e públicos 148
Integrações 153
Usando o Express 153
Pré-requisitos 154
Configurando a integração 154
Configurar autorização 155
Implementando o middleware de autorização 158
Testando a integração 159
Solução de problemas 159
Próximas etapas 159
Autorizar solicitações 160
Operações de API 161

Modelo de teste ... 162
Integração com aplicativos ... 164
Como fazer solicitações de API ... 167
Endpoints de permissões verificadas ... 167
Parâmetros de solicitação ... 167
Identificadores de solicitação ... 167
Autenticação da API de ... 167
Bibliotecas disponíveis ... 167
Fazendo solicitações de API usando o método POST ... 168
Segurança ... 171
Proteção de dados ... 171
Criptografia de dados ... 173
Chaves gerenciadas pelo cliente ... 173
Gerenciamento de identidade e acesso ... 194
Público ... 194
Autenticação com identidades ... 195
Gerenciar o acesso usando políticas ... 197
Como o Amazon Verified Permissions funciona com IAM ... 199
IAM políticas para permissões verificadas ... 205
Exemplos de políticas baseadas em identidade ... 208
AWS políticas gerenciadas ... 211
Solução de problemas ... 214
Validação de conformidade ... 216
Resiliência ... 217
Monitoramento ... 218
CloudTrail troncos ... 218
Informações de permissões verificadas em CloudTrail ... 219
Noções básicas sobre entradas de arquivo de log do Verified Permissions ... 220
Trabalhando com AWS CloudFormation ... 242
Permissões e CloudFormation modelos verificados ... 242
AWS Construções CDK ... 243
Saiba mais sobre CloudFormation ... 243
Usando AWS PrivateLink ... 244
Considerações ... 244
Como criar um endpoint de interface ... 244
Criar uma política de endpoint ... 245

Cotas 247
Cotas para recursos 247
Cotas para hierarquias 249
Cotas para operações por segundo 250
Termos e conceitos 255
Modelo de autorização 256
Solicitação de autorização 256
Resposta de autorização 256
Políticas consideradas 256
Dados de contexto 257
Políticas determinantes 257
Dados da entidade 257
Permissões, autorização e entidades principais 257
Aplicação de políticas 257
Armazenamentos de políticas 258
Apelido do repositório de políticas 258
Nome da política 258
Nome do modelo de política 259
Políticas atendidas 259
Diferenças em relação ao Cedar 259
Suporte do modelo de política 259
Suporte ao esquema 260
Definição de grupos de ação 260
Formatação de entidades 260
Limites de comprimento e tamanho 265
Perguntas frequentes sobre o Cedar v4 267
Por que recebo uma mensagem de erro dizendo que o Cedar 2 não é mais suportado quando ligo para Amazon Verified Permissions? 267
Por que algumas políticas, modelos de políticas e esquemas não são compatíveis com o Cedar 4? 268
Como posso saber se meu repositório de políticas está usando o Cedar 2 ou o Cedar 4? 268
Como faço o upgrade para o Cedar 4? 270
Posso fazer o downgrade da minha loja de políticas do Cedar 4 para o Cedar 2? 270
Por que estou recebendo uma mensagem de erro dizendo que meu repositório de políticas está configurado para o Cedar 2? 271
Como faço para tornar meu esquema compatível com o Cedar 4? 271

Como faço para tornar minhas políticas e modelos compatíveis com o Cedar 4? 272
Histórico do documento 274
..... cclxxvi

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 do Cedar para definir permissões refinadas para proteger os recursos do seu aplicativo.

Para obter orientação e exemplos sobre como configurar um ponto de decisão de política (PDP) usando permissões verificadas, consulte Implementando uma PDP usando as permissões verificadas da Amazon em Orientação AWS prescritiva.

Tópicos

- Autorização no Verified Permissions
- Linguagem de política Cedar
- Benefícios do Verified Permissions
- Serviços relacionados
- Como acessar o Amazon Permissions
- Preços do Verified Permissions

Autorização no Verified Permissions

As Permissões verificadas fornecem autorização verificando se um diretor tem permissão para realizar uma ação em um recurso em um determinado contexto em seu aplicativo. As permissões verificadas pressupõem que o principal tenha sido previamente identificado e autenticado por outros meios, como o uso de protocolos como o OpenID Connect, um provedor Amazon Cognito hospedado ou outra solução de autenticação. As permissões verificadas são independentes de onde o principal é gerenciado e como foi autenticado.

As Permissões Verificadas são um serviço que permite aos clientes criar, manter e testar políticas no Console de gerenciamento da AWS, usando programaticamente as Permissões Verificadas ou por meio de infraestrutura APIs, como soluções de código. CloudFormation As permissões são expressas com a linguagem de política Cedar. O aplicativo cliente solicita autorização APIs para avaliar as políticas do Cedar armazenadas com o 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 um aplicativo, você precisa garantir que somente diretores autorizados, usuários humanos ou máquinas possam acessar o aplicativo e fazer 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 4.7 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.

O Verified Permissions fornece integrações com estruturas de desenvolvimento populares, facilitando a implementação da autorização em seus aplicativos com o mínimo de alterações no código. Essas integrações permitem que você se concentre em sua lógica comercial principal, enquanto as Permissões Verificadas lidam com as decisões de autorização.

- Express.js — Uma integração baseada em middleware que permite proteger endpoints de API em seus aplicativos Express sem modificar os manipuladores de rotas existentes. Para obter mais informações, consulte the section called "Usando o Express".

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 tokens e AWS credenciais de acesso 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 Amazon Cognito usuários. 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 em API Gateway. Para obter mais informações sobre API Gateway, consulte o Guia do API Gateway desenvolvedor.
- Centro de Identidade do AWS IAM: 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 Guia do usuário do Centro de Identidade do AWS IAM.

Como acessar o Amazon Permissions

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

Console de gerenciamento da AWS

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 Ferramentas da AWS para 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 código de amostra para várias linguagens e plataformas de programação (Java, Python, Ruby, .NET, iOS, Android etc.). Eles SDKs fornecem uma maneira conveniente de criar acesso programático às permissões verificadas e. AWS Por exemplo, SDKs cuidar de tarefas como assinar criptograficamente solicitações, gerenciar erros e repetir solicitações automaticamente.

Para saber mais e fazer o download AWS SDKs, consulte Ferramentas para Amazon Web Services.

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

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

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. CloudFormation Construções ou componentes de nuvem reutilizáveis podem ser usados para criar modelos. CloudFormation Esses modelos podem então ser usados para implantar sua infraestrutura de nuvem.

Para saber mais e baixar o AWS CDK, 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 Gerenciamento de Faturamento e Custos da AWS. 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.

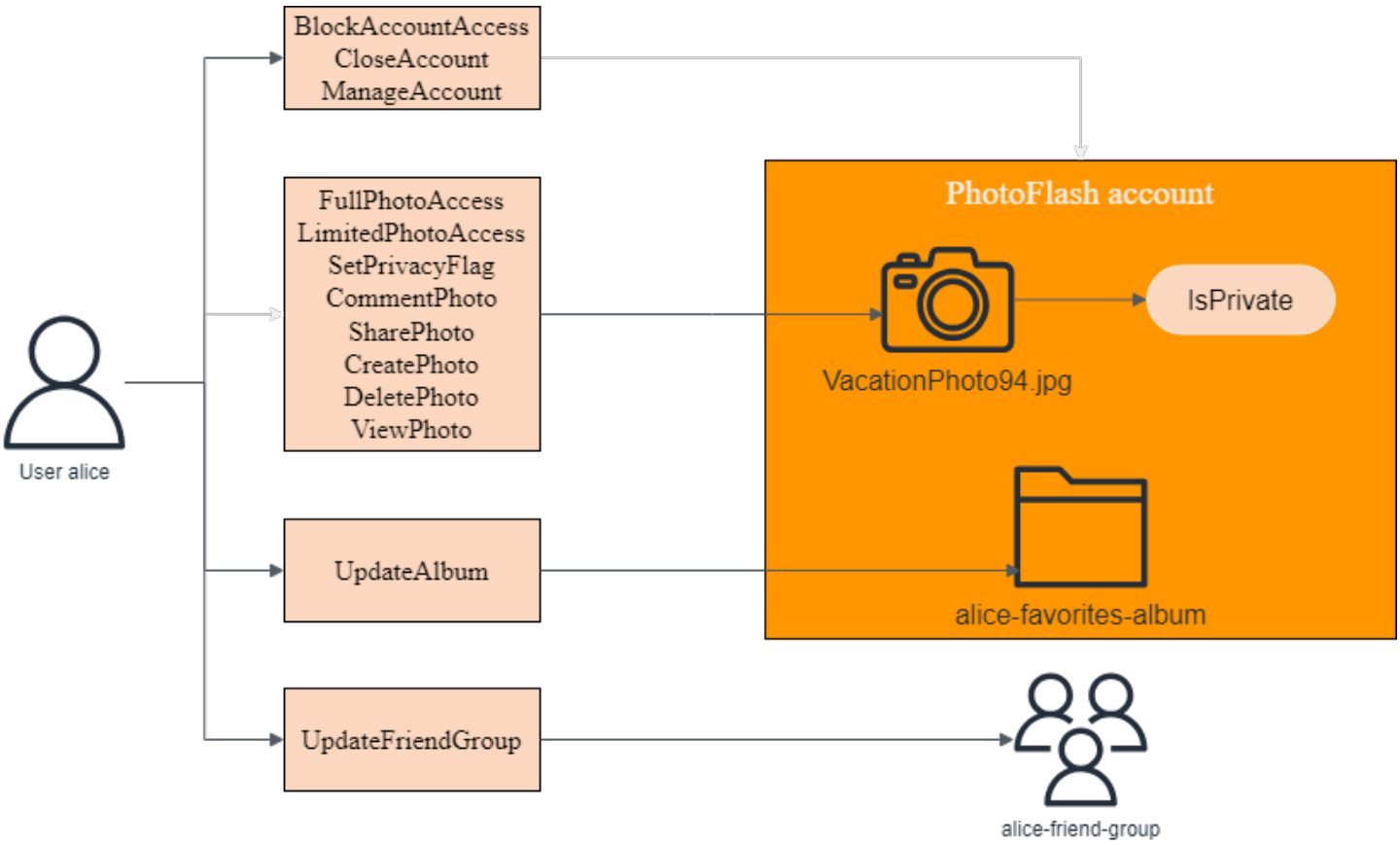
Se você tiver dúvidas sobre AWS faturamento, contas e eventos, entre em contato com Suporte.

Crie seu primeiro repositório de políticas de permissões verificadas da Amazon

Para este tutorial, vamos supor que você seja o desenvolvedor de um aplicativo de compartilhamento de fotos e esteja procurando uma maneira de controlar quais ações os usuários do aplicativo podem realizar. Você quer controlar quem pode adicionar, excluir ou ver fotos e álbuns de fotos. Você também quer controlar quais ações um usuário pode realizar em sua conta. Eles podem gerenciar sua conta? Que tal a conta de um amigo? Para controlar essas ações, você criaria políticas que permitem ou proíbem essas ações com base na identidade do usuário. O Verified Permissions oferece repositórios de políticas, ou contêineres, para abrigar essas políticas.

Neste tutorial, abordaremos a criação de um exemplo de armazenamento de políticas usando o console Amazon Verified Permissions. O console oferece alguns exemplos de opções de armazenamento de políticas e vamos criar um repositório PhotoFlashde políticas. Esse repositório de políticas permite que diretores, como usuários, realizem ações, como compartilhamento, em recursos, como fotos ou álbuns.

O diagrama a seguir ilustra as relações entre uma diretora e as ações que ela pode realizar com vários recursos, a saber, sua PhotoFlash conta, o `VactionPhoto94.jpg` arquivo, o álbum `alice-favorites-album` de fotos e o grupo `alice-friend-group` de usuários. `User::alice`



Agora que você tem uma compreensão do repositório de PhotoFlashpolíticas, vamos criar o repositório de políticas e explorá-lo.

Pré-requisitos

Inscreva-se para um Conta da AWS

Para começar AWS, você precisa de um Conta da AWS. Para obter informações sobre como criar um Conta da AWS, consulte Introdução a um Conta da AWS no Guia de AWS Gerenciamento de contas referência.

Etapa 1: criar um repositório PhotoFlash de políticas

No procedimento a seguir, você criará um repositório PhotoFlashde políticas usando o AWS console.

Para criar um repositório PhotoFlash de políticas

1. No console de Permissões verificadas, escolha Criar novo repositório de políticas.

2. Para opções iniciais, escolha Iniciar em um exemplo de armazenamento de políticas.
3. Em Projeto de amostra, escolha PhotoFlash.
4. Escolha Criar armazenamento de políticas.

Depois de ver a mensagem “Armazenamento de políticas criado e configurado”, escolha Ir para a visão geral para explorar seu repositório de políticas.

Etapa 2: criar uma política

Quando você criou o repositório de políticas, foi criada uma política padrão que permite que os usuários tenham controle total sobre suas próprias contas. Essa é uma política útil, mas, para nossos propósitos, vamos criar uma política mais restritiva para explorar as nuances das permissões verificadas. Se você se lembra do diagrama que vimos anteriormente no tutorial, tínhamos um diretor,`User::alice`, que poderia realizar uma ação,`UpdateAlbum`, em um recurso,`alice-favorites-album`. Vamos adicionar a política que permitirá que Alice, e somente Alice, gerencie este álbum.

Como criar uma política

1. No console de Permissões verificadas, escolha o repositório de políticas que você criou na etapa 1.
2. Na navegação, escolha Políticas.
3. Selecione Criar política e escolha Criar política estática.
4. Para Efeito da política, escolha Permitir.
5. Em Escopo de principais, escolha Principal específico e, em Especificar tipo de entidade, escolha PhotoFlash: :Usuário e, em Especificar identificador de entidade, insira. **`alice`**
6. Em Escopo de recursos, escolha Recurso específico e, em Especificar tipo de entidade, escolha PhotoFlash: :Álbum e, em Especificar identificador de entidade, insira**`alice-favorites-album`**.
7. Em Escopo de ações, escolha Conjunto específico de ações e, em seguida, em Ação (ões) às quais essa política deve se aplicar, selecione UpdateAlbum.
8. Escolha Próximo.
9. Em Detalhes, para Descrição da política - opcional, insira**`Policy allowing alice to update alice-favorites-album.`**.

10. Escolha Criar política.

Agora que você criou uma política, pode testá-la no console de Permissões verificadas.

Etapa 3: Testando um repositório de políticas

Depois de criar seu repositório de políticas e sua política, você pode testá-los executando uma solicitação de autorização simulada usando a bancada de testes de permissões verificadas.

Para testar as políticas do repositório de políticas

1. Abra o console de Permissões verificadas. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Banco de testes.
3. Escolha Modo visual.
4. Para Diretor, faça o seguinte:
    a. Em Principal tomando ação, escolha PhotoFlash: :Usuário e, em Especificar identificador da entidade, insira**`alice`**.
    b. Em Atributos, em Conta: Entidade, verifique se a entidade PhotoFlash: :Conta está selecionada e, em Especificar identificador da entidade, insira**`alice-account`**.
5. Em Recurso, em Recurso no qual o principal está atuando, escolha o tipo de recurso PhotoFlash: :Album e, em Especificar identificador de entidade, insira**`alice-favorites-album`**.
6. Em Ação, escolha PhotoFlash: :Ação::” UpdateAlbum "na lista de ações válidas.
7. Na parte superior da página, escolha Executar solicitação de autorização para simular a solicitação de autorização para as políticas do Cedar no repositório de políticas de amostra. A bancada de testes deve exibir Decisão: Permitir, indicando que nossa política está funcionando conforme o esperado.

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 na política que você criou na etapa 2.

Valor da entidade principal	Valor Account: Entity da entidade principal	Valor do recurso	Valor pai do recurso	Ação	Decisão de autorização
PhotoFlash: :Usuário | bob	PhotoFlash: :Conta | alice - conta	PhotoFlash: :Álbum | alice-favorites-album	N/A	PhotoFlash: :Ação::"" UpdateAlbum	Deny
PhotoFlash: :Usuário | alice	PhotoFlash: :Conta | alice - conta	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Conta | bob-account	PhotoFlash: :Ação::"" ViewPhoto	Deny
PhotoFlash: :Usuário | alice	PhotoFlash: :Conta | alice - conta	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Conta | alice - conta	PhotoFlash: :Ação::"" ViewPhoto	Permitir
PhotoFlash: :Usuário | alice	PhotoFlash: :Conta | alice - conta	PhotoFlash: :Foto | bob-photo.jpeg	PhotoFlash: :Álbum | Bob-Vacation-Album	PhotoFlash: :Ação::"" DeletePhoto	Deny

Etapa 4: Limpar os recursos

Depois de terminar de explorar seu repositório de políticas, exclua-o.

Para excluir um armazenamento de políticas

1. No console de Permissões verificadas, escolha o repositório de políticas que você criou na etapa 1.
2. Na navegação, escolha Configurações.
3. Em Excluir repositório de políticas, escolha Excluir este repositório de políticas.
4. No repositório Excluir esta política? caixa de diálogo, digite delete e, em seguida, escolha Excluir.

Práticas recomendadas para criar um modelo de autorização

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. 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? Como eles estão organizados? Por exemplo, os arquivos residem em uma pasta?
- A organização dos recursos desempenha um papel no modelo de permissões?
- 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 funções são globais ou têm escopo definido? Por exemplo, um “operador” está limitado a um único inquilino ou “operador” significa operador em todo o aplicativo?
- 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. Em vez disso, é antes de tudo 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 podem fazer 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.

Para ajudar a responder às perguntas e chegar a um modelo ideal, faça o seguinte:

- Analise os padrões de design do Cedar no Guia de referência da linguagem política do Cedar.
- Considere as melhores práticas no Guia de referência da linguagem política da Cedar.
- Considere as melhores práticas incluídas nesta página.

Práticas recomendadas

- Não existe um modelo canônico “correto”
- Retorna 403 erros proibidos em vez de 404 erros não encontrados
- Concentre-se em seus recursos além das operações de API
- Considerações sobre multilocação

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.

Retorna 403 erros proibidos em vez de 404 erros não encontrados

É melhor retornar um erro 403 Forbidden para solicitações que incluam uma entidade, especialmente um recurso, que não corresponde a nenhuma política, em vez de um erro 404 Não encontrado. Isso fornece o mais alto nível de segurança porque você não está expondo se uma entidade existe ou não, apenas que a solicitação não atendeu às condições da política em nenhuma política no repositório de políticas.

Concentre-se em seus recursos além das operações de API

Na maioria dos aplicativos, as permissões são modeladas com base nos recursos suportados. 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, à medida que você desenvolve seu modelo, essa abordagem focada em API pode não ser adequada para aplicativos com modelos de autorização muito granulares, pois APIs é apenas um proxy do que seus clientes estão realmente tentando proteger: os dados e os recursos subjacentes. Se vários APIs controlam o acesso aos mesmos recursos, pode ser difícil para os administradores raciocinar sobre os caminhos para 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 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.

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. Para obter mais informações, consulte Considerações sobre o design multilocatário do Amazon Verified Permissions na AWS Orientação prescritiva.

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 terá um grande impacto em sua AWS fatura. 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

Quando seu aplicativo tem vários repositórios de políticas, seus modelos de políticas e um esquema de armazenamento de políticas adicionam um nível de sobrecarga de design e manutenção em cada armazenamento 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.Deve incluir identificadores de inquilinos nas políticas e solicitações de autorização.	Baixo. O isolamento é o comportamento padrão. As políticas específicas do inquilino são inacessíveis para outros inquilinos.
Fluxo de autorização	Baixo. Todas as consultas têm como alvo um repositório de políticas.	Médio. É necessário manter mapeamentos entre cada inquilino e seu ID do repositório de políticas.
Gerenciamento de modelos e esquemas	Baixo. É preciso fazer com que um esquema funcione para todos os inquilinos.	Alto. Esquemas e modelos podem ser menos complexos individualmente, mas as mudanças exigem mais coordenação e complexidade.

Gerenciamento de políticas globais	Baixo. Todas as políticas são globais e podem ser atualizadas centralmente.	Alto. Você deve adicionar políticas globais a cada repositório de políticas na integração. Replique as atualizações de políticas globais entre vários repositórios de políticas.
Desembarque do inquilino	Alto. Deve identificar e excluir somente as políticas específicas do inquilino.	Baixo. Exclua o repositório de políticas.
Cotas de recursos de serviço	Alto. Os inquilinos compartilham cotas de recursos que afetam os repositórios de políticas, como tamanho do esquema, tamanho da política por recurso e fontes de identidade por armazenamento de políticas.	Baixo. Cada inquilino tem cotas de recursos dedicadas.

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.

Armazenamentos de políticas do Amazon Verified Permissions

Um armazenamento de políticas é um contêiner para políticas e modelos de políticas. Em cada repositório de políticas, você pode criar um esquema usado para validar políticas adicionadas ao repositório de políticas. Além disso, você pode ativar a validação da política. Se você adicionar uma política a um repositório de políticas com a validação de política ativada, os tipos de entidade, os tipos comuns e as ações definidos na política serão validados em relação ao esquema e as políticas inválidas serão rejeitadas.

A proteção contra exclusão impede a exclusão acidental de um repositório de políticas. A proteção contra exclusão está ativada em todos os novos repositórios de políticas criados por meio do Console de gerenciamento da AWS. Por outro lado, ele está desativado para todos os repositórios de políticas criados por meio de uma chamada de API ou SDK.

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. Você também pode criar aliases de armazenamento de políticas para se referir aos seus repositórios de políticas por nomes amigáveis. Para obter mais informações, consulte Aliases da loja de políticas de permissões verificadas da Amazon.

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. Por exemplo, `MyApplicationNamespace::exampleType`. As permissões verificadas oferecem suporte a até 100 namespaces por repositório de políticas. Esses namespaces ajudam a manter as coisas em ordem quando você trabalha com vários aplicativos semelhantes. Por exemplo, em aplicativos multilocatários, usar um namespace para acrescentar o nome do inquilino aos tipos definidos no esquema os tornará distintos de seus equivalentes similares usados pelos outros inquilinos. Ao examinar os registros das solicitações de autorização, você poderá identificar facilmente o inquilino que processou a solicitação de autorização. Para obter mais informações, consulte Namespaces no Guia de referência da linguagem de política Cedar.

Tópicos

- Criação de armazenamentos de políticas do Verified Permissions
- Armazenamentos de políticas vinculados à API

- Exclusão de armazenamentos de políticas

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.
- Configurar com API Gateway 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 associação a grupos de usuários ou outros atributos.
- 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 repositório de políticas usando o Configuração guiada método de configuração

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, um aplicativo web de atualizações meteorológicas.
4. Na seção Detalhes, digite um Namespace para seu esquema. Para obter mais informações sobre namespaces, consulte Namespaces na documentação do Cedar.
5. Escolha Próximo.
6. Na janela Tipo de recurso, digite um nome para seu tipo de recurso. Por exemplo, `currentTemperature` pode ser um recurso para o aplicativo web de atualizações meteorológicas.
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. Por exemplo, `temperatureFormat` pode ser um atributo do `currentTemperature` recurso e ser Fahrenheit ou Celsius. 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. Por exemplo, `viewTemperature` pode ser uma ação no aplicativo web de atualizações meteorológicas. 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. Por padrão, o usuário é adicionado a esse campo, mas pode ser substituído.
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. O Verified Permissions usa os valores de atributo especificados ao verificar as políticas com base no esquema. Para remover um atributo que foi adicionado ao tipo principal, escolha Remover ao lado do atributo.
    - Escolha o Cognito User Pool se o ID e os atributos do principal forem fornecidos a partir de um ID ou token de acesso gerado por. Amazon Cognito Escolha Conectar grupo de usuários. Selecione Região da AWSe digite a ID do grupo de Amazon Cognito usuários ao

qual se conectar. Selecione Conectar. Para obter mais informações, consulte Autorização com o Amazon Verified Permissions no Guia do desenvolvedor do Amazon Cognito.

- Escolha Provedor externo do OIDC se o ID e os atributos do principal forem extraídos de um token de and/or acesso ao ID, gerado por um provedor externo do OIDC, e adicione os detalhes do provedor e do token.

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. Por exemplo, `user-id` pode ser um identificador de entidade no aplicativo web de atualizações meteorológicas.

      > **Note**
      >
      > Se você estiver usando Amazon Cognito, o identificador da entidade deve ser formatado como`<userpool-id>|<sub>`.

    - 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. Por exemplo, `temperature-id` pode ser um identificador de entidade no aplicativo web de atualizações meteorológicas.
    - 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 Configurar com API Gateway e uma fonte de identidade método de configuração

A API Gateway opção 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 Gateway API 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 API Gateway 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 no API Gateway console e selecione Importar API para ver as atualizações.

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 Amazon Cognito usuários ou um tipo de IdP do OpenID Connect (OIDC).

5. Se você escolheu 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 API-linked autorização.

   c. Em Validação do cliente do aplicativo, você pode limitar o escopo de um repositório de políticas a um subconjunto dos clientes do Amazon Cognito aplicativo 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 externo do 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 obter mais informações, consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens OIDC para o esquema.

   c. (opcional) Em Reivindicações de token - opcional, escolha Adicionar uma declaração de token, insira um nome para o token e selecione um tipo de valor.

   d. Em reivindicações de token de usuário e grupo, faça o seguinte:

i. Insira um nome de declaração de usuário no token 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.

ii. Insira um nome de declaração de grupo no token da 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.

e. Na validação do Audience, escolha `Add value` e adicione um valor que você deseja que seu repositório de políticas aceite nas solicitações de autorização.

f. Escolha Próximo.

7. Se você escolher Amazon Cognito, as Permissões verificadas consultam 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, escolha se você deseja anexar manualmente o autorizador Lambda manualmente mais tarde ou se deseja que as Permissões Verificadas façam isso por você agora e 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 repositório de políticas.

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 de implantação do autorizador Lambda mostrar sucesso, se você optar por anexar o autorizador manualmente, 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 API Gateway Lambda 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 as políticas.

Sample policy store

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

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.

- DigitalPetStoreé 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 repositório de políticas de DigitalPetStoreamostra não inclui modelos de políticas. 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 de permissões verificadas da Amazon

Empty policy store

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

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 de permissões verificadas da Amazon.

AWS CLI

Para criar um repositório de políticas vazio usando o 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 de permissões verificadas da Amazon.

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

AWS SDKs

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

Implementando as permissões verificadas pela Amazon no Rust com o AWS SDK

Este tópico fornece um exemplo prático da implementação das permissões verificadas da Amazon no Rust com o AWS SDK. Este exemplo mostra como desenvolver um modelo de autorização que pode testar se um usuário consegue ver uma foto. O código de amostra usa a caixa aws-sdk-verifiedpermissions do AWS SDK para Rust, que oferece um conjunto robusto de ferramentas para interagir com. Serviços da AWS

Pré-requisitos

Antes de começar, verifique se você tem a AWS CLI configurada em seu sistema e se está familiarizado com o Rust.

- Para obter instruções sobre como instalar o AWS CLI, consulte o guia de instalação da AWS CLI.
- Para obter instruções sobre como configurar o AWS CLI, consulte Definindo configurações para o AWS CLI e Configurações e configurações do arquivo de credenciais no. AWS CLI
- Para obter mais informações sobre Rust, consulte rust-lang.org e o AWS SDK for Rust Developer Guide.

Com seu ambiente preparado, vamos explorar como implementar as permissões verificadas no Rust.

Teste o código de amostra

O código de amostra faz o seguinte:

- Configura o cliente SDK para se comunicar com AWS
- Cria um repositório de políticas
- Define a estrutura do repositório de políticas adicionando um esquema
- Adiciona uma política para verificar solicitações de autorização

- Envia uma solicitação de autorização de teste para verificar se tudo está configurado corretamente

Para testar o código de exemplo

1. Crie um projeto Rust.
2. Substitua qualquer código existente pelo código a seguir: `main.rs`

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

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

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

//Function that creates a policy in the policy store in the client
```

```
async fn create_policy(client: &Client, ps_id: &str,
 policy_definition:&PolicyDefinition) -> CreatePolicyOutput {
    let create_policy =
 client.create_policy().definition(policy_definition.clone()).policy_store_id(ps_id).send()
    return create_policy.unwrap();
}

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

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

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

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

//Add schema to policy store
    let schema= r#"{
        "PhotoFlash": {
            "actions": {
                "ViewPhoto": {
                    "appliesTo": {
                        "context": {
                            "type": "Record",
                            "attributes": {}
                        },
```

```
                        "principalTypes": [
                            "User"
                        ],
                        "resourceTypes": [
                            "Photo"
                        ]
                    },
                    "memberOf": []
                }
            },
            "entityTypes": {
                "Photo": {
                    "memberOfTypes": [],
                    "shape": {
                        "type": "Record",
                        "attributes": {
                            "IsPrivate": {
                                "type": "Boolean"
                            }
                        }
                    }
                },
                "User": {
                    "memberOfTypes": [],
                    "shape": {
                        "attributes": {},
                        "type": "Record"
                    }
                }
            }
        }
    }"#;
    let put_schema = put_schema(&client, &policy_store.policy_store_id,
 schema).await;
    println!(
        "Created Schema with Namespace: {:?}",
        put_schema.namespaces
    );

//Create policy
    let policy_text = r#"
        permit (
            principal in PhotoFlash::User::"alice",
            action == PhotoFlash::Action::"ViewPhoto",
```

```
            resource == PhotoFlash::Photo::"VacationPhoto94.jpg"
        );
        "#;
    let policy_definition =
 PolicyDefinition::Static(StaticPolicyDefinition::builder().statement(policy_text).build().
    let policy = create_policy(&client, &policy_store.policy_store_id,
 &policy_definition).await;
    println!(
        "Created Policy with ID: {:?}",
        policy.policy_id
    );

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

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

3. Execute o código entrando `cargo run` no terminal.

Se o código for executado corretamente, o terminal será exibido `Decision: Allow` seguido pelo ID da política determinante. Isso significa que você criou com sucesso um repositório de políticas e o testou usando o AWS SDK para Rust.

Limpar os recursos

Depois de terminar de explorar seu repositório de políticas, exclua-o.

Para excluir um armazenamento de políticas

Você pode excluir um repositório de políticas usando a `delete-policy-store` operação, *`PSEXAMPLEabcdefg111111`* substituindo-o pelo ID do repositório de políticas que você deseja excluir.

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

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

Armazenamentos de políticas vinculados à API

Um caso de uso comum é usar as Permissões Verificadas da Amazon para autorizar o acesso do usuário aos APIs hospedados no Amazon API Gateway. Usando um assistente no AWS console, você pode criar políticas de acesso baseadas em funções para usuários gerenciados no Amazon Cognito ou em qualquer provedor de identidade (IdP) do OIDC e AWS Lambda implantar um Autorizador que chame Permissões Verificadas para avaliar essas políticas.

Para concluir o assistente, escolha Configurar com API Gateway um provedor de identidade ao criar um novo repositório de políticas e siga as etapas.

Um repositório de políticas vinculado à API é criado e provisiona seu modelo de autorização e recursos para solicitações de autorização. O repositório de políticas tem uma fonte de identidade e um autorizador Lambda que se conecta API Gateway às Permissões Verificadas. Depois que o repositório de políticas for criado, você poderá 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.

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

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

Para ver uma demonstração, consulte Amazon Verified Permissions - Quick Start Overview and Demo no Amazon Web Services YouTube canal.

Tópicos

- Como as permissões verificadas autorizam solicitações de API
- Considerações sobre repositórios de políticas vinculados à API
- Adicionando controle de acesso baseado em atributos (ABAC)
- Passando para a produção com AWS CloudFormation
- Solução de problemas de repositórios de políticas vinculados à API

> **Important**
>
> Os repositórios de políticas que você cria com a opção Configurar com 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 (AWS CDK) Para obter mais informações, consulte Passando para a produção com AWS CloudFormation.

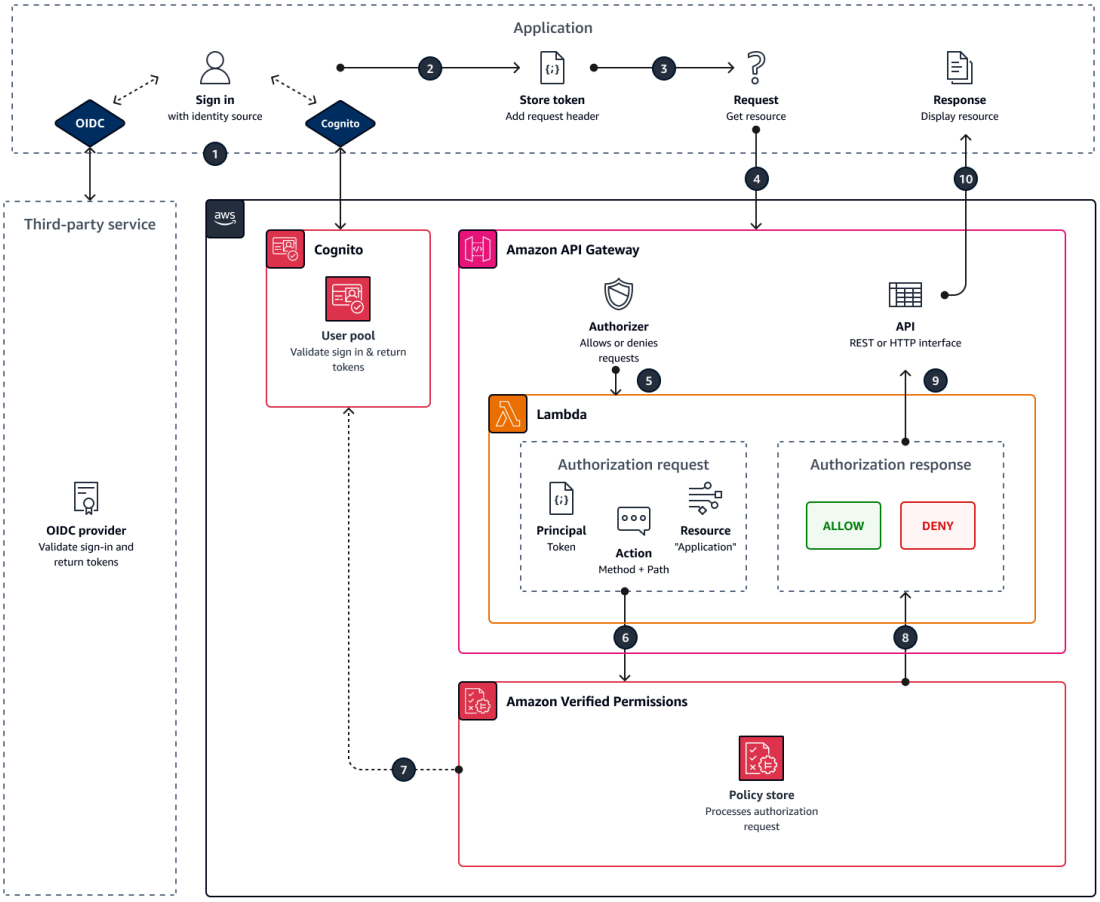
Em um repositório de políticas vinculado a uma API e a uma fonte de identidade, seu aplicativo apresenta um token de grupo de usuários em um cabeçalho de autorização ao fazer uma solicitação à API. A fonte de identidade do seu repositório de políticas fornece validação de token para permissões verificadas. 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 API Gateway autorização com permissões verificadas

- Um grupo de Amazon Cognitousuários 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 API GatewayREST. 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 para. API Gateway O fluxo de trabalho Configurar com API Gateway uma fonte de identidade 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 Amazon Cognito usuários ou grupo de provedores do OIDC. 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 API Gateway uma fonte de identidade, as Permissões verificadas criam políticas e esquemas de armazenamento de políticas. O esquema e as políticas refletem as ações da API e os grupos de usuários que você deseja autorizar a realizar as ações. As permissões verificadas também criam a função e o autorizador do Lambda.



1. Seu usuário faz login com seu aplicativo por meio de Amazon Cognito ou outro IdP do OIDC. O IdP emite tokens de ID e acesso com as informações do usuário.
2. Seu aplicativo armazena JWTs o. Para obter mais informações, consulte Uso de tokens com grupos de usuários no Guia do Amazon Cognito desenvolvedor.
3. Seu usuário solicita dados que seu aplicativo deve recuperar de uma API externa.
4. Seu aplicativo solicita dados de uma API REST em 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, API Gateway transmita 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 Amazon Cognito tokens são validados, consulte Autorização com permissões verificadas pela Amazon no Guia do Amazon Cognito desenvolvedor.
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 resposta `Allow` or `Deny` para. 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.

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:

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.

Amazon Cognito grupos 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 é iniciada na VPC padrão. Por padrão. APIs que têm acesso à rede restrito ao privado não VPCs 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 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.

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 em Amazon Cognito, consulte Como usar tokens com grupos de usuários no Guia do Amazon Cognito desenvolvedor.

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 ID

Os tokens de ID representam os atributos do usuário e têm um alto nível 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 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.

Consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens OIDC para o esquema, por exemplo, políticas que processam declarações em tokens de ID e acesso.

Passando para a produção com AWS CloudFormation

Os repositórios de políticas vinculados à API são uma forma de criar rapidamente um modelo de autorização para uma API Gateway API. 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: :Função,: AWS:: :Role,IAM::: :Policy,AWS::IAM: :Autorizador,AWS ApiGateway AWS::Lambda::Permission

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 API Gateway console, 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 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, consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens do OIDC para o esquema.

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.

Exclusão de armazenamentos de políticas

Você pode excluir os repositórios de políticas de permissões verificadas da Amazon usando o Console de gerenciamento da AWS ou AWS CLI o. A exclusão de um repositório de políticas exclui permanentemente o esquema e quaisquer políticas e modelos de políticas no repositório de políticas. Qualquer alias do repositório de políticas associado ao repositório de políticas excluído também será excluído.

A proteção contra exclusão impede a exclusão acidental de um repositório de políticas. A proteção contra exclusão está ativada em todos os novos repositórios de políticas criados por meio do Console de gerenciamento da AWS. Por outro lado, ele está desativado para todos os repositórios de políticas criados por meio de uma chamada de API ou SDK.

Talvez você queira excluir os repositórios de políticas pelos seguintes motivos:

- Você atingiu a cota de repositórios de políticas disponíveis em uma determinada região. Para obter mais informações, consulte Cotas para recursos.
- Você não está mais oferecendo suporte a um inquilino em um aplicativo multilocatário e, portanto, não precisa mais desse armazenamento de políticas.

Console de gerenciamento da AWS

Para excluir um armazenamento de políticas

1. Abra o console de Permissões verificadas. 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.

> **Note**
>
> Se a proteção contra exclusão estiver ativada, você precisará desativá-la antes de escolher Excluir. Para desativá-la, selecione Desativar proteção contra exclusão.

AWS CLI

Para excluir um armazenamento de políticas

Você pode excluir um repositório de políticas usando a `delete-policy-store` operação, *PSEXAMPLEabcdefg111111* substituindo-o pelo ID do repositório de políticas que você deseja excluir.

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

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

> **Note**
>
> Se a proteção contra exclusão estiver habilitada para esse repositório de políticas, você deverá primeiro executar a `update-policy-store` operação e desabilitar a proteção contra exclusão.
>
> ```
> aws verifiedpermissions update-policy-store \
>     --deletion-protection "DISABLED" \
>     --policy-store-id PSEXAMPLEabcdefg111111
> ```

Aliases da loja de políticas de permissões verificadas da Amazon

Um alias de armazenamento de políticas é um nome amigável para um repositório de políticas. Por exemplo, os aliases do repositório de políticas permitem que você se refira a um repositório de políticas usando `policy-store-alias/example-policy-store` em vez de`PSEXAMPLEabcdefg111111`. Os aliases do repositório de políticas podem ser usados em qualquer operação de Permissões Verificadas que aceite um parâmetro `policyStoreId` de entrada.

Você pode criar um alias de armazenamento de políticas para um repositório de políticas usando a `CreatePolicyStoreAlias` API ou usando o `AWS::VerifiedPermissions::PolicyStoreAlias` CloudFormation recurso.

A API de permissões verificadas da Amazon fornece controle total dos aliases do armazenamento de políticas em cada Conta da AWS região. A API inclui operações para criar um alias do repositório de políticas (`CreatePolicyStoreAlias`), visualizar nomes de alias do repositório de políticas e ARNs do alias do repositório de políticas (`GetPolicyStoreAlias`,`ListPolicyStoreAliases`) e excluir um alias do repositório de políticas (). `DeletePolicyStoreAlias`

Tópicos

- Propriedades dos aliases do repositório de políticas
- Criação de aliases de armazenamento de políticas de permissões verificadas da Amazon
- Recuperação de aliases da loja de políticas de permissões verificadas da Amazon
- Excluindo aliases da loja de políticas de permissões verificadas da Amazon
- Usando a política de permissões verificadas da Amazon, armazene aliases em operações de API
- Controle do acesso aos aliases do repositório de políticas

Propriedades dos aliases do repositório de políticas

Como os aliases de armazenamento de políticas funcionam nas Permissões Verificadas da Amazon.

Um alias de armazenamento de políticas é independente AWS recurso

Um alias de armazenamento de políticas não é propriedade de um repositório de políticas. As ações que você executa no alias do repositório de políticas não afetam o armazenamento de políticas

associado. Você pode excluir o alias do repositório de políticas sem nenhum efeito no repositório de políticas associado. Se você excluir um repositório de políticas, todos os aliases do repositório de políticas associados a esse repositório de políticas também serão excluídos.

Cada alias do repositório de políticas tem um Amazon Resource Name (ARN) que identifica exclusivamente o alias do repositório de políticas. Se você especificar um alias do repositório de políticas como o recurso em uma política do IAM, a política se referirá ao alias do armazenamento de políticas, não ao armazenamento de políticas associado.

Cada alias do repositório de políticas tem dois formatos

Ao criar um alias do repositório de políticas, você especifica o nome do alias do repositório de políticas. O Amazon Verified Permissions cria o alias ARN do repositório de políticas para você.

- Um alias de armazenamento de políticas ARN é um nome de recurso da Amazon (ARN) que identifica exclusivamente o alias do repositório de políticas.

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

- Um nome de alias do repositório de políticas que é exclusivo na região Conta da AWS e. Na API de permissões verificadas da Amazon, o nome do alias do repositório de políticas é sempre prefixado por. `policy-store-alias/`

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

Os aliases do repositório de políticas não são secretos

Os aliases do repositório de políticas podem ser exibidos em texto simples em CloudTrail registros e outras saídas. Não inclua informações confidenciais ou sigilosas no nome do alias do repositório de políticas.

Cada alias do repositório de políticas está associado a um repositório de políticas por vez

O alias do repositório de políticas e seu repositório de políticas associado devem pertencer à mesma Conta da AWS região. Você pode associar um alias do repositório de políticas a qualquer repositório de políticas na mesma Conta da AWS região.

Por exemplo, essa `ListPolicyStoreAliases` saída mostra que o alias do repositório de `example-policy-store` políticas está associado a exatamente um repositório de políticas de destino, representado pela `policyStoreId` propriedade.

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

Vários aliases podem ser associados ao mesmo repositório de políticas

Por exemplo, você pode associar os `example-policy-store-2` aliases `example-policy-store` e ao mesmo repositório de políticas.

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

Um alias de armazenamento de políticas deve ser exclusivo em um Conta da AWS e região

Por exemplo, você pode ter somente um alias de armazenamento de políticas com o nome `example-policy-store` em cada Conta da AWS região. Os aliases do repositório de políticas diferenciam maiúsculas de minúsculas. Você não pode alterar o nome do alias de um repositório de

políticas. No entanto, você pode excluir o alias do repositório de políticas e criar um novo alias do repositório de políticas com o nome desejado após o término do período de reserva de 24 horas.

Você pode criar aliases de armazenamento de políticas com o mesmo nome em diferentes regiões. Cada alias de armazenamento de políticas terá um ARN exclusivo. Se seu código se referir a um alias de armazenamento de políticas`policy-store-alias/example-policy-store`, como, você poderá executá-lo em várias regiões. Em cada região, ele usa um repositório de políticas diferente.

Os aliases do repositório de políticas oferecem suporte a exclusões reversíveis e definitivas

Por padrão, quando um alias do repositório de políticas é excluído, ele é excluído temporariamente. O nome do alias do repositório de políticas é reservado por um período de 24 horas. Se você tentar criar um alias de armazenamento de políticas com o mesmo nome durante esse período, a solicitação será rejeitada. Durante esse período, `GetPolicyStoreAlias` retorna o alias do repositório de políticas com o `PendingDeletion` estado. Você também pode excluir definitivamente um alias do repositório de políticas especificando `HardDelete` como o. `deletionMode` Um alias do repositório de políticas excluído permanentemente é imediatamente excluído e o nome do alias do repositório de políticas fica disponível para reutilização imediata. Para obter mais informações, consulte Excluindo aliases da loja de políticas de permissões verificadas da Amazon.

Você pode usar aliases para identificar repositórios de políticas

Você pode usar um alias de armazenamento de políticas para identificar um repositório de políticas em todas as operações que aceitam um `policyStoreId` (por exemplo,`IsAuthorized`). Nesses casos, o nome do alias do repositório de políticas deve ser prefixado com. `policy-store-alias/` Os aliases do repositório de políticas não podem ser usados para identificar um repositório de políticas para a `DeletePolicyStore` operação.

Você não pode usar um nome de alias do repositório de políticas ou o ARN do alias do repositório de políticas para identificar um repositório de políticas no `Resource` elemento de uma política do IAM. Para controlar o acesso a um repositório de políticas quando ele é referenciado por meio de um alias do repositório de políticas, consulte. Controle do acesso aos aliases do repositório de políticas

Criação de aliases de armazenamento de políticas de permissões verificadas da Amazon

Você pode criar um alias de armazenamento de políticas para referenciar um repositório de políticas usando um nome amigável. O nome de um alias do repositório de políticas deve ser exclusivo por

Conta da AWS região. Os aliases do repositório de políticas só podem ser associados a repositórios de políticas pertencentes à mesma Conta da AWS e ativos na mesma região do alias do repositório de políticas. Os aliases do repositório de políticas são recursos separados com autorização própria ARNs e do IAM.

Por padrão, somente 10 aliases do repositório de políticas podem ser associados ao mesmo repositório de políticas.

> **Note**
>
> `CreatePolicyStoreAlias`é idempotente. Se você chamar a `CreatePolicyStoreAlias` operação com um nome de alias de armazenamento de políticas e ID de armazenamento de políticas que correspondam a um alias de armazenamento de políticas existente, a `CreatePolicyStoreAlias` operação será bem-sucedida e retornará o alias de armazenamento de políticas existente. No entanto, se você chamar a `CreatePolicyStoreAlias` operação com um nome de alias do repositório de políticas existente, mas com uma ID de armazenamento de políticas diferente, a operação retornará a. `ConflictException`

AWS CLI

Para criar um alias de armazenamento de políticas

Você pode criar um alias de armazenamento de políticas usando a CreatePolicyStoreAliasoperação. O exemplo a seguir cria um alias de armazenamento de políticas com o nome`example-policy-store`.

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

Recuperação de aliases da loja de políticas de permissões verificadas da Amazon

Você pode recuperar informações sobre aliases de armazenamento de políticas usando a `GetPolicyStoreAlias` operação para obter detalhes sobre um alias de armazenamento de políticas específico ou a `ListPolicyStoreAliases` operação para listar todos os aliases de armazenamento de políticas em sua região. Conta da AWS

Obtendo um alias de armazenamento de políticas

Use a `GetPolicyStoreAlias` operação para recuperar detalhes sobre um alias específico do repositório de políticas, incluindo o ID do repositório de políticas associado.

AWS CLI

Para recuperar detalhes sobre um alias do repositório de políticas

Você pode recuperar um alias do repositório de políticas usando a GetPolicyStoreAliasoperação. O exemplo a seguir recupera detalhes de um alias de armazenamento de políticas com o nome. `example-policy-store`

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

Listando aliases do repositório de políticas

Use a `ListPolicyStoreAliases` operação para listar todos os aliases do repositório de políticas em sua Conta da AWS região. Você pode usar o `filter` parâmetro para listar somente os aliases do repositório de políticas associados a um repositório de políticas específico.

AWS CLI

Para listar todos os aliases do repositório de políticas

Você pode listar os aliases do repositório de políticas usando a ListPolicyStoreAliasesoperação. O exemplo a seguir lista todos os aliases do repositório de políticas pertencentes ao Conta da AWS 123456789012 na região us-west-2.

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

Para listar os aliases do repositório de políticas para um repositório de políticas específico

Use o `filter` parâmetro para listar somente aliases associados a um repositório de políticas específico.

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

Excluindo aliases da loja de políticas de permissões verificadas da Amazon

Você pode excluir um alias do repositório de políticas quando ele não for mais necessário. A exclusão de um alias do repositório de políticas não afeta o repositório de políticas associado. A exclusão de um repositório de políticas exclui todos os aliases do repositório de políticas associados a esse repositório de políticas.

O Amazon Verified Permissions oferece suporte a dois modos de exclusão para aliases do repositório de políticas:

- Exclusão temporária (padrão): o alias do repositório de políticas entra em um `PendingDeletion` estado. O nome do alias do repositório de políticas é reservado por 24 horas e não pode ser reutilizado durante esse período. Durante esse período, `GetPolicyStoreAlias` retorna o alias do repositório de políticas com o `PendingDeletion` estado. Esse é o comportamento padrão quando você não especifica um `deletionMode` ou quando especifica`SoftDelete`.

- Exclusão definitiva: o alias do repositório de políticas é excluído imediatamente. O nome do alias do repositório de políticas fica disponível para reutilização imediata. Para realizar uma exclusão definitiva, especifique `HardDelete` como `deletionMode` o.

Exclusão reversível de um alias do repositório de políticas

Por padrão, a exclusão de um alias do repositório de políticas executa uma exclusão temporária. O alias do repositório de políticas entra no `PendingDeletion` estado e o nome do alias do repositório de políticas é reservado por 24 horas.

AWS CLI

Para excluir de forma reversível um alias do repositório de políticas

Você pode excluir de forma reversível um alias do repositório de políticas usando a DeletePolicyStoreAliasoperação. O exemplo a seguir exclui temporariamente um alias de armazenamento de políticas com o nome. `example-policy-store`

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

Você também pode especificar explicitamente o modo de exclusão reversível.

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

Exclusão definitiva de um alias do repositório de políticas

Para excluir imediatamente um alias do repositório de políticas, especifique `HardDelete` como o. `deletionMode` Um alias do repositório de políticas excluído permanentemente não entra no `PendingDeletion` estado e o nome do alias do repositório de políticas fica disponível para reutilização imediata.

Se você excluir definitivamente um alias do repositório de políticas que foi excluído temporariamente anteriormente, o alias do repositório de políticas será excluído imediatamente.

> **Important**
>
> Eventualmente, as permissões verificadas da Amazon são consistentes. Se você excluir definitivamente um alias do repositório de políticas e recriá-lo imediatamente para apontar para outro repositório de políticas, as solicitações que fazem referência ao alias do repositório de políticas poderão continuar sendo resolvidas no repositório de políticas anteriormente associado por um curto período de tempo. Para evitar resultados inesperados de autorização, aguarde até que a exclusão se propague antes de recriar um alias de armazenamento de políticas com o mesmo nome.

AWS CLI

Para excluir definitivamente um alias do repositório de políticas

Você pode excluir definitivamente um alias do repositório de políticas usando a DeletePolicyStoreAliasoperação com o `deletion-mode` parâmetro definido como. `HardDelete` O exemplo a seguir exclui imediatamente um alias do repositório de políticas com o nome. `example-policy-store`

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

Usando a política de permissões verificadas da Amazon, armazene aliases em operações de API

Qualquer operação de permissões verificadas da Amazon que aceite um `policyStoreId` parâmetro, como IsAuthorized, e IsAuthorizedWithToken GetPolicyStore, pode aceitar um nome de alias de armazenamento de políticas no lugar do ID do armazenamento de políticas.

> **Important**
>
> Ao usar um alias de armazenamento de políticas como o valor de um `policyStoreId` parâmetro, você deve incluir o `policy-store-alias/` prefixo. Por exemplo, use`policy-store-alias/example-policy-store`, não`example-policy-store`.

Usando aliases do Policy Store em operações

O `IsAuthorized` comando a seguir usa um alias de armazenamento de políticas com o nome `example-policy-store` para identificar um repositório de políticas.

AWS CLI

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

> **Note**
>
> Você não pode usar um alias de armazenamento de políticas no lugar do `policyStoreId` campo para a DeletePolicyStore operação.

Usando aliases do Policy Store Across Regiões da AWS

Um dos usos mais poderosos dos aliases é em aplicações executadas em várias Regiões da AWS. Por exemplo, você pode ter um aplicativo global que usa diferentes repositórios de políticas em cada região.

- Em us-east-1, você deseja usar. `PSEXAMPLEabcdefg111111`
- Em eu-west-1, você deseja usar. `PSEXAMPLEabcdefg222222`

Você pode criar uma versão diferente do seu aplicativo em cada região ou usar um dicionário ou uma instrução switch para selecionar o repositório de políticas certo para cada região. Mas é muito mais fácil criar um alias de armazenamento de políticas com o mesmo nome de alias de armazenamento de políticas em cada região. Lembre-se de que o nome do alias do repositório de políticas diferencia maiúsculas de minúsculas.

AWS CLI

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

```
    --policy-store-id PSEXAMPLEabcdefg111111

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

Em seguida, use o alias do repositório de políticas em seu código. Quando seu código é executado em cada região, o alias do repositório de políticas se referirá ao repositório de políticas associado nessa região.

AWS CLI

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

No entanto, existe o risco de que o alias do repositório de políticas seja excluído. Nesse caso, as tentativas do aplicativo de usar o nome do alias do repositório de políticas falharão e talvez seja necessário recriar ou atualizar o alias do repositório de políticas. Para reduzir esse risco, tenha cuidado ao conceder permissão aos diretores para gerenciar os aliases do repositório de políticas que você usa em seu aplicativo.

Os aliases do repositório de políticas não são um mecanismo de controle de tráfego

Um alias de armazenamento de políticas é um nome estável e amigável para um repositório de políticas. Não é um mecanismo para transferir, dividir ou ponderar o tráfego de autorização entre repositórios de políticas. Se você estiver familiarizado com os recursos que direcionam o tráfego entre as versões de um recurso, como aliases do Lambda, observe que os aliases do repositório de políticas não se comportam da mesma maneira. Por design, um alias de armazenamento de políticas está mais próximo de um alias de AWS KMS chave: ele fornece um nome durável para um recurso, não uma camada de roteamento na frente dele.

Por esse motivo, intencionalmente não fornecemos uma `UpdatePolicyStoreAlias` operação. Para alterar o repositório de políticas para o qual um alias de armazenamento de políticas aponta, você exclui o alias do armazenamento de políticas e cria um novo alias de armazenamento de

políticas que tenha o mesmo nome e tenha como alvo um repositório de políticas diferente. Essa não é uma operação atômica e não oferece as garantias que um mecanismo de controle de tráfego forneceria.

Quando você altera o repositório de políticas para o qual um alias de armazenamento de políticas resolve, a alteração não entra em vigor em todos os lugares no mesmo instante:

- O mapeamento atualizado deve se propagar do plano de controle para o plano de dados que avalia as solicitações de autorização. O mapeamento não se propaga instantaneamente.
- Como a resolução de alias do repositório de políticas é eventualmente consistente e pode ser armazenada em cache, existe uma janela de transição após uma alteração. Durante essa janela, o repositório de políticas associado anteriormente atende a algumas solicitações e o repositório de políticas recém-associado atende a outras solicitações. O comprimento dessa janela é indeterminado.

Durante essa janela, seu aplicativo pode receber decisões de autorização de qualquer repositório de políticas. Como diferentes repositórios de políticas podem conter políticas, entidades e esquemas diferentes, a mesma solicitação pode produzir decisões diferentes, dependendo de qual repositório de políticas a atende. Os aliases do repositório de políticas, portanto, não podem fornecer uma transição atômica entre os repositórios de políticas e não substituem uma estratégia de implantação ou gerenciamento de tráfego.

⚠ Não use aliases como mecanismo de substituição

Não crie uma infraestrutura como código ou uma primitiva de implantação de alto nível que redirecione um alias de armazenamento de políticas para alternar o tráfego de autorização de um repositório de políticas para outro. Como a alteração não é atômica, as solicitações podem ser autorizadas nos dois repositórios de políticas ao mesmo tempo por um período indeterminado. Isso pode levar a decisões de autorização inconsistentes. Para alternar os repositórios de políticas com segurança, consulteRealizando alterações no repositório de políticas sem tempo de inatividade.

Realizando alterações no repositório de políticas sem tempo de inatividade

Como os aliases do repositório de políticas não fornecem uma transferência atômica (consulteOs aliases do repositório de políticas não são um mecanismo de controle de tráfego), recomendamos

que você evite migrar de um repositório de políticas para outro reposicionando um alias do repositório de políticas. Em vez disso, controle a migração de dentro do seu aplicativo para que você possa validar o novo repositório de políticas e reverter instantaneamente, se necessário. A abordagem a seguir permite que você alterne os repositórios de políticas sem tempo de inatividade:

1. Crie o novo repositório de políticas e dê a ele seu próprio alias de armazenamento de políticas, para que o repositório de políticas atual e o novo repositório de políticas tenham um nome distinto e estável. Por exemplo, continue `policy-store-alias/example-policy-store` apontando para seu repositório de políticas atual e crie `policy-store-alias/example-policy-store-2` para o novo repositório de políticas. Não reutilize nem reaponte um único alias de armazenamento de políticas para realizar a troca.
2. Replique suas políticas, esquemas e qualquer outra configuração no novo repositório de políticas e execute-os em paralelo com o armazenamento de políticas atual.
3. Adicione um valor de configuração ou sinalizador de recurso ao seu aplicativo (por exemplo, usando AppConfig) que determine qual alias de armazenamento de políticas é autoritário para decisões de autorização. Não confie na alteração do alias em si para alternar o tráfego.
4. Execute o novo repositório de políticas no modo sombra. Envie o mesmo `IsAuthorized` ou `IsAuthorizedWithToken` solicite para os dois repositórios de políticas e compare as decisões. Registre e investigue quaisquer discrepâncias até que o novo repositório de políticas retorne as decisões que você espera.
5. Use o sinalizador de recurso para transferir gradualmente as decisões de autorização para o novo alias do repositório de políticas, por exemplo, em uma distribuição em fases em seus hosts de aplicativos ou segmentos de usuários. Monitore as decisões de autorização e as taxas de erro à medida que você prossegue.
6. Use o sinalizador de recurso para voltar imediatamente ao alias original do repositório de políticas se você detectar um problema. Como seu aplicativo controla o switch, a reversão é instantânea e não depende da propagação do alias.
7. Desative o antigo repositório de políticas e seu alias do repositório de políticas depois que o novo repositório de políticas estiver totalmente validado e atender a todo o tráfego.

Esse padrão dá ao seu aplicativo, em vez da propagação de aliases, o controle sobre qual repositório de políticas atende a cada solicitação. Esse controle é o que torna a migração segura e reversível e evita a janela de transição que o reposicionamento de um alias de armazenamento de políticas introduziria.

Controle do acesso aos aliases do repositório de políticas

Os diretores que gerenciam os aliases do repositório de políticas devem ter permissão para interagir com esses aliases do repositório de políticas e, para algumas operações, com o repositório de políticas ao qual o alias do repositório de políticas está associado. Você pode fornecer essas permissões usando IAM políticas.

As seções a seguir descrevem as permissões necessárias para criar e gerenciar aliases do repositório de políticas.

permissões verificadas: CreatePolicyStoreAlias

Para criar um alias do repositório de políticas, o diretor precisa das seguintes permissões para o alias do repositório de políticas e para o repositório de políticas associado.

- `verifiedpermissions:CreatePolicyStoreAlias`para o alias do repositório de políticas. Forneça essa permissão em uma IAM política anexada ao principal que tem permissão para criar o alias do repositório de políticas.

  O exemplo de declaração de política a seguir especifica um alias de armazenamento de políticas específico em um `Resource` elemento. Mas você pode listar vários alias do repositório de políticas ARNs ou especificar um padrão de alias do repositório de políticas, como. `"sample*"` Você também pode especificar um `Resource` valor de `"*"` para permitir que o principal crie qualquer alias de armazenamento de políticas na região Conta da AWS e.

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

- `verifiedpermissions:CreatePolicyStoreAlias`para o repositório de políticas associado. Essa permissão deve ser fornecida em uma IAM política.

  ```
  {
    "Sid": "PolicyStorePermissionForAlias",
    "Effect": "Allow",
    "Action": "verifiedpermissions:CreatePolicyStoreAlias",
  ```

```
  "Resource": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
}
```

permissões verificadas: GetPolicyStoreAlias

Para obter detalhes sobre um alias de armazenamento de políticas específico, o diretor deve ter `verifiedpermissions:GetPolicyStoreAlias` permissão para o alias do repositório de políticas em uma IAM política.

O exemplo de declaração de política a seguir dá ao principal permissão para obter um alias de armazenamento de políticas específico.

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

permissões verificadas: ListPolicyStoreAliases

Para listar os aliases do repositório de políticas na região Conta da AWS e, o diretor deve ter `verifiedpermissions:ListPolicyStoreAliases` permissão em uma IAM política. Como essa política não está relacionada a nenhum repositório de políticas específico ou recurso de alias do repositório de políticas, o valor do elemento de recurso na política deve ser`"*"`.

Por exemplo, a declaração de IAM política a seguir dá ao principal permissão para listar todos os aliases do repositório de políticas no Conta da AWS.

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

permissões verificadas: DeletePolicyStoreAlias

Para excluir um alias do repositório de políticas, o diretor precisa de permissão apenas para o alias do repositório de políticas.

> **Note**
>
> A exclusão de um alias do repositório de políticas não tem efeito no repositório de políticas associado, embora os aplicativos que fazem referência ao alias do repositório de políticas recebam erros. Se você excluir por engano um alias do repositório de políticas, poderá recriá-lo após o período de reserva de 24 horas.

O diretor precisa de `verifiedpermissions:DeletePolicyStoreAlias` permissão para o alias do repositório de políticas. Forneça essa permissão em uma IAM política anexada ao principal que tem permissão para excluir o alias do repositório de políticas.

O exemplo de declaração de política a seguir especifica o alias do repositório de políticas em um `Resource` elemento. Mas você pode listar vários alias do repositório de políticas ARNs ou especificar um padrão de alias do repositório de políticas, como. `"sample*"` Você também pode especificar um `Resource` valor de `"*"` para permitir que o principal exclua qualquer alias do repositório de políticas na região Conta da AWS e.

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

Limitando as permissões do alias do repositório de políticas

Você pode usar um alias de armazenamento de políticas para referenciar um repositório de políticas em qualquer operação que aceite um `policyStoreId` campo como entrada. Quando você faz isso, o Amazon Verified Permissions autoriza `verifiedpermissions:GetPolicyStoreAlias` o alias do repositório de políticas e a operação solicitada no armazenamento de políticas associado.

Por exemplo, se a `IsAuthorized` operação for executada usando um alias de armazenamento de políticas, o principal precisará de ambos:

- `verifiedpermissions:GetPolicyStoreAlias`permissão para o alias do repositório de políticas
- `verifiedpermissions:IsAuthorized`permissão para o repositório de políticas associado

O exemplo de política a seguir concede permissão para chamar `IsAuthorized` usando um alias específico do repositório de políticas.

```
{
  "Sid": "IAMPolicyForAliasUsage",
  "Effect": "Allow",
  "Action": "verifiedpermissions:GetPolicyStoreAlias",
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/
example-policy-store"
},
{
  "Sid": "IAMPolicyForPolicyStoreOperation",
  "Effect": "Allow",
  "Action": "verifiedpermissions:IsAuthorized",
  "Resource": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
}
```

Para limitar quais aliases de armazenamento de políticas um diretor pode usar, restrinja a `verifiedpermissions:GetPolicyStoreAlias` permissão. Por exemplo, a política a seguir permite que o diretor use qualquer alias do repositório de políticas, exceto aqueles que começam com`Restricted`.

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

```
  "Resource": "arn:aws:verifiedpermissions:us-east-1:123456789012:policy-store-alias/
Restricted*"
}
```

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 ver a diferença entre como as Permissões Verificadas e o Cedar lidam com esquemas, consulte. Suporte ao esquema

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.

Console de gerenciamento da AWS

Para criar um esquema

1. Abra o console de Permissões verificadas. 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 usando o 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.

Editando esquemas de armazenamento de políticas

Quando você seleciona Esquema no console de permissões verificadas da Amazon, os tipos de entidade e ações que compõem seu esquema são exibidos. Você pode visualizar e editar seu esquema no modo Visual ou no modo JSON. O modo visual permite que você atualize o esquema adicionando novos tipos e ações usando vários assistentes. Usando o modo JSON, você pode começar a atualizar o código JSON do esquema diretamente no editor JSON.

Visual Mode

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 dos diagramas. Há dois diagramas disponíveis:

- 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 entidades — 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 de Permissões verificadas. 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.

JSON mode

Ao fazer atualizações, você notará que o editor JSON valida seu código em relação à sintaxe JSON e identificará erros e avisos à medida que você edita, facilitando a localização rápida de problemas. Além disso, você não precisa se preocupar com a formatação do JSON, basta escolher Formatar JSON depois de fazer as atualizações e o formato será atualizado para corresponder à formatação JSON esperada.

Para editar um esquema do Verified Permissions no modo JSON

1. Abra o console de Permissões verificadas. 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.

Ativando o modo de validação da política de permissões verificadas da Amazon

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. As Permissões verificadas rejeitam a tentativa de solicitação se a validação falhar. Por esse motivo, recomendamos deixar a validação desativada durante o desenvolvimento do aplicativo, ativá-la para testes e deixá-la ativada enquanto o aplicativo estiver em produção.

Console de gerenciamento da AWS

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

1. Abra o console de Permissões verificadas. 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 todas as outras políticas. 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.

Opcionalmente, você pode atribuir um nome de política a uma política. Os nomes das políticas devem ser exclusivos para todas as políticas no repositório de políticas e prefixados com`name/`. Você pode usar um nome de política no lugar do ID da política em operações do plano de controle que aceitam um `policyId` parâmetro. O exemplo a seguir usa um nome de política para recuperar uma política com`GetPolicy`.

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

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 identificadores universalmente exclusivos (UUIDs) para todos os identificadores 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 == Role: :"A1B2C3D4-E5F6-A1B2-C3D4-Example11111",//administradores

- 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.

Tópicos

- Criação de políticas estáticas do Amazon Verified Permissions
- Edição de políticas estáticas do Amazon Verified Permissions
- Adicionando contexto
- Usando a bancada de testes de permissões verificadas da Amazon
- Tamanho da política por recurso
- Exemplos de políticas do Amazon Verified Permissions

Criação de políticas estáticas do Amazon Verified Permissions

Você pode criar uma política estática para os diretores para permitir ou proibir que eles executem ações específicas em recursos específicos para seu aplicativo. Uma política estática tem valores específicos incluídos para `principal` e `resource` e está pronta para ser usada em decisões de autorização.

Console de gerenciamento da AWS

Para criar uma política estática

1. Abra o console de Permissões verificadas. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Políticas.
3. Selecione Criar política e escolha Criar política estática.

> **Note**
>
> Se você tiver uma declaração de política que gostaria de usar, vá para a Etapa 8 e cole a política na seção Política na próxima página.

4. Na seção Efeito da política, escolha se a política permitirá ou proibirá quando uma solicitação corresponder à política. Se você escolher Permitir, a política permitirá que os diretores executem as ações nos recursos. Por outro lado, se você escolher Proibir, a política não permitirá que os diretores executem as ações nos recursos.
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 do principal que terá permissão ou 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. Selecione 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"
}
```

Para criar uma política com um nome de política

Opcionalmente, você pode especificar um nome de política ao criar uma política. O nome deve ser exclusivo para todas as políticas no repositório de políticas e prefixado com`name/`. Você pode usar o nome no lugar do ID da política.

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

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

> **Note**
>
> Se você especificar um nome que já esteja associado a outra política no repositório de políticas, receberá um `ConflictException` erro.

Edição de políticas estáticas do Amazon Verified Permissions

Você pode editar uma política estática existente no seu repositório de políticas. Você só pode atualizar diretamente as políticas estáticas. Para alterar uma política vinculada a um modelo, você deve atualizar o modelo de política. Para obter mais informações, consulte Editando modelos de política de permissões verificadas da Amazon.

Você pode alterar os seguintes elementos de uma política estática:

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

Você não pode alterar os seguintes elementos de uma política estática. Para alterar qualquer um desses elementos, você precisará excluir e recriar a política.

- Uma política de uma política estática para uma política vinculada a um modelo.
- O efeito de uma política estática de `permit` ou`forbid`.
- O `principal` referenciado por uma política estática.
- O `resource` referenciado por uma política estática.

Console de gerenciamento da AWS

Para editar uma política estática

1. Abra o console de Permissões verificadas. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha 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"
    }
}
```

Para atualizar o nome de uma política

Você pode definir ou atualizar o nome de uma política ao atualizar uma política. O nome deve ser exclusivo para todas as políticas no repositório de políticas e prefixado com`name/`. Se você não incluir o campo de nome na solicitação de atualização, o nome existente permanecerá inalterado. Para remover um nome, defina-o como uma string vazia.

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

```
        "entityType": "UserGroup"
    },
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

Adicionando contexto

O contexto é a informação relevante para as decisões políticas, mas não faz parte da identidade de seu diretor, ação ou recurso. A reivindicação do token de acesso é contextual. 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 estar JSON-formatted em um `contextMap` elemento.

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": {
    "AccountCodes": {
      "set": [
        {
          "long": 111122223333
        },
        {
          "long": 444455556666
        },
        {
          "long": 123456789012
        }
      ]
    },
    "approvedBy": {
    "entityIdentifier": {
      "entityId": "Bob",
```

```
      "entityType": "DigitalPetStore::User"
    }
    },
    "MfaAuthorized": {
      "boolean": true
    },
    "NetworkInfo": {
      "record": {
        "IPAddress": {
          "string": "192.0.2.178"
        },
        "Country": {
          "string": "United States of America"
        },
        "SSL": {
          "boolean": true
        }
    }
    },
    "RequestedOrderCount": {
      "long": 4
    },
    "UserAgent": {
      "string": "My UserAgent 1.12"
    }
  }
}
```

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. Para exemplos de contexto com informações de token de acesso como contexto, consulte Mapeamento de tokens de Amazon Cognito acesso e Mapeamento de tokens de acesso OIDC.

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

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

```
"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.AccountCodes.contains(111122223333) &&
    context.approvedBy in DigitalPetStore::Role::"Employee" &&
    context.MfaAuthorized == true &&
    context.NetworkInfo.Country like "*United States*" &&
```

```
    context.NetworkInfo.IPAddress like "192.0.2.*" &&
    context.NetworkInfo.SSL == true &&
    context.RequestedOrderCount <= 4 &&
    context.UserAgent like "*My UserAgent*"
};
```

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 da América, 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": {
      "AccountCodes": {
        "set": [
          {"long": 111122223333},
          {"long": 444455556666},
          {"long": 123456789012}
        ]
      },
      "approvedBy": {
        "entityIdentifier": {
          "entityId": "Bob",
          "entityType": "DigitalPetStore::User"
        }
      },
      "MfaAuthorized": {
        "boolean": true
      },
      "NetworkInfo": {
        "record": {
          "Country": {"string": "United States of America"},
          "IPAddress": {"string": "192.0.2.178"},
          "SSL": {"boolean": true}
        }
      },
      "RequestedOrderCount":{
        "long": 4
      },
      "UserAgent": {
        "string": "My UserAgent 1.12"
      }
    }
  },
  "entities": {
    "entityList": [
      {
        "identifier": {
          "entityType": "DigitalPetStore::User",
          "entityId": "Alice"
        },
        "attributes": {
          "memberId": {
            "string": "801b87f2-1a5c-40b3-b580-eacad506d4e6"
          }
        },
```

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

Usando a bancada de testes de permissões verificadas da Amazon

Use a bancada de testes de Permissões Verificadas para testar e solucionar problemas de políticas de Permissões Verificadas executando solicitações de autorização em relação a elas. 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, consulte Amazon Verified Permissions - Policy Creation and Testing (Primer Series #3) no AWS YouTube canal.

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 de Permissões verificadas. 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 de Permissões verificadas. 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.

Tamanho da política por recurso

As permissões verificadas limitam o tamanho total de todas as políticas que fazem referência ao mesmo recurso. Esse limite é o tamanho da política por cota de recursos e seu valor padrão é 200.000 bytes. Quando você liga para `CreatePolicy` ou`UpdatePolicy`, as Permissões

verificadas calculam o novo total do recurso referenciado. Se o novo total exceder a cota, a solicitação falhará com um`ServiceQuotaExceededException`.

As políticas que não especificam um recurso se aplicam a todos os recursos. Essas políticas compartilham um total separado para o `"unspecified"` recurso, que é limitado ao mesmo valor da cota.

Para ver a lista de todas as cotas de Permissões Verificadas, consulteCotas do Amazon Verified Permissions.

Tópicos

- Como os tamanhos das apólices são calculados
- Permanecendo dentro da cota
- Template-linked exemplo de tamanho de política

Como os tamanhos das apólices são calculados

Cada política contribui para o total de um recurso da seguinte forma:

- Políticas estáticas — Uma política estática contribui com seu tamanho total. Esse é o mesmo valor medido em relação à cota de tamanho da política.
- Template-linked políticas — Uma política vinculada a um modelo contribui com o tamanho combinado do principal e do recurso usado para instanciá-la, bem como com alguma sobrecarga fixa armazenada no item do banco de dados. Se um modelo não tiver espaço principal ou de recurso, o comprimento dessa entidade não especificada será contado como 0. O tamanho do corpo do modelo de política em si não conta para essa cota.

Cada política também adiciona uma sobrecarga fixa de até algumas centenas de bytes ao total. Template-linked as políticas adicionam mais sobrecarga do que as políticas estáticas.

Permanecendo dentro da cota

Se seu aplicativo se aproximar do tamanho da política por cota de recursos, considere as seguintes estratégias:

- Use o tipo de política apropriado. Se muitas políticas concederem permissões semelhantes no mesmo recurso, os modelos de política poderão reduzir o consumo de armazenamento do seu

repositório de políticas. Cada política vinculada ao modelo armazena somente seu principal e seu recurso. O modelo de política armazena a declaração compartilhada uma vez, mas cada política vinculada também carrega uma sobrecarga fixa de armazenamento. Os modelos só se pagam quando a declaração compartilhada é grande o suficiente para compensar essa sobrecarga, aproximadamente 200 bytes ou mais. Abaixo desse limite, as políticas estáticas individuais consomem menos espaço. Para obter mais informações, consulte Modelos de política de permissões verificadas da Amazon e políticas vinculadas a modelos.

- Especifique princípios e recursos no escopo da política. As políticas que não especificam um recurso compartilham um único total do `"unspecified"` recurso em todo o repositório de políticas. O escopo de cada política para um recurso específico distribui suas políticas em totais separados, um para cada recurso.
- Agrupe seus diretores e recursos. Em vez de escrever uma política para cada diretor, escreva uma política para um grupo e adicione diretores ao grupo. O agrupamento de recursos em contêineres também pode reduzir o número de políticas que fazem referência a um único recurso.
- Solicite um aumento da cota. Você pode solicitar um aumento dessa cota se o design da política atender a determinadas restrições. Para se qualificar, suas políticas devem especificar o principal ou o recurso no escopo da política. Um aumento também pode reduzir o número de pais hierárquicos que seus diretores e recursos podem ter. Mesmo com uma cota maior, um sublimite de 200.000 bytes ainda se aplica às políticas que fazem referência ao mesmo princípio e ao mesmo recurso. Você não pode solicitar um aumento para esse sublimite.

Template-linked exemplo de tamanho de política

Para descobrir como as políticas vinculadas a modelos contribuem para o tamanho da política por cota de recursos, adicione a extensão do principal e a extensão do recurso. Se você não especificar um recurso, as Permissões verificadas contabilizam seu tamanho no total do `"unspecified"` recurso.

Analise o seguinte modelo:

```
permit (
  principal in ?principal,
  action in
    [MyApplication::Action::"ContentManagement",
     MyApplication::Action::"AccountAdministration",
     MyApplication::Action::"BillingOperations"],
  resource in ?resource
```

```
)
when {
  resource has authorizedTenantIds &&
  principal has tenantId &&
  resource.authorizedTenantIds.contains(principal.tenantId)
};
```

Crie as seguintes políticas a partir desse modelo:

```
TemplateLinkedPolicy {
  policyId: "policy1",
  templateId: "template1",
  principal: User::"alice",
  resource: Photo::"car.jpg"
}

TemplateLinkedPolicy {
  policyId: "policy2",
  templateId: "template1",
  principal: User::"bob",
  resource: Photo::"boat.jpg"
}

TemplateLinkedPolicy {
  policyId: "policy3",
  templateId: "template1",
  principal: User::"jane",
  resource: Photo::"car.jpg"
}

TemplateLinkedPolicy {
  policyId: "policy4",
  templateId: "template1",
  principal: User::"jane",
  resource
}
```

Calcule o tamanho dessas políticas contando os bytes em `principal` e `resource` para cada uma. As permissões verificadas medem o tamanho em UTF-8 bytes. Os caracteres no intervalo ASCII imprimível contam como 1 byte cada, mas outros caracteres podem usar mais. Os exemplos a seguir usam somente caracteres ASCII, então cada caractere conta como 1 byte. Esses cálculos excluem a sobrecarga fixa. Seus totais reais serão maiores.

O tamanho de `policy1` é o comprimento do principal `User::"alice"` (13) mais o comprimento do recurso `Photo::"car.jpg"` (16). O total é 13 + 16 = 29 bytes.

O tamanho de `policy2` é o comprimento do principal `User::"bob"` (11) mais o comprimento do recurso `Photo::"boat.jpg"` (17). O total é 11 + 17 = 28 bytes.

O tamanho de `policy3` é o comprimento do principal `User::"jane"` (12) mais o comprimento do recurso `Photo::"car.jpg"` (16). O total é 12 + 16 = 28 bytes.

O tamanho de `policy4` é o comprimento do principal `User::"jane"` (12) mais o comprimento do recurso (0). O total é 12 + 0 = 12 bytes.

Como `policy2` é a única política que faz referência ao recurso`Photo::"boat.jpg"`, o tamanho total do recurso é de 28 bytes.

Como `policy1` `policy3` ambos fazem referência ao recurso`Photo::"car.jpg"`, o tamanho total do recurso é 29 + 28 = 57 bytes.

Como `policy4` é a única política que faz referência ao `"unspecified"` recurso, o tamanho total do recurso é de 12 bytes.

Exemplos de políticas do Amazon Verified Permissions

Alguns dos exemplos de políticas incluídos aqui são exemplos básicos de políticas da Cedar e alguns são verificados Permissions-specific. Os básicos estão vinculados ao Guia de referência da linguagem política do Cedar e estão incluídos nele. 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 acesso a entidades individuais
- Permite acesso a grupos de entidades
- Permite o acesso de qualquer entidade
- Permite acesso aos atributos de uma entidade (ABAC)
- Nega acesso
- Usa a notação de colchetes para referenciar atributos de token
- Usa notação de pontos para referenciar atributos

- Reflete Amazon Cognito Atributos do token de ID
- Reflete os atributos do token de ID OIDC
- Reflete Amazon Cognito atributos do token de acesso
- Reflete os atributos do token de acesso OIDC

Usa a notação de colchetes para referenciar atributos de token

O exemplo a seguir mostra como você pode criar uma política que usa a notação de colchetes para referenciar atributos de token.

Para obter mais informações sobre o uso de atributos de token em políticas em Permissões verificadas, consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens do OIDC para o esquema.

```
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.*"
};
```

Usa notação de pontos para referenciar atributos

O exemplo a seguir mostra como você pode criar uma política que usa notação de pontos para referenciar atributos.

Para obter mais informações sobre o uso de atributos de token em políticas em Permissões verificadas, consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens do OIDC para o esquema.

```
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"
};
```

Reflete Amazon Cognito Atributos do token de ID

O exemplo a seguir mostra como você pode criar uma política que faça referência aos atributos do token de ID Amazon Cognito.

Para obter mais informações sobre o uso de atributos de token em políticas em Permissões verificadas, consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens do OIDC para o esquema.

```
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"
};
```

Reflete os atributos do token de ID OIDC

O exemplo a seguir mostra como você pode criar uma política que faça referência aos atributos do token de ID de um provedor OIDC.

Para obter mais informações sobre o uso de atributos de token em políticas em Permissões verificadas, consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens do OIDC para o esquema.

```
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*"
};
```

Reflete Amazon Cognito atributos do token de acesso

O exemplo a seguir mostra como você pode criar uma política que faça referência aos atributos do token de acesso a partir de Amazon Cognito.

Para obter mais informações sobre o uso de atributos de token em políticas em Permissões verificadas, consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens do OIDC para o esquema.

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

Reflete os atributos do token de acesso OIDC

O exemplo a seguir mostra como você pode criar uma política que faça referência aos atributos do token de acesso de um provedor OIDC.

Para obter mais informações sobre o uso de atributos de token em políticas em Permissões verificadas, consulte Mapeamento de Amazon Cognito tokens para o esquema e Mapeamento de tokens do OIDC para o esquema.

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

Modelos de política de permissões verificadas da Amazon e políticas vinculadas a modelos

Nas Permissões verificadas, os modelos de política são políticas com espaços reservados para `principal`ou `resource`, ou para ambos. Os modelos de política por si só não podem ser usados para lidar com solicitações de autorização. Para lidar com solicitações de autorização, uma política vinculada ao modelo deve ser criada com base em um modelo de política. Os modelos de política permitem que uma política seja definida uma vez e depois usada com vários diretores e recursos. As atualizações do modelo de política são refletidas em todas as políticas que usam o modelo. Para obter mais informações, consulte Modelos de políticas do Cedar no Guia de referência da linguagem de política Cedar.

Opcionalmente, você pode atribuir um nome de modelo de política a um modelo de política. Os nomes dos modelos de política devem ser exclusivos no repositório de políticas e prefixados com`name/`. Você pode usar um nome de modelo de política no lugar do ID do modelo de política em operações do plano de controle que aceitam um `policyTemplateId` parâmetro. Somente `GetPolicyTemplate` e `ListPolicyTemplates` retorne o nome na saída. O exemplo a seguir usa um nome de modelo de política para recuperar um modelo de política com`GetPolicyTemplate`.

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

Por exemplo, o modelo de política a seguir forneceRead,`Edit`, e `Comment` permissões para o principal e o recurso que usam o modelo de política.

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

Se você criasse uma política nomeada `Editor` com base nesse modelo, quando um diretor fosse designado como editor de um recurso específico, seu aplicativo criaria uma política que fornecesse permissões para o diretor ler, editar e comentar sobre o recurso.

Diferentemente das políticas estáticas, as políticas vinculadas a modelos são dinâmicas. Veja o exemplo anterior: se você removesse a `Comment` ação do modelo de política, qualquer política

vinculada ou baseada nesse modelo seria atualizada adequadamente e os diretores especificados nas políticas não poderiam mais comentar sobre os recursos correspondentes.

Para obter mais exemplos de políticas vinculadas a modelos, consulte. Exemplo de políticas vinculadas a modelos de permissões verificadas da Amazon

Criação de modelos de política de permissões verificadas da Amazon

Você pode criar modelos de política em Permissões verificadas usando o Console de gerenciamento da AWS AWS CLI, o ou AWS SDKs o. Os modelos de política permitem que uma política seja definida uma vez e depois usada com vários diretores e recursos. Depois de criar um modelo de política, você pode criar políticas vinculadas a modelos para usar os modelos de política com princípios e recursos específicos. Para obter mais informações, consulte Criação de políticas vinculadas a modelos de permissões verificadas da Amazon.

Console de gerenciamento da AWS

Como criar um modelo de política

1. Abra o console de Permissões verificadas. 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"
}
```

Para criar um modelo de política com um nome de modelo de política

Opcionalmente, você pode especificar um nome de modelo de política ao criar um modelo de política. O nome deve ser exclusivo para todos os modelos de política no repositório de políticas e prefixado com`name/`. Você pode usar o nome no lugar do ID do modelo de política.

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

> **Note**
>
> Se você especificar um nome que já esteja associado a outro modelo de política no repositório de políticas, receberá um `ConflictException` erro.

Criação de políticas vinculadas a modelos de permissões verificadas da Amazon

Você pode criar políticas vinculadas a modelos ou políticas baseadas em um modelo de política usando o Console de gerenciamento da AWS AWS CLI, ou o. AWS SDKs 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.

Para exemplos de políticas vinculadas a modelos, consulte. Exemplo de políticas vinculadas a modelos de permissões verificadas da Amazon

Console de gerenciamento da AWS

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

1. Abra o console de Permissões verificadas. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha 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 == PhotoFlash::Action::"view",
    resource == PhotoFlash::Photo::"VacationPhoto94.jpg"
);
```

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

```
{
    "templateLinked": {
        "policyTemplateId": "PTEXAMPLEabcdefg111111",
        "principal": {
            "entityType": "PhotoFlash::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": "PhotoFlash::User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "PhotoFlash::Photo"
    }
}
```

Editando modelos de política de permissões verificadas da Amazon

Você pode editar ou atualizar modelos de política em Permissões verificadas usando o Console de gerenciamento da AWS AWS CLI, o ou AWS SDKs o. A edição de um modelo de política atualizará automaticamente as políticas vinculadas ou baseadas no modelo, portanto, tome cuidado ao editar os modelos de política e certifique-se de não introduzir acidentalmente uma alteração que interrompa seu aplicativo.

Você pode alterar os seguintes elementos de um modelo de política:

- O `action` referenciado pelo modelo de política
- Uma cláusula de condição, como e `when` `unless`

Você não pode alterar os seguintes elementos de um modelo de política. Para alterar qualquer um desses elementos, você precisará excluir e recriar o modelo de política.

- O efeito de um modelo de política de `permit` ou `forbid`
- O `principal` referenciado por um modelo de política
- O `resource` referenciado por um modelo de política

Console de gerenciamento da AWS

Para editar os modelos de política

1. Abra o console de Permissões verificadas. 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 editar 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"
}
```

Para atualizar o nome de um modelo de política

Você pode definir ou atualizar o nome de um modelo de política ao atualizar um modelo de política. O nome deve ser exclusivo para todos os modelos de política no repositório de políticas e prefixado com`name/`. Se você não incluir o campo de nome na solicitação de atualização, o nome existente permanecerá inalterado. Para remover um nome, defina-o como uma string vazia.

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

Exemplo de políticas vinculadas a modelos de permissões verificadas da Amazon

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.

PhotoFlash exemplos

O exemplo a seguir 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"
 );
```

O exemplo a seguir 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"
 );
```

O exemplo a seguir 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"
 );
```

O exemplo a seguir 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"
 );
```

O exemplo a seguir mostra como você pode criar uma política vinculada a um modelo que usa o modelo de política. Conceda 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"
 );
```

O exemplo a seguir 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 exemplos

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

O exemplo a seguir mostra como você pode criar uma política vinculada ao modelo que usa o modelo de política que dá ao espectador acesso a uma lista individual de usuários e 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"
);
```

O exemplo a seguir mostra como você pode criar uma política vinculada ao modelo que usa o modelo de política que dá acesso ao editor para uma lista individual de usuários e 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"
);
```

Proteja seus aplicativos com fontes de identidade e tokens

Proteja seus aplicativos rapidamente criando uma fonte de identidade para representar 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. Você pode adicionar um grupo de usuários do Amazon Cognito ou um IdP personalizado do OpenID Connect (OIDC) como sua fonte de identidade.

Você pode usar os provedores de identidade () do OpenID Connect (OIDC) com permissões IdPs verificadas. Seu aplicativo pode gerar solicitações de autorização com tokens web JSON (JWTs) gerados por um provedor de identidade compatível com OIDC. A identidade do usuário no token é mapeada para a ID principal. Com tokens de ID, as Permissões Verificadas mapeiam as declarações de atributos aos atributos principais. Com os tokens de acesso, essas declarações são mapeadas de acordo com o 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).

> **Note**
>
> As Permissões Verificadas tomam decisões de autorização com base nas informações de um token de IdP, mas não interagem diretamente com o IdP de forma alguma.

Para ver um step-by-step passo a passo que cria a lógica de autorização para o Amazon API Gateway REST APIs usando um grupo de Amazon Cognito usuários ou um provedor de identidade OIDC, consulte Autorize usando API Gateway APIs Amazon Verified Permissions com Amazon Cognito ou traga seu próprio provedor de identidade no Blog de segurança.AWS

Tópicos

- Escolhendo o provedor de identidade certo
- Trabalhando com fontes de Amazon Cognito identidade
- Trabalhando com fontes de identidade do OIDC

Escolhendo o provedor de identidade certo

Embora as Permissões Verificadas funcionem com uma variedade de IdPs, considere o seguinte ao decidir qual delas usar em seu aplicativo:

Use Amazon Cognito quando:

- Você está criando novos aplicativos sem a infraestrutura de identidade existente
- Você quer grupos AWS de usuários gerenciados com recursos de segurança integrados
- Você precisa de integração com provedores de identidade social
- Você quer um gerenciamento simplificado de tokens

Use os provedores do OIDC quando:

- Você tem uma infraestrutura de identidade existente (Auth0, Okta, Azure AD)
- Você precisa manter o gerenciamento centralizado de usuários
- Você tem requisitos de conformidade específicos IdPs

Trabalhando com fontes de Amazon Cognito identidade

As permissões verificadas trabalham em estreita colaboração com os grupos de usuários do Amazon Cognito. Amazon Cognito JWTs 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 principal que você deseja associar à sua fonte de identidade, por exemplo`MyCorp::User`.
- O tipo de entidade do grupo principal que você deseja associar à sua fonte de identidade, por exemplo`MyCorp::UserGroup`.
- O cliente IDs 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` Nesse caso, você referenciaria um usuário nesse grupo de usuários com ID `a1b2c3d4-5678-90ab-cdef-EXAMPLE22222` como `us-west-2_EXAMPLE|a1b2c3d4-5678-90ab-cdef-EXAMPLE22222`

As declarações de token do grupo de usuários podem conter atributos, escopos, grupos IDs, clientes e dados personalizados. Amazon Cognito JWTstêm 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, emMapeamento de Amazon Cognito tokens para o esquema.

> **Important**
>
> Embora você possa revogar Amazon Cognito os tokens antes que eles expirem, eles JWTs são considerados recursos apátridas que são 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 é possível que as Permissões Verificadas concedam acesso com base em um token que foi revogado ou emitido para um usuário que foi 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. Para obter mais informações, consulte Encerramento de sessões de usuário com revogação de token

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

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

O exemplo a seguir mostra como você pode criar uma política que faça referência a um principal que é um usuário em um grupo de usuários do Cognito. Observe que o ID principal assume a forma de`"<userpool-id>|<sub>"`.

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

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`

Para obter mais informações sobre o mapeamento de declarações, consulteMapeamento de Amazon Cognito tokens para o esquema. Para obter mais informações sobre autorização para

Amazon Cognito usuários, consulte Autorização com permissões verificadas da Amazon no Guia do desenvolvedor do Amazon Cognito.

Tópicos

- Criação de fontes de Amazon Cognito identidade do Amazon Verified Permissions
- Editando fontes de Amazon Cognito identidade do Amazon Verified Permissions
- Mapeamento de Amazon Cognito tokens para o esquema
- Validação de clientes e públicos para Amazon Cognito

Criação de fontes de Amazon Cognito identidade do Amazon Verified Permissions

O procedimento a seguir adiciona uma fonte de identidade a um repositório de políticas existente.

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 opção Configuração guiada ou API Gateway Configurar com 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 create-identity-sourcena API de permissões verificadas AWS CLI ou CreateIdentitySourcena API de permissões verificadas. 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 Amazon Cognito identidades com seu esquema de armazenamento de políticas, você deve definir o tipo de entidade principal como uma entidade de armazenamento de políticas compatível.

Console de gerenciamento da AWS

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

1. Abra o console de Permissões verificadas. 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, em Tipo principal, escolha o tipo de entidade para principais dessa origem. 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 o aplicativo IDs cliente.
   - Para validar o aplicativo cliente IDs, escolha Aceitar somente tokens com o aplicativo IDs cliente correspondente. 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 validar o aplicativo cliente IDs se você não quiser validar o aplicativo cliente. IDs
8. Escolha Criar origem de identidade.
9. (Opcional) Se seu repositório de políticas tiver um esquema, antes de fazer referência aos atributos extraídos dos tokens de identidade ou acesso em suas políticas do Cedar, você deverá atualizar 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 de atributos de Amazon Cognito token para os atributos principais do Cedar, consulteMapeamento de Amazon Cognito tokens para o esquema.

> **Note**
>
> Quando você cria um repositório de políticas vinculado à API ou usa Configurar com API Gateway um provedor de identidade ao criar repositórios de políticas, 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.

10. Crie políticas que usem informações dos tokens para tomar decisões de autorização. Para obter mais informações, consulte Criação de políticas estáticas do Amazon Verified Permissions.

Agora que você criou uma fonte de identidade, atualizou o esquema e criou políticas, use `IsAuthorizedWithToken` para que as Permissões Verificadas tomem decisões de autorização. Para obter mais informações, consulte IsAuthorizedWithTokeno guia de referência da Amazon Verified Permissions API.

AWS CLI

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

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

1. Crie um `config.txt` arquivo que contenha os seguintes detalhes do grupo de Amazon Cognito usuários para uso pelo `--configuration` parâmetro no `create-identity-source` comando.

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

2. Execute o comando a seguir para criar uma fonte de Amazon Cognito identidade.

```
$ aws verifiedpermissions create-identity-source \
    --configuration file://config.txt \
    --principal-entity-type "User" \
```

```
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

3. (Opcional) Se seu repositório de políticas tiver um esquema, antes de fazer referência aos atributos extraídos dos tokens de identidade ou acesso em suas políticas do Cedar, você deverá atualizar 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 de atributos de Amazon Cognito token para os atributos principais do Cedar, consulteMapeamento de Amazon Cognito tokens para o esquema.

   > **Note**
   >
   > Quando você cria um repositório de políticas vinculado à API ou usa Configurar com API Gateway um provedor de identidade ao criar repositórios de políticas, 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.

4. Crie políticas que usem informações dos tokens para tomar decisões de autorização. Para obter mais informações, consulte Criação de políticas estáticas do Amazon Verified Permissions.

Agora que você criou uma fonte de identidade, atualizou o esquema e criou políticas, use `IsAuthorizedWithToken` para que as Permissões Verificadas tomem decisões de autorização. Para obter mais informações, consulte IsAuthorizedWithTokeno guia de referência da Amazon Verified Permissions API.

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.

Editando fontes de Amazon Cognito identidade do Amazon Verified Permissions

Você pode editar alguns parâmetros da sua fonte de identidade depois de criá-la. Você não pode alterar o tipo de fonte de identidade, você precisa excluir a fonte de identidade e criar uma nova para mudar para o OIDC ou Amazon Cognito para o OIDC. Amazon Cognito 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.

Console de gerenciamento da AWS

Para atualizar uma fonte Amazon Cognito de identidade

1. Abra o console de Permissões verificadas. 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. Escolha 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 grupos do 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 o aplicativo IDs cliente.
   - Para validar o aplicativo cliente IDs, escolha Aceitar somente tokens com o aplicativo IDs cliente correspondente. 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 validar o aplicativo cliente IDs se você não quiser validar o aplicativo cliente. IDs
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 fonte Amazon Cognito de identidade

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

1. Crie um `config.txt` arquivo que contenha os seguintes detalhes do grupo de Amazon Cognito usuários para uso pelo `--configuration` parâmetro no `update-identity-source` comando.

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

2. Execute o comando a seguir para atualizar uma fonte de Amazon Cognito identidade.

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

> **Note**
>
> Se 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.

Mapeamento de Amazon Cognito tokens para o esquema

Talvez você queira adicionar uma fonte de identidade a um repositório de políticas e mapear declarações, ou tokens, do provedor ao esquema do repositório de políticas. Você pode automatizar esse processo usando a Configuração guiada para criar seu repositório de políticas com uma fonte de identidade ou atualizar seu esquema manualmente após a criação do repositório de políticas. Depois de mapear os tokens para o esquema, você pode criar políticas que façam referência a eles.

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 declarações de Amazon Cognito token 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 que foram criados 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 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 Amazon Cognito os tokens para seu esquema nas seguintes condições:

- Você criou um repositório de políticas vazio 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 Amazon Cognito como fonte de identidade em seu repositório de políticas de Permissões Verificadas, você deve ter atributos de provedor em seu esquema. O esquema é fixo e deve corresponder às entidades que os tokens do provedor criam IsAuthorizedWithTokenou às solicitações de BatchIsAuthorizedWithTokenAPI. 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 de provedor ao esquema que correspondam às entidades criadas usando solicitações de 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

- Mapeamento de tokens de ID para o esquema
- Mapeamento de tokens de acesso
- Notação alternativa para declarações delimitadas por Amazon Cognito dois pontos
- Coisas que você deve saber sobre mapeamento de esquemas

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.

Amazon Cognito Os tokens de ID funcionam com a maioria das bibliotecas independentes 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 Amazon Cognito desenvolvedor.

Reivindicações úteis em tokens Amazon Cognito de ID

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 a Amazon Cognito atributos 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 Amazon Cognito desenvolvedor.

O exemplo de token de ID a seguir tem cada um dos quatro tipos de atributos. Ela inclui a reivindicação Amazon Cognito específica`cognito:username`, a declaração

personalizadacustom:employmentStoreCode, a reivindicação padrão e a reivindicação email transitória. 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 Amazon Cognito usuários, você especifica o tipo de entidade principal com a qual o Verified Permissions gera nas solicitações de autorizaçãoIsAuthorizedWithToken. 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 Editando

esquemas de armazenamento de políticas. 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
         }
      }
   }
}
```

Para obter um exemplo de política que será validada em relação a esse esquema, consulte. Reflete Amazon Cognito Atributos do token de ID

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.

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

Reivindicações úteis em tokens de Amazon Cognito acesso

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 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 Amazon Cognito desenvolvedor.

Um token de Amazon Cognito acesso 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 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 Editando esquemas de armazenamento de políticas.

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

Para obter um exemplo de política que será validada em relação a esse esquema, consulte. Reflete Amazon Cognito atributos do token de acesso

Notação alternativa para declarações delimitadas por Amazon Cognito dois pontos

No momento em que as Permissões verificadas foram lançadas, o esquema recomendado para declarações de Amazon Cognito token, como `cognito:groups` e `custom:store` converteu essas cadeias de caracteres delimitadas por dois pontos para usar o caractere como um 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. 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
         }
      }
   }
}
```

Para obter um exemplo de política que validará esse esquema e usará a notação de pontos, consulte. Usa notação de pontos para referenciar atributos

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 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 um provedor 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 obter mais informações, consulte Mapeamento de tokens de acesso.

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 Amazon Cognito identidade, você pode escolher o tipo de token ao criar um repositório 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. API Gateway 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:username` 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"
      ]
   }
}
```

Para obter um exemplo de política que será validada em relação a esse esquema, consulte. Usa a notação de colchetes para referenciar atributos de token

Validação de clientes e públicos para Amazon Cognito

Quando você adiciona uma fonte de identidade a um repositório de políticas, as Permissões Verificadas têm opções de configuração que validam a ID do cliente e o público dos tokens de ID e acesso, além de verificar a `token_use` declaração. Essa validação acontece no processamento `IsAuthorizedWithToken` e nas solicitações de `BatchIsAuthorizedWithToken` API. O comportamento difere entre tokens de ID e acesso Amazon Cognito e entre fontes de identidade 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 à instância do provedor de identidade que seu aplicativo usa, por exemplo`1example23456789`. Um público é um caminho de URL associado à parte confiável ou ao destino pretendido do token de acesso, por exemplo`https://mytoken.example.com`. Ao usar tokens de acesso, a `aud` reivindicação está sempre associada ao público.

Amazon Cognito Os tokens de ID 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 do cliente em sua fonte de identidade, as Permissões verificadas comparam 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. Amazon Cognito

Client-side autorização 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 aplicativo usando um JWT.[1]

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

    let principalEntity = {
        entityType: "PhotoFlash::User", // the application needs to fill in the
 relevant user type
        entityId: payload["sub"], // the application need to use the claim that
 represents the user-id
    };
    let resourceEntity = {
```

```
        entityType: "PhotoFlash::Photo", //the application needs to fill in the
 relevant resource type
        entityId: "jane_photo_123.jpg", // the application needs to fill in the
 relevant resource id
    };
    let action = {
        actionType: "PhotoFlash::Action", //the application needs to fill in the
 relevant action id
        actionId: "GetPhoto", //the application needs to fill in the relevant action
 type
    };
    let entities = {
        entityList: [],
    };
    entities.entityList.push(...getUserEntitiesFromToken(payload));
    let policyStoreId = "PSEXAMPLEabcdefg111111"; // set your own policy store id

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

    return authResult;

}

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

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

  });

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

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

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 JWTs produzem 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.
- Registros CNAME que não incluem curingas. Por exemplo, não `a.example.com` pode ser mapeado para`*.example.net`. Por outro lado, não `*.example.com` pode ser mapeado para. `a.example.net`
- 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 à entidade IDs de usuário e grupo. Na CLI e na API, você pode escolher esse prefixo. Nos repositórios de políticas que você cria com a opção Configurar com o API Gateway e um provedor de identidade ou 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"`

Para obter mais informações sobre o uso de operações de API para autorizar solicitações de fontes do OIDC, consulte. Operações de API disponíveis para autorização

O exemplo a seguir 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.

Observe que, ao fazer referência a um grupo no principal, você deve usar o `in` operador para que a política seja avaliada corretamente.

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

Tópicos

- Criação de fontes de identidade OIDC do Amazon Verified Permissions
- Editando fontes de identidade do Amazon Verified Permissions OIDC
- Mapeando tokens OIDC para o esquema
- Validação de clientes e públicos para fornecedores de OIDC

Criação de fontes de identidade OIDC do Amazon Verified Permissions

O procedimento a seguir adiciona uma fonte de identidade a um repositório de políticas existente.

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 opção Configuração guiada ou API Gateway Configurar com 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 create-identity-sourcena API de permissões verificadas AWS CLI ou CreateIdentitySourcena API de permissões verificadas. 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 Amazon Cognito identidades com seu esquema de armazenamento de políticas, você deve definir o tipo de entidade principal como uma entidade de armazenamento de políticas compatível.

Console de gerenciamento da AWS

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

1. Abra o console de Permissões verificadas. Escolha seu repositório de políticas.
2. No painel de navegação à esquerda, escolha Origens de identidade.

3. Escolha Criar origem de identidade.
4. Escolha provedor OIDC externo.
5. Em URL do emissor, insira a URL do 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 obter mais informações, consulte Mapeando tokens OIDC para o esquema.
7. Em Mapear reivindicações de token para entidades do esquema, escolha uma entidade de usuário e uma declaração 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. (Opcional) Em Mapear declarações de token para entidades do esquema, escolha uma entidade de grupo e uma declaração de grupo para a fonte de identidade. A entidade do Grupo é mãe 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 - opcional, insira o cliente IDs ou público URLs que você deseja que seu repositório de políticas aceite nas solicitações de autorização, se houver.
10. Escolha Criar origem de identidade.
11. (Opcional) Se seu repositório de políticas tiver um esquema, antes de fazer referência aos atributos que você extrai dos tokens de identidade ou acesso em suas políticas do Cedar, você deverá atualizar 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 do token OIDC para os atributos principais do Cedar, consulte. Mapeando tokens OIDC para o esquema
12. Crie políticas que usem informações dos tokens para tomar decisões de autorização. Para obter mais informações, consulte Criação de políticas estáticas do Amazon Verified Permissions.

Agora que você criou uma fonte de identidade, atualizou o esquema e criou políticas, use `IsAuthorizedWithToken` para que as Permissões Verificadas tomem decisões de autorização. Para obter mais informações, consulte IsAuthorizedWithTokeno guia de referência da Amazon Verified Permissions API.

AWS CLI

Para criar uma fonte de identidade OIDC

Você pode criar uma fonte de identidade usando a CreateIdentitySourceoperação. O exemplo a seguir cria uma fonte de identidade que pode acessar identidades autenticadas de um provedor de identidade OIDC (IdP).

1. Crie um `config.txt` arquivo que contenha os seguintes detalhes de um IdP do OIDC para uso pelo `--configuration` parâmetro do comando. `create-identity-source`

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

2. Execute o comando a seguir para criar uma fonte de identidade OIDC.

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

```
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

3. (Opcional) Se seu repositório de políticas tiver um esquema, antes de fazer referência aos atributos que você extrai dos tokens de identidade ou acesso em suas políticas do Cedar, você deverá atualizar 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 do token OIDC para os atributos principais do Cedar, consulte. Mapeando tokens OIDC para o esquema
4. Crie políticas que usem informações dos tokens para tomar decisões de autorização. Para obter mais informações, consulte Criação de políticas estáticas do Amazon Verified Permissions.

Agora que você criou uma fonte de identidade, atualizou o esquema e criou políticas, use `IsAuthorizedWithToken` para que as Permissões Verificadas tomem decisões de autorização. Para obter mais informações, consulte IsAuthorizedWithTokeno guia de referência da Amazon Verified Permissions API.

Editando fontes de identidade do Amazon Verified Permissions OIDC

Você pode editar alguns parâmetros da sua fonte de identidade depois de criá-la. Você não pode alterar o tipo de fonte de identidade, você precisa excluir a fonte de identidade e criar uma nova para mudar para o OIDC ou Amazon Cognito para o OIDC. Amazon Cognito 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.

Console de gerenciamento da AWS

Para atualizar uma fonte de identidade do OIDC

1. Abra o console de Permissões verificadas. 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. Escolha 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 UpdateIdentitySourceoperação. O exemplo a seguir atualiza a fonte de identidade especificada para usar um provedor OIDC diferente.

1. Crie um `config.txt` arquivo que contenha os seguintes detalhes de um IdP do OIDC para uso pelo `--configuration` parâmetro do comando. `update-identity-source`

```
{
    "openIdConnectConfiguration": {
        "issuer": "https://auth2.example.com",
        "tokenSelection": {
                "identityTokenOnly": {
                        "clientIds":["2example10111213"],
                        "principalIdClaim": "sub"
                },
        },
        "entityIdPrefix": "MyOIDCProvider",
        "groupConfiguration": {
              "groupClaim": "groups",
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

2. Execute o comando a seguir para atualizar uma fonte de identidade do OIDC.

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

> **Note**
>
> Se 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.

Mapeando tokens OIDC para o esquema

Talvez você queira adicionar uma fonte de identidade a um repositório de políticas e mapear declarações, ou tokens, do provedor ao esquema do repositório de políticas. Você pode automatizar esse processo usando a Configuração guiada para criar seu repositório de políticas com uma fonte de identidade ou atualizar seu esquema manualmente após a criação do repositório de políticas. Depois de mapear os tokens para o esquema, você pode criar políticas que façam referência a eles.

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 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 que foram criados 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 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.

Para usar 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. O esquema é fixo e deve corresponder às entidades que os tokens do provedor criam IsAuthorizedWithTokenou às solicitações de BatchIsAuthorizedWithTokenAPI. 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 de provedor ao esquema que correspondam às entidades criadas usando solicitações de API. Em seguida, você pode escrever políticas usando atributos do token do provedor.

Tópicos

- Mapeamento de tokens de ID para o esquema
- Mapeamento de tokens de acesso
- Coisas que você deve saber sobre mapeamento de esquemas

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.

Trabalhar com tokens de ID de um provedor de OIDC é quase o mesmo que trabalhar com tokens de Amazon Cognito ID. 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 obter 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"
         }
      }
   }
}
```

Para obter um exemplo de política que será validada em relação a esse esquema, consulte. Reflete os atributos do token de ID OIDC

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.

A maioria dos tokens de acesso de provedores externos do OIDC se alinha estreitamente aos Amazon Cognito tokens de acesso. 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 Editando esquemas de armazenamento de políticas.

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

Para obter um exemplo de política que será validada em relação a esse esquema, consulte. Reflete os atributos do token de acesso OIDC

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 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 um provedor 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 obter 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 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:username` 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"
      ]
   }
}
```

Para obter um exemplo de política que será validada em relação a esse esquema, consulte. Usa a notação de colchetes para referenciar atributos de token

Validação de clientes e públicos para fornecedores de OIDC

Quando você adiciona uma fonte de identidade a um repositório de políticas, as Permissões Verificadas têm opções de configuração que validam a ID do cliente e o público dos tokens de ID e acesso. Essa validação acontece no processamento `IsAuthorizedWithToken` e nas solicitações de `BatchIsAuthorizedWithToken` API. O comportamento difere entre tokens de ID e acesso

Amazon Cognito e entre fontes de identidade 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.

Ao contrário dos Amazon Cognito tokens, os tokens OIDC não têm uma declaração padrão que indique se um token é um token de ID ou um token de acesso. Como resultado, as Permissões verificadas não verificam o tipo de token das fontes de identidade do OIDC. Você é responsável por transmitir o tipo de token que seu repositório de políticas está configurado para processar em cada `IsAuthorizedWithToken` `BatchIsAuthorizedWithToken` solicitação.

Um ID de cliente é um identificador associado à instância do provedor de identidade que seu aplicativo usa, por exemplo`1example23456789`. Um público é um caminho de URL associado à parte confiável ou ao destino pretendido do token de acesso, por exemplo`https://mytoken.example.com`. Ao usar tokens de acesso, a `aud` reivindicação está sempre associada ao público.

Os tokens de ID do OIDC têm uma `aud` declaração que contém IDs de clientes, como. `1example23456789`

Os tokens de acesso do OIDC têm uma `aud` declaração que contém a URL do público do token, como`https://myapplication.example.com`, e uma `client_id` declaração que contém IDs de clientes, como. `1example23456789`

Ao configurar seu repositório de políticas, insira um ou mais valores para validação de público que seu armazenamento de políticas usa para validar o público de um token.

- Tokens de ID — As permissões verificadas validam o ID do cliente verificando se pelo menos um membro dos IDs do cliente na `aud` declaração corresponde a um valor de validação de público.
- Tokens de acesso — as permissões verificadas validam o público verificando se o URL na `aud` declaração corresponde a um valor de validação do público. Se nenhuma `aud` reivindicação existir, o público poderá ser validado usando as `client_id` reivindicações `cid` ou. Verifique com seu provedor de identidade a afirmação e o formato corretos do público.

Client-side autorização 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 os atributos da sua entidade de um JSON Web Token (JWT) e analisá-los em Permissões verificadas.

Este exemplo mostra como você pode chamar Permissões Verificadas de um aplicativo usando um JWT.[1]

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

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

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

```
    return authResult;

}

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

  });

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

```
    }
  };
  return [entityItem];
}
```

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

Integrações para permissões verificadas pela Amazon

As integrações do Amazon Verified Permissions ajudam você a implementar autorizações refinadas em seus aplicativos, minimizando o código e seguindo as melhores práticas específicas da estrutura. Essas integrações fornecem componentes e utilitários de middleware que conectam perfeitamente seu aplicativo às permissões verificadas.

Com as integrações, você pode:

- Implemente a autorização em minutos
- Siga padrões e convenções específicos da estrutura
- Reduza a sobrecarga de manutenção
- Minimize possíveis erros de implementação de segurança
- Concentre-se na lógica de negócios em vez do código de autorização

Quando adicionadas ao seu aplicativo, as integrações fazem o seguinte:

1. Intercepte solicitações recebidas por meio de middleware específico da estrutura
2. Extraia o contexto de autorização relevante das solicitações
3. Determine as decisões de autorização usando permissões verificadas
4. Aplique o controle de acesso com base nos resultados da autorização

Atualmente, as permissões verificadas oferecem suporte às seguintes estruturas:

- Express.js para aplicativos Node.js

Integrando o Express com as permissões verificadas da Amazon

A integração do Verified Permissions Express fornece uma abordagem baseada em middleware para implementar a autorização em seus aplicativos Express.js. Com essa integração, você pode proteger seus endpoints de API usando políticas de autorização refinadas sem modificar seus manipuladores de rotas existentes. A integração processa as verificações de autorização automaticamente interceptando solicitações, avaliando-as de acordo com suas políticas definidas e garantindo que somente usuários autorizados possam acessar recursos protegidos.

Este tópico explica como configurar a integração do Express, desde a criação de um repositório de políticas até a implementação e o teste do middleware de autorização. Seguindo essas etapas, você pode adicionar controles de autorização robustos ao seu aplicativo Express com o mínimo de alterações no código.

Os seguintes GitHub repositórios são referenciados em todo este tópico:

- cedar-policy/ authorization-for-expressjs - O middleware de autorização Cedar para Express.js
- verifiedpermissions/ authorization-clients-js - Os clientes de autorização de permissões verificadas para JavaScript
- verifiedpermissions/examples/express-petstore - Exemplo de implementação usando o middleware Express.js

Pré-requisitos

Antes de implementar a integração Express, certifique-se de ter:

- Uma AWS conta com acesso a permissões verificadas
- Node.js e npm instalados
- Um aplicativo Express.js
- Um provedor de identidade OpenID Connect (OIDC) (como) Amazon Cognito
- AWS CLIconfigurado com as permissões apropriadas

Configurando a integração

Etapa 1: criar um repositório de políticas

Crie um repositório de políticas usando AWS CLI:

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

> **Note**
>
> Salve o ID do repositório de políticas retornado na resposta para uso nas etapas subsequentes.

Etapa 2: instalar dependências

Instale os pacotes necessários em seu aplicativo Express:

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

Configurar autorização

Etapa 1: gerar e carregar o esquema Cedar

Um esquema define o modelo de autorização para um aplicativo, incluindo os tipos de entidades no aplicativo e as ações que os usuários podem realizar. Recomendamos definir um namespace para seu esquema. Neste exemplo, usamos `YourNamespace`. Você anexa seu esquema aos seus repositórios de políticas de Permissões Verificadas e, quando as políticas são adicionadas ou modificadas, o serviço valida automaticamente as políticas em relação ao esquema.

O `@cedar-policy/authorization-for-expressjs` pacote pode analisar as especificações da OpenAPI do seu aplicativo e gerar um esquema Cedar. Especificamente, o objeto paths é obrigatório em sua especificação.

Se você não tiver uma especificação OpenAPI, siga as instruções rápidas do express-openapi-generatorpacote para gerar uma especificação OpenAPI.

Gere um esquema a partir da sua especificação da OpenAPI:

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

Em seguida, formate o esquema Cedar para uso com o. AWS CLI Para obter mais informações sobre o formato específico necessário, consulteEsquema do armazenamento de políticas.
Se precisar de ajuda para formatar o esquema, há um script chamado `prepare-cedar-schema.sh` no repositório GitHubverifiedpermissions/examples. Veja a seguir um exemplo de chamada para esse script que gera o esquema formatado de Permissões Verificadas no arquivo. `v2.cedarschema.forAVP.json`

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

Faça upload do esquema formatado em seu repositório de políticas, `policy-store-id` substituindo-o pelo ID do repositório de políticas:

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

Etapa 2: criar políticas de autorização

Se nenhuma política for configurada, o Cedar negará todas as solicitações de autorização. A integração da estrutura Express ajuda a iniciar esse processo gerando políticas de exemplo com base no esquema gerado anteriormente.

Ao usar essa integração em seus aplicativos de produção, recomendamos criar novas políticas usando ferramentas de infraestrutura como código (IAAC). Para obter mais informações, consulte Trabalhando com AWS CloudFormation.

Gere exemplos de políticas do Cedar:

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

Isso gerará exemplos de políticas no `/policies` diretório. Em seguida, você pode personalizar essas políticas com base em seus casos de uso. Por exemplo:

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

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

```
);
```

Formate as políticas para uso com AWS CLI o. Para obter mais informações sobre o formato exigido, consulte create-policy na referência.AWS CLI Se precisar de ajuda para formatar as políticas, há um script chamado `convert_cedar_policies.sh` no repositório GitHubverifiedpermissions/examples. A seguir está uma chamada para esse script:

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

Faça upload das políticas formatadas para Permissões verificadas, `policy_1.json` substituindo-as pelo caminho e nome do seu arquivo de política e `policy-store-id` pelo ID do repositório de políticas:

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

Etapa 3: Conectar um provedor de identidade

Por padrão, o middleware autorizador de permissões verificadas lê um token Web JSON (JWT) fornecido no cabeçalho de autorização da solicitação da API para obter informações do usuário. As permissões verificadas podem validar o token, além de realizar a avaliação da política de autorização.

Crie um arquivo de configuração de origem `identity-source-configuration.txt` de identidade chamado semelhante ao seguinte com seu `userPoolArn` `eclientId`:

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

Crie a fonte de identidade executando o seguinte AWS CLI comando, `policy-store-id` substituindo-o pelo ID do repositório de políticas:

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

Implementando o middleware de autorização

Atualize seu aplicativo Express para incluir o middleware de autorização. Neste exemplo, estamos usando tokens de identidade, mas você também pode usar tokens de acesso. Para obter mais informações, consulte authorization-for-expressjsemGitHub.

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

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

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

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

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

Testando a integração

Você pode testar sua implementação de autorização fazendo solicitações aos seus endpoints da API com diferentes tokens de usuário. O middleware de autorização avaliará automaticamente cada solicitação em relação às suas políticas definidas.

Por exemplo, se você configurou grupos de usuários diferentes com permissões diferentes:

- Administradores: acesso total a todos os recursos e funções de gerenciamento
- Funcionários: podem visualizar, criar e atualizar recursos
- Clientes: só podem visualizar recursos

Você pode validar se as políticas de permissões estão funcionando conforme o esperado fazendo login com usuários diferentes e tentando várias operações. No terminal do aplicativo Express, você pode ver a saída do log que fornece detalhes adicionais sobre as decisões de autorização.

Solução de problemas

Se você tiver falhas na autorização, tente o seguinte:

- Verifique se o ID do repositório de políticas está correto
- Certifique-se de que sua fonte de identidade esteja configurada corretamente
- Verifique se suas políticas estão formatadas corretamente
- Valide se seus tokens JWT são válidos

Próximas etapas

Depois de implementar a integração básica, considere:

- Implementando mapeadores personalizados para cenários de autorização específicos
- Configurando o monitoramento e o registro para decisões de autorização
- Criação de políticas adicionais para diferentes funções de usuário

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 de autorização 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 externas APIs. 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 repositório DigitalPetStorede políticas de amostra 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

Tópicos

- Operações de API disponíveis para autorização
- Testando seu modelo de autorização
- Integrando seus modelos de autorização com aplicativos

Operações de API disponíveis 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 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 a partir dos dados do usuário em tokens web JSON (JWTs). As permissões verificadas funcionam diretamente com os provedores do OIDC, 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 uma fonte de identidade.

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.

BatchIsAuthorized

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.

BatchIsAuthorizedWithToken

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 em um diretório de usuários.

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 da Amazon quando você implanta seu aplicativo, você pode avaliar suas políticas à medida que as desenvolve com Usando a bancada de testes de permissões verificadas da Amazon e com as solicitações da API REST HTTPS para Permissões Verificadas. A bancada de testes é uma ferramenta Console de gerenciamento da AWS 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 BatchIsAuthorizedcomo solicitações de AWS API 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 de Permissões verificadas no console de Permissões verificadas. Crie um repositório de políticas a partir do repositório de políticas de amostra com o nome DigitalPetStore.
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 do Exemplo 4 que fazem referência à DigitalPetStoreamostra.

    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 no 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 de Permissões verificadas no console de Permissões verificadas. Crie um repositório de políticas a partir do repositório de políticas de amostra com o nome DigitalPetStore.
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 DigitalPetStoreamostra.
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 repositório DigitalPetStorede políticas 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.

Integrando seus modelos de autorização com aplicativos

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 se originar de uma fonte de Amazon Cognito identidade 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 estão incorporadas AWS SDKs. Para incluir permissões verificadas em um aplicativo, integre o AWS SDK do idioma escolhido ao pacote do aplicativo.

Para saber mais e fazer o download AWS SDKs, consulte Ferramentas para Amazon Web Services.

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

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

O AWS SDK para 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

Como fazer solicitações de API

As solicitações de consulta para as Permissões Verificadas da Amazon são solicitações HTTP ou HTTPS que usam o `POST` método.

Endpoints de permissões verificadas

Um endpoint é uma URL que serve como ponto de entrada para um serviço web. Você pode selecionar um Região da AWS endpoint apropriado ao fazer suas solicitações para reduzir a latência. Para obter informações sobre os endpoints usados pelas Permissões verificadas, consulte Permissões verificadas da Amazon no Referência geral da Amazon Web Services.

Parâmetros de solicitação

O Amazon Verified Permissions usa um JSON-based protocolo. Você passa os parâmetros da solicitação como JSON no corpo da sua solicitação HTTP. As listas são representadas como matrizes JSON. Para obter mais informações, consulte Parâmetros comuns na referência da API de permissões verificadas da Amazon.

Identificadores de solicitação

Em cada resposta, AWS retorna um ID de solicitação por meio do cabeçalho de resposta `x-amzn-RequestId` HTTP. Essa string é um identificador exclusivo que é AWS atribuído para fornecer informações de rastreamento. Embora o ID da solicitação esteja incluído em todas as respostas, ele não está listado nas páginas individuais de documentação da API para melhorar a legibilidade e reduzir a redundância.

Autenticação da API de

Você envia solicitações de consulta por HTTPS. Você deve incluir uma assinatura em cada solicitação de consulta. Para obter mais informações sobre como criar e incluir uma assinatura, consulte Assinatura de solicitações de AWS API no Referência geral da Amazon Web Services.

Bibliotecas disponíveis

AWS fornece bibliotecas, exemplos de código, tutoriais e outros recursos para desenvolvedores de software que preferem criar aplicativos usando APIs específicas da linguagem em vez das

ferramentas de linha de comando e da API de consulta. Essas bibliotecas fornecem funções básicas (não incluídas nas APIs), como autenticação de solicitações, novas tentativas de solicitação e tratamento de erros, para facilitar o início. As bibliotecas e os recursos de permissões verificadas estão disponíveis para os seguintes idiomas e plataformas:

- AWS SDK para Go
- AWS SDK para Java 2.x
- AWS SDK para Java 1.x
- AWS SDK para JavaScript
- AWS SDK para .NET
- AWS SDK para PHP
- AWS SDK para Python (Boto)
- AWS SDK para Ruby

Para obter mais informações sobre bibliotecas e exemplos de código em todas as linguagens, consulte Código de amostra e bibliotecas.

Fazendo solicitações de API usando o método **POST**

Se você não usa um dos AWS SDKs, pode fazer solicitações de Permissões Verificadas por HTTPS usando o método de POST solicitação. O POST método exige que você especifique a operação no cabeçalho da solicitação e forneça os dados da operação no formato JSON no corpo da solicitação.

Nome do cabeçalho	Valor de cabeçalho
`Host`	O endpoint de permissões verificadas da Amazon. Por exemplo: `verifiedpermissions.us-east-1.amazonaws.com`
`X-Amz-Date`	Você deve fornecer o timestamp no cabeçalho HTTP Date ou no cabeçalho AWS x-amz-date. Algumas bibliotecas de cliente HTTP não permitem que você defina o cabeçalho de data. Quando um cabeçalho x-amz-date está presente, o sistema ignora qualquer cabeçalho de data durante a autenticação da solicitação. O cabeçalho x-amz-date deve ser especificado no formato básico ISO 8601. Por exemplo: `20130315T092054Z`

Nome do cabeçalho	Valor de cabeçalho
`Authorization`	O conjunto de parâmetros de autorização AWS usado para garantir a validade e a autenticidade da solicitação. Para obter mais informações sobre a construção desse cabeçalho, consulte Processo de assinatura do Signature versão 4 no Referência geral da Amazon Web Services.
`X-Amz-Target`	Especifica a operação de Permissões Verificadas que você deseja realizar. `VerifiedPermissions.` *`API_Name`* Por exemplo, para chamar a `CreatePolicy` operação, use o valor alvo a seguir. `VerifiedPermissions.CreatePolicy`
`Content-Type`	Especifica o formato de entrada. Use o valor a seguir. `application/x-amz-json-1.0`
`Accept`	Especifica o formato da resposta. Use o valor a seguir. `application/x-amz-json-1.0`
`Content-Length`	Tamanho da carga em bytes.
`Content-Encoding`	Especifica o formato de codificação da entrada e da saída. Use o valor a seguir. `amz-1.0`

Veja a seguir um exemplo de cabeçalho de uma solicitação HTTP para retornar uma lista de todas as políticas no repositório de políticas especificado no Conta da AWS qual as `Principal` referências são `User` nomeadas`alice`. Neste exemplo, a `Authorization` linha é dividida aqui para facilitar a leitura. Não inclua isso em sua solicitação real.

```
POST / HTTP/1.1
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230101T200059Z
```

```
Accept-Encoding: identity
Content-Type: application/x-amz-json-1.0
X-Amz-Target: VerifiedPermissions.ListPolicies
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
    "PolicyStoreId": "PSExAmPLE222222222",
    "Filter": {
        "Principal": {
            "Id": {
                "EntityType": "User",
                "EntityId": "alice"
            }
        }
    }
}
```

Segurança no Amazon Verified Permissions

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

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

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam às permissões verificadas da Amazon, consulte AWS Serviços no escopo do programa de conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da 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. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos de Permissões Verificadas.

Tópicos

- Proteção de dados no Amazon Verified Permissions
- Gerenciamento de identidades e acesso para Amazon Verified Permissions
- Validação de conformidade do Amazon Verified Permissions
- Resiliência no Amazon Verified Permissions

Proteção de dados no Amazon Verified Permissions

O modelo de responsabilidade AWS compartilhada se aplica à proteção de dados nas Permissões Verificadas da Amazon. 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 o conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança do Serviços da AWS que você usa. Para saber mais sobre a privacidade de dados, consulte as Data Privacy FAQ. Para saber mais sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

- Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM ou 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 AWS os recursos. O TLS 1.2 é obrigatório.
  - Configure a API e o registro de atividades do usuário com AWS CloudTrail.
  - Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
  - Use serviços de segurança gerenciados avançados Amazon Macie, como, que ajudam a descobrir e proteger dados confidenciais armazenados em. Amazon S3
  - Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-2.
- É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com permissões verificadas ou outras Serviços da AWS usando o console AWS CLI, a API ou AWS SDKs. Quaisquer dados inseridos em tags ou em 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, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse 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 um Chave gerenciada pela AWS. As permissões verificadas da Amazon também permitem que os clientes utilizem a chave gerenciada pelo cliente para criptografar seus dados.

Para obter informações detalhadas sobre o uso de chaves gerenciadas pelo cliente para criptografia, consultethe section called “Chaves gerenciadas pelo cliente”.

Criptografando recursos nas permissões verificadas da Amazon

O Amazon Verified Permissions fornece criptografia por padrão para proteger dados confidenciais de clientes em repouso usando chaves de criptografia AWS próprias. Como uma camada extra de proteção, o Amazon Verified Permissions permite que você criptografe seus repositórios de políticas usando AWS Key Management Service (AWS KMS) chave gerenciada pelo cliente s (CMK). Essa funcionalidade garante a proteção de dados confidenciais por meio de criptografia em repouso, o que ajuda você a:

- Reduza a carga operacional do lado do seu aplicativo para proteger dados confidenciais
- Mantenha o controle sobre quem pode ver os detalhes de suas políticas de autorização por meio de seus próprios AWS KMS chave gerenciada pelo cliente
- Criar aplicações com exigências de segurança que atendam a rigorosos requisitos regulatórios e de conformidade de criptografia.

As seções a seguir explicam como configurar a criptografia para novos repositórios de políticas e como gerenciar suas chaves de criptografia.

AWS KMS Tipos de chaves para permissões verificadas pela Amazon

O Amazon Verified Permissions se integra AWS KMS ao gerenciamento de chaves de criptografia usadas para dados de encrypting/decrypting clientes. Para saber mais sobre tipos e estados de chave, consulte AWS Key Management Service concepts no Guia do desenvolvedor do AWS KMS . Ao criar um novo repositório de políticas, você pode escolher entre os seguintes tipos de AWS KMS chave para criptografar seus dados:

AWS Chave própria

O tipo de criptografia padrão. O Amazon Verified Permissions possui a chave sem custo adicional para você e criptografa os dados do recurso em repouso no momento da criação. Nenhuma configuração adicional é necessária em seu código ou aplicativos para encrypt/decrypt seus dados usando a chave de propriedade das Permissões Verificadas.

Chave gerenciada pelo cliente

Você cria, possui e gerencia a chave na sua AWS conta. Você tem controle total sobre a AWS KMS chave. AWS KMS cobranças se aplicam chave gerenciada pelo cliente a s. Para obter mais informações, consulte a página de Definição de preço do AWS KMS. Para obter mais informações sobre os tipos de chaves, consulte Chaves gerenciadas pelo cliente no Guia do AWS KMS desenvolvedor.

Quando você especifica uma criptografia chave gerenciada pelo cliente para recursos de nível superior (ou seja, repositório de políticas), as Permissões Verificadas criptografam o recurso, bem como seus recursos secundários, com essa chave. Para criptografar um repositório de políticas usando um chave gerenciada pelo cliente, você precisa conceder acesso às Permissões Verificadas em sua política de chaves. Uma política chave é uma política baseada em recursos que você anexa à sua chave gerenciada pelo cliente para controlar o acesso a ela. Consulte the section called

"Autorizando o uso de seu AWS KMS chave para permissões verificadas pela Amazon" para obter mais detalhes.

Além disso, para criar um repositório de políticas criptografado com um chave gerenciada pelo cliente, ou para fazer chamadas de API para um armazenamento de políticas criptografado por um chave gerenciada pelo cliente, o usuário ou função do IAM que faz a chamada também deve ter acesso à chave. Se as Permissões Verificadas não conseguirem acessar a chave, qualquer decisão de autorização que envolva recursos criptografados por essa chave pode estar obsoleta ou imprecisa. Quando você não tiver acesso à chave, não conseguirá read/update /excluir recursos criptografados por essa chave, e qualquer chamada de criação para utilizar a chave para criptografia falhará.

> **Note**
>
> A criptografia de permissões verificadas em repouso está disponível em todas as AWS regiões em que as permissões verificadas estão disponíveis.

> **Important**
>
> Depois de usar a para criptografar um repositório de políticas, você NÃO PODE atualizar o recurso para usar uma chave diferente para criptografia ou remover a chave desse armazenamento de políticas. chave gerenciada pelo cliente

Utilizar AWS KMS e chaves de dados com Amazon Verified Permissions

O recurso de criptografia em repouso do Amazon Verified Permissions usa uma AWS KMS chave e uma hierarquia de chaves de dados para proteger seus dados de recursos.

> **Note**
>
> As permissões verificadas da Amazon são compatíveis somente com AWS KMS chaves simétricas. Você não pode usar uma AWS KMS chave assimétrica para criptografar seus recursos de Permissões Verificadas da Amazon.

Utilizar AWS Chaves próprias

O Amazon Verified Permissions criptografa todos os recursos por padrão com chaves AWS próprias. Essas chaves são de uso gratuito e são alternadas anualmente para proteger os recursos da sua conta. Você não precisa visualizar, gerenciar, usar ou auditar essas chaves; portanto, nenhuma ação é necessária proteger os dados. Para obter mais informações sobre chaves AWS próprias, consulte chaves AWS próprias no Guia do AWS KMS desenvolvedor.

Usando chaves gerenciadas pelo cliente

Selecionar um chave gerenciada pelo cliente para criptografia oferece os seguintes benefícios:

- Você cria e gerencia a AWS KMS chave, incluindo a definição das principais políticas e IAM políticas para controlar o acesso à AWS KMS chave. Você pode ativar e desativar a AWS KMS chave, ativar e desativar a rotação automática da chave e excluir a AWS KMS chave quando ela não estiver mais em uso.
- Você pode usar um chave gerenciada pelo cliente com material de chave importado ou um chave gerenciada pelo cliente em um armazenamento de chaves personalizado que você possui e gerencia.
- Você pode auditar a criptografia e a descriptografia de seus recursos de Permissões Verificadas examinando as chamadas AWS KMS da API de Permissões Verificadas da Amazon para os registros. AWS CloudTrail

Para que as Permissões Verificadas da Amazon usem seus chave gerenciada pelo cliente s encryption/decryption, você precisará adicionar políticas-chave específicas para permitir que as Permissões Verificadas da Amazon encrypt/decrypt acessem recursos em seu nome.

Autorizando o uso de seu AWS KMS chave para permissões verificadas pela Amazon

No mínimo, as Permissões Verificadas da Amazon exigem as seguintes permissões em um chave gerenciada pelo cliente:

- `kms:Encrypt`
- `kms:GenerateDataKeyWithoutPlaintext`
- `kms:DescribeKey`
- `kms:ReEncryptTo`
- `kms:ReEncryptFrom`
- `kms:Decrypt`

Um exemplo de política chave pode ser visto abaixo:

```
{
    "Sid": "Enable AVP to use the KMS key for encrypting project J.A.K. policy
 resources",
    "Effect": "Allow",
    "Principal": {
        "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Encrypt",
        "kms:ReEncryptFrom",
        "kms:ReEncryptTo",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Compreendendo o contexto de origem

O contexto de origem fornece informações sobre a tentativa do chamador de origem de realizar AWS KMS ações em relação a uma determinada chave. Isso evita confusão ou uso indevido de dados criptografados ao vincular o contexto à fonte dos dados.

Os clientes podem utilizar o contexto de origem como condições adicionais em sua política principal, como as seguintes declarações de política principais:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable this account full access to this key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable AVP to retrieve this key's metadata",
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
                }
            }
        },
        {
            "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:Encrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
                }
            }
        }
    ]
}
```

Essa política de chaves permite que as Permissões Verificadas façam AWS KMS chamadas em seu nome, se a conta de origem for a mesma da conta em que essa AWS KMS chave reside. Esses valores devem ser verificáveis ao verificar os registros de AWS CloudTrail auditoria da chave CMK. Para obter mais informações sobre chaves de AWS condição globais, consulte Como usar `aws:SourceArn` nossas chaves de `aws:SourceAccount` condição.

Entendendo o contexto de criptografia

O contexto de criptografia é um conjunto de pares de valores-chave que contêm dados autenticados adicionais para verificações de integridade da criptografia. Quando você inclui um contexto de criptografia em uma solicitação para criptografar dados, vincula AWS KMS criptograficamente o contexto de criptografia aos dados criptografados. Para descriptografar os dados, você deve passar o mesmo contexto de criptografia.

As Permissões Verificadas da Amazon usam o mesmo contexto de criptografia em todas as operações AWS KMS criptográficas e podem ser verificadas em AWS CloudTrail registros quando as Permissões Verificadas fazem AWS KMS chamadas em seu nome para encryption/decryption processos. Por padrão, as Permissões verificadas utilizam os seguintes pares de valores-chave do contexto de criptografia ao criptografar seus recursos:

```
{
    "aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
}
```

As Permissões Verificadas da Amazon também permitem que você acrescente um contexto de criptografia personalizado como parte dos metadados adicionais que você deseja incluir durante os processos. encryption/decryption Isso significa que sua política de chaves pode ser mais refinada na concessão de permissões, como no exemplo abaixo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable this account full access to this key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
```

```
            "Resource": "*"
        },
        {
            "Sid": "Enable AVP to retrieve this key's metadata",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*"
        },
        {
            "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:Encrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/*",
                    "kms:EncryptionContext:policy_owner": "Tim"
                }
            }
        }
    ]
}
```

Essa política de chaves permite que as Permissões Verificadas façam AWS KMS chamadas em seu nome, se o mapa de contexto de criptografia contiver uma chave `aws:verifiedpermissions:policy-store-arn` cujo valor segue o formato de `arn:aws:verifiedpermissions::111122223333:policy-store/*` e também contém um par de valores-chave. `"policy_owner": "Tim"` Veja the section called “Criando um repositório de políticas criptografadas” como definir um contexto de criptografia personalizado.

> **Note**
>
> É recomendável que as políticas de chave com condições baseadas no contexto de criptografia sejam para um subconjunto do mapa de contexto de criptografia, em vez de verificar cada par de valores-chave. O serviço e suas dependências upstream podem adicionar outros pares de valores-chave que não são visíveis para você e podem afetar o acesso às chaves das Permissões Verificadas se a política de chaves permitir condicionalmente, com base na aparência exata do mapa de contexto de criptografia.

Entendendo o kms: ViaService

A chave de `kms:ViaService` condição limita o uso de uma AWS KMS chave às solicitações de AWS serviços especificados. Essa chave de condição só se aplica às sessões de acesso direto (FAS). Para obter mais informações sobre`kms:ViaService`, consulte kms: ViaService no Guia do AWS KMS desenvolvedor.

Por exemplo, a seguinte declaração de política chave usa a chave de `kms:ViaService` condição para permitir que chave gerenciada pelo clientea seja usada para as ações especificadas somente quando a solicitação vem da Amazon Verified Permissions na região Leste dos EUA (Norte da Virgínia) em nome de`BrentRole`.

```
{
    "Sid": "Enable AVP to encrypt/decrypt resources using credentials of BrentRole",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/BrentRole"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Encrypt",
        "kms:ReEncryptFrom",
        "kms:ReEncryptTo",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": [
                "verifiedpermissions.us-east-1.amazonaws.com"
```

```
            ]
        }
    }
}
```

Isso é necessário para que as Permissões Verificadas possam transmitir sua identidade, permissões e atributos de sessão quando as Permissões Verificadas fizerem uma solicitação AWS KMS em seu nome para encryption/decryption. Para obter mais informações sobre solicitações do FAS, consulte Sessões de acesso direto no Guia IAM do usuário.

Concluído AWS KMS Política chave

Com base nos conceitos das seções anteriores, esse é um exemplo de política de chaves que permitirá que a Amazon Verified Permissions use uma CMK para encryption/decryption:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable this account full access to this key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable AVP to retrieve this key's metadata",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
                }
```

```
            }
        },
        {
            "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
            "Effect": "Allow",
            "Principal": {
                "Service": "verifiedpermissions.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:Encrypt",
                "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/*",
                    "kms:EncryptionContext:policy_owner": "Tim"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "Enable AVP to encrypt/decrypt resources using credentials of
 BrentRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/BrentRole"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:Encrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "verifiedpermissions.us-east-1.amazonaws.com"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/*",
                    "kms:EncryptionContext:policy_owner": "Tim"
                }
            }
        },
        {
            "Sid": "Enable AVP to describe this key using credentials of BrentRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/BrentRole"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "verifiedpermissions.us-east-1.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Warning

Tenha cuidado ao modificar AWS KMS políticas de chaves para chaves que já estão em uso pela Amazon Verified Permissions. Embora o Verified Permissions valide as permissões de criptografia e descriptografia quando você configura inicialmente uma AWS KMS chave durante a criação de recursos de nível superior, ele não pode verificar alterações de política subsequentes sob demanda. A remoção inadvertida das permissões necessárias pode interromper suas decisões de autorização e os fluxos regulares do serviço de Permissões

Verificadas. Para obter orientação sobre como solucionar erros comuns relacionados a chave gerenciada pelo cliente s nas Permissões Verificadas da Amazon, consultethe section called "Solucionar problemas de chaves gerenciadas pelo cliente nas permissões verificadas da Amazon".

Obrigatório IAM Políticas para recursos criptografados

Os clientes que chamarem as Permissões Verificadas por meio de uma IAM função em sua conta precisarão garantir que a IAM política correspondente tenha as permissões adequadas para utilizá-las chave gerenciada pelo cliente para criptografia e descriptografia de recursos.

Para criar repositórios de políticas criptografados por um chave gerenciada pelo cliente, a IAM política a seguir ilustra as ações mínimas necessárias AWS KMS e de permissões verificadas para fazer isso:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "verifiedpermissions:CreatePolicyStore",
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "kms:Decrypt",
                "kms:Encrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:DescribeKey",
                "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

> **Note**
>
> Para recuperar (operações Get* e List*) e excluir repositórios de políticas criptografados por um chave gerenciada pelo cliente, nenhuma permissão adicional é necessária.

Para atualizar um repositório de políticas criptografado por a chave gerenciada pelo cliente, recuperar (operações Get* e List*), atualizar e excluir recursos secundários de um repositório de políticas criptografado por a chave gerenciada pelo cliente, a política a seguir IAM ilustra o mínimo necessário AWS KMS e as ações de Permissões Verificadas fazem isso:

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

Como uma IAM política única, os clientes podem simplesmente adicionar o seguinte à política de IAM funções:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "verifiedpermissions:*",
            "Resource": "*",
```

```
            "Effect": "Allow"
        },
        {
            "Action": [
                "kms:Decrypt",
                "kms:Encrypt",
                "kms:ReEncryptTo",
                "kms:ReEncryptFrom",
                "kms:DescribeKey",
                "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Gerenciando repositórios de políticas criptografadas

Os repositórios de políticas são o contêiner básico que conterá todos os recursos de políticas relacionados. Para obter mais informações sobre repositórios de políticas e a hierarquia de recursos secundários, consulte Armazenamentos de políticas de permissões verificadas da Amazon no Guia do usuário de permissões verificadas da Amazon.

Ao criar um repositório de políticas em Permissões verificadas, você pode ativar a criptografia em repouso usando AWS KMS chaves. Isso garante que:

- Todas as operações de leitura, atualização e exclusão nos repositórios de políticas e seus recursos secundários utilizarão o fornecido chave gerenciada pelo cliente para processos de descriptografia
- Qualquer chamada de decisão de autorização (ou seja IsAuthorized BatchIsAuthorized, IsAuthorizedWithToken,, etc.) usará o fornecido chave gerenciada pelo cliente para processos de descriptografia

Criando um repositório de políticas criptografadas

Antes de criar um repositório de políticas criptografado, certifique-se de que o que chave gerenciada pelo cliente você está usando tenha as declarações de política chave adequadas definidas para as Permissões Verificadas da Amazon para utilizar a chave encryption/decryption. Veja the section

called “Autorizando o uso de seu AWS KMS chave para permissões verificadas pela Amazon” quais permissões são necessárias.

Usando AWS CLI:

```
aws verifiedpermissions create-policy-store --region us-east-1 --encryption-settings
 file://encrypted.json --validation-settings "{\"mode\": \"OFF\"}"
```

Onde `encrypted.json` parece:

```
{
    "kmsEncryptionSettings": {
        "key": "arn:aws:kms:us-east-1:111122223333:key/12345678-90ab-cdef-ghij-
klmnopqrstuv",
        "encryptionContext": {
            "<ENCRYPTION_CONTEXT_KEY_1>": "<ENCRYPTION_CONTEXT_VALUE_1>",
            "<ENCRYPTION_CONTEXT_KEY_2>": "<ENCRYPTION_CONTEXT_VALUE_2>",
            ...
        }
    }
}
```

Certifique-se de substituir por `key` seu chave gerenciada pelo cliente ARN e de substituir `<ENCRYPTION_CONTEXT_KEY>` e emparelhar com os `<ENCRYPTION_CONTEXT_VALUE>` pares de `encryptionContext` valores-chave desejados. `encryptionContext`pode ser omitido completamente se nenhuma adição de pares de valores-chave for desejada.

> **Important**
>
> Não inclua o par de valores-chave `aws:verifiedpermissions:policy-store-arn` em seu contexto de criptografia personalizado. Isso é adicionado automaticamente e resultará em erros de validação se fizer parte dos pares de valores-chave do contexto de criptografia personalizado que você passou.

Para obter mais informações sobre as APIs disponíveis de recursos secundários de um repositório de políticas, consulte Ações no Guia de referência da API Amazon Verified Permissions.

> **Note**
>
> Se os recursos AWS KMS chave gerenciada pelo cliente em uso por seus recursos de Permissões Verificadas da Amazon forem excluídos, desativados ou inacessíveis devido a uma política de AWS KMS chaves incorreta, a descriptografia dos recursos falhará, resultando em decisões de autorização obsoletas. A perda de acesso pode ser temporária (uma política de chave pode ser corrigida) ou permanente (uma chave excluída não pode ser restaurada), dependendo das circunstâncias. Recomendamos que você restrinja o acesso a operações críticas, como excluir ou desativar a AWS KMS chave. Além disso, recomendamos que sua organização configure procedimentos AWS de acesso rápidos para garantir que seus usuários privilegiados possam acessar AWS no caso improvável de as Permissões Verificadas da Amazon ficarem inacessíveis.

Monitorando a interação de permissões verificadas da Amazon com AWS KMS

Você pode monitorar o uso de suas chave gerenciada pelo cliente permissões verificadas pela Amazon. AWS CloudTrail Cada solicitação AWS KMS por meio de Permissões Verificadas inclui o contexto de criptografia e o ARN da chave que está sendo utilizado (seu chave gerenciada pelo cliente) nos parâmetros da solicitação:

Exemplo de entrada de AWS CloudTrail registro para`GenerateDataKeyWithoutPlaintext`:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "verifiedpermissions.amazonaws.com"
    },
    "eventTime": "2025-09-28T16:51:04Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "verifiedpermissions.amazonaws.com",
    "userAgent": "verifiedpermissions.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwx",
        "encryptionContext": {
```

```
            "aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012",
            "policy_store_editor": "Janus"
        },
        ...
    },
    ...
}
```

Exemplo de entrada de AWS CloudTrail registro para`Decrypt`:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "verifiedpermissions.amazonaws.com"
    },
    "eventTime": "2025-09-28T16:53:21Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "verifiedpermissions.amazonaws.com",
    "userAgent": "verifiedpermissions.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwx",
        "encryptionContext": {
            "aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012",
            "policy_store_owner": "Elias"
        }
    },
    ...
}
```

Exemplo de entrada de AWS CloudTrail registro para`ReEncrypt`:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "verifiedpermissions.amazonaws.com"
```

```
    },
    "eventTime": "2025-09-28T16:51:04Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReEncrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "verifiedpermissions.amazonaws.com",
    "userAgent": "verifiedpermissions.amazonaws.com",
    "requestParameters": {
        "sourceKeyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwx",
        "destinationEncryptionContext": {
            "aws:verifiedpermissions:policy-store-arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
        },
        "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "destinationKeyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-
ijkl-4567-mnopqrstuvwx",
        "sourceEncryptionContext": {
            "aws:verifiedpermissions:policy_store_arn":
 "arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
        },
        "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        ...
    },
    ...
}
```

Observe que as entradas de registro incluem a `invokedBy` referência ao principal da Amazon Verified Permissions e `encryptionContext/sourceEncryptionContext/destinationEncryptionContext` a inclusão no mapa. `requestParameters`

Exemplo de entrada de AWS CloudTrail registro para`DescribeKey`:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "verifiedpermissions.amazonaws.com"
    },
    "eventTime": "2025-09-28T16:51:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-east-1",
```

```
    "sourceIPAddress": "verifiedpermissions.amazonaws.com",
    "userAgent": "verifiedpermissions.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwx"
    },
    ...
}
```

Observe que a entrada de registro inclui a `invokedBy` referência à entidade principal da Amazon Verified Permissions.

Para obter mais informações sobre entradas de AWS CloudTrail registro, consulte Entendendo AWS CloudTrail eventos no Guia AWS CloudTrail do usuário.

Limitações

Este tópico descreve as limitações atuais das permissões verificadas e a utilização de chave gerenciada pelo cliente s para criptografia de recursos.

- Você não pode desativar a criptografia para um repositório de políticas depois de ativada
- Depois de criar um repositório de políticas sem criptografia, você não pode atualizar o repositório de políticas para ser criptografado por um chave gerenciada pelo cliente
- Depois de revogar o acesso das Permissões Verificadas a um chave gerenciada pelo cliente repositório de políticas criptografado existente, existe a possibilidade de decisões de autorização obsoletas.
- Depois de criar um repositório de políticas com um chave gerenciada pelo cliente, você não pode modificar valores de contexto de criptografia personalizados; eles são valores estáticos definidos durante a criação do repositório de políticas criptografado

Solucionar problemas de chaves gerenciadas pelo cliente nas permissões verificadas da Amazon

Este tópico descreve erros comuns chave gerenciada pelo cliente relacionados que você pode encontrar ao usar as Permissões Verificadas da Amazon e fornece etapas de solução de problemas para resolvê-los.

Acesso negado: AWS KMS Problema de permissão

Erro: “O serviço ou o chamador não está autorizado a usar a AWS KMS chave fornecida porque o recurso não existe nesta região, nenhuma política baseada em recursos permite o acesso ou uma política baseada em recursos nega explicitamente o acesso”

Isso pode significar que o serviço ou o chamador não têm as permissões de `kms:*` ação necessárias em sua IAM política/política de AWS KMS chaves ou que a chave referenciada não existe ou não existe mais.

Solução de problemas com AWS CloudTrail:

- Procure `kms.amazonaws.com` eventos em AWS CloudTrail
- Pesquise o nome do evento da AWS KMS operação que foi identificada como não permitida (ou seja `DecryptReEncrypt`,`GenerateDataKeyWithoutPlaintext`,,`DescribeKey`, etc.)
- Revise os campos `errorCode` e `errorMessage`
- Verifique `userIdentity` para confirmar qual entidade principal tentou a operação

Para resolver esse problema, conceda ao usuário ou IAM diretor as permissões de acesso adequadas à AWS KMS operação em sua IAM política e política de AWS KMS chaves. Para obter mais informações, consulte the section called “Concluído AWS KMS Política chave”.

Exceção de validação: AWS KMS Configuração da chave

Erro: “A AWS KMS chave configurada não tem uma configuração válida”

Isso significa que a chave referenciada não pode ser usada pelo serviço para chave gerenciada pelo cliente criptografia devido à sua configuração atual. Os motivos podem incluir a desativação da chave, a chave não é suportada EncryptionAlgorithm ou a chave tem um tipo não suportado KeyUsage .

Exceção de limitação: AWS KMS Limites de taxa

Erro: “Você excedeu a taxa na qual você pode ligar” AWS KMS

Esse erro significa que você excedeu o AWS KMS limite de operações criptográficas de sua chave:. https://docs.aws.amazon.com/kms/latest/developerguide/requests-per-second.html

Informações relacionadas

- Gerenciando repositórios de políticas de permissões verificadas

- AWS KMS Práticas recomendadas
- AWS KMS Contexto de criptografia
- AWS CloudTrail Integration
- AWS CloudTrail Exemplos de entrada de registro

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

- Público
- Autenticação com identidades
- Gerenciar o acesso usando políticas
- Como o Amazon Verified Permissions funciona com IAM
- IAM políticas para permissões verificadas
- Exemplos de políticas baseadas em identidade para Amazon Verified Permissions
- AWS políticas gerenciadas 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 com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solução de problemas de identidade e acesso do Amazon Verified Permissions).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como o Amazon Verified Permissions funciona com IAM)

- IAM administrador - escreva políticas para gerenciar o acesso (consulteExemplos de políticas baseadas em identidade para Amazon Verified Permissions)

Autenticação com identidades

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

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para obter mais informações, consulte AWS Signature versão 4 para solicitações de API no Guia IAM do usuário.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Para tarefas que exigem credenciais de usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do IAM usuário.

Identidade federada

Como prática recomendada, exija que os usuários humanos 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 corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte O que é o IAM Identity Center? no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do IAM Usuário.

Um IAM grupo especifica uma coleção de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para obter mais informações, consulte Casos de uso para IAM usuários no Guia IAM do usuário.

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 Console de gerenciamento da AWS 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, é possível 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 por ele. Para obter informações sobre funções para federação, consulte Criar uma função para um provedor de identidade terceirizado (federação) no Guia IAM do usuário. Se 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 do Centro de Identidade do AWS IAM .
- 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 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.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do IAM usuário.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um IAM administrador cria IAM políticas e as adiciona às funções, que os usuários podem então assumir. IAM as políticas definem as permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia do IAM usuário.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre políticas gerenciadas e em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do IAM usuário.

Políticas baseadas em recursos

Políticas baseadas em atributos são documentos de políticas JSON que você anexa a um atributo. Os exemplos incluem políticas de confiança de IAM funções e políticas Amazon S3 de bucket. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário especificar uma entidade principal em uma política baseada em recursos.

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 diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Amazon S3, AWS WAF, e a Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a 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 que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões — defina as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade. Para obter mais informações, consulte Limites de permissões para IAM entidades no Guia IAM do usuário.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.

- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. 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 recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de políticas	Não
ACLs	Não
ABAC (tags em políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Não

IAM recurso	Suporte do Verified Permissions
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 Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente 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. 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 Amazon S3 de bucket. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário 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. 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 podem ser usadas para permitir ou negar acesso em uma política. 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. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de recursos de Permissões Verificadas e seus ARNs, consulte Tipos de recursos definidos pelas Permissões Verificadas da Amazon na Referência de Autorização de 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` especifica quando as instruções são executadas com base em critérios definidos. 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. Para ver

todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia IAM do usuário.

ACLs em Permissões verificadas

Suporta ACLs	Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às 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)	Sim

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a IAM entidades e AWS recursos e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

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 for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do IAM usuário. 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

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte Credenciais de segurança temporárias no IAM e Serviços da AWS que funcionam IAM no Guia do IAM usuário.

Permissões de entidade principal entre serviços para Verified Permissions

Suporta permissões de entidades principais	Sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

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 uma função para delegar permissões a um AWS service (Serviço da AWS) no Guia do IAM usuário.

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 assumir o perfil 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 Perfil

vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

IAM políticas para permissões verificadas

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

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 perfis podem realizar, em quais atributos 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). 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 você pode usar em uma política JSON, consulte a referência aos elementos da política IAM JSON no Guia do IAM usuário.

Ação	Descrição
CreateIdentitySource	Ação para criar uma nova fonte de identidade.
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.
CreatePolicyStore	Ação para criar um novo armazenamento de políticas.
CreatePolicyTemplate	Ação para criar um novo modelo de política.
DeleteIdentitySource	Ação para excluir uma fonte de identidade.

Ação	Descrição
DeletePolicy	Ação para excluir uma política de um armazenamento de políticas.
DeletePolicyStore	Ação para excluir um armazenamento de políticas.
DeletePolicyTemplate	Ação para excluir um modelo de política.
GetIdentitySource	Ação para obter uma fonte de identidade.
GetPolicy	Ação para recuperar informações sobre uma política específica.
GetPolicyStore	Ação para recuperar informações sobre um repositório de políticas especificado.
GetPolicyTemplate	Ação para obter um modelo de política.
GetSchema	Ação para obter um esquema.
IsAuthorized	Ação para obter uma resposta de autorização com base nos parâmetros descritos na solicitação de autorização.
IsAuthorizedWithToken	Ação para obter uma resposta de autorização com base nos parâmetros descritos na solicitação de autorização em que o principal vem de um token de identidade.
ListIdentitySources	Ação para listar todas as fontes de identidade no Conta da AWS.
ListPolicies	Ação para listar todas as políticas de um armazenamento de políticas.
ListPolicyStores	Ação para listar todos os repositórios de políticas no Conta da AWS.

Ação	Descrição
ListPolicyTemplates	Ação para listar todos os modelos de política no Conta da AWS.
ListTagsForResource	Ação para listar todas as tags de um recurso.
PutSchema	Ação para adicionar um esquema a um repositório de políticas.
TagResource	Ação para adicionar uma tag a um recurso.
UpdateIdentitySource	Ação para atualizar uma fonte de identidade.
UpdatePolicy	Ação para atualizar uma política em um repositório de políticas.
UpdatePolicyStore	Ação para atualizar um repositório de políticas.
UpdatePolicyTemplate	Ação para atualizar um modelo de política.
UntagResource	Ação para remover uma tag de um recurso.

Exemplo IAM de política para permissão para a CreatePolicy ação:

JSON

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

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 Console de gerenciamento da AWS, 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 pelas Permissões verificadas, incluindo o formato ARNs de cada um dos tipos de recursos, consulte Ações, recursos e chaves de condição para permissões verificadas da Amazon na Referência de autorização de serviço.

Tópicos

- Práticas recomendadas de política
- Uso do console do Verified Permissions
- Permitir que os usuários visualizem suas próprias permissões

Práticas recomendadas 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 sua 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 recursos 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, é possível escrever 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 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 cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validar políticas com o IAM Access Analyzer no Guia do IAM usuário.
- 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 Acesso seguro à API com MFA no Guia do IAM usuário.

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. Caso crie uma política baseada em identidade mais restritiva 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 à 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 os 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": "*"
        }
```

```
    ]
}
```

AWS políticas gerenciadas para Amazon Verified Permissions

Para adicionar permissões a usuários, grupos e funções, é mais fácil usar políticas AWS gerenciadas do que escrever políticas você mesmo. É preciso tempo e experiência para criar políticas gerenciadas pelo IAM cliente que forneçam à sua equipe somente as permissões necessárias. Para começar rapidamente, você pode usar nossas políticas AWS gerenciadas. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua Conta da AWS. Para obter mais informações sobre políticas AWS gerenciadas, consulte políticas AWS gerenciadas no Guia IAM do usuário.

AWS os serviços mantêm e atualizam as políticas AWS gerenciadas. Você não pode alterar as permissões nas políticas AWS gerenciadas. Os serviços ocasionalmente acrescentam permissões adicionais a uma política gerenciada pela AWS para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. É mais provável que os serviços atualizem uma política gerenciada pela AWS quando um novo recurso for iniciado ou novas operações se tornarem disponíveis. Os serviços não removem as permissões de uma política AWS gerenciada, portanto, as atualizações de políticas não violarão suas permissões existentes.

Além disso, AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política ReadOnlyAccess AWS gerenciada fornece acesso somente de leitura a todos os AWS serviços e recursos. Quando um serviço lança um novo recurso, AWS adiciona permissões somente de leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de funções de trabalho, consulte políticas AWS gerenciadas para funções de trabalho no Guia IAM do usuário.

AWS política gerenciada: AmazonVerifiedPermissionsFullAccess

A política `AmazonVerifiedPermissionsFullAccess` gerenciada concede acesso total às permissões verificadas. Para trabalhar com fontes de identidade Amazon Cognito baseadas, você precisará anexar uma política separada, como a AmazonCognitoReadOnlypolítica.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AccountLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:CreatePolicyStore",
        "verifiedpermissions:ListPolicyStores"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PolicyStoreLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:*"
      ],
      "Resource": [
        "arn:aws:verifiedpermissions::*:policy-store/*"
      ]
    }
  ]
}
```

AWS política gerenciada: AmazonVerifiedPermissionsReadOnlyAccess

A política `AmazonVerifiedPermissionsReadOnlyAccess` gerenciada concede acesso somente para leitura às Permissões Verificadas.

Esta política concede acesso a todas as operações de leitura das Permissões Verificadas da Amazon, incluindo a consulta de autorização APIs `IsAuthorized` `IsAuthorizedWithToken` e.

> **Note**
>
> O acesso `BatchIsAuthorized` e `BatchIsAuthorizedWithToken` são concedidos automaticamente quando o acesso é concedido a `IsAuthorized` e`IsAuthorizedWithToken`, respectivamente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AccountLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:ListPolicyStores"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PolicyStoreLevelPermissions",
      "Effect": "Allow",
      "Action": [
        "verifiedpermissions:GetIdentitySource",
        "verifiedpermissions:GetPolicy",
        "verifiedpermissions:GetPolicyStore",
        "verifiedpermissions:GetPolicyTemplate",
        "verifiedpermissions:GetSchema",
        "verifiedpermissions:IsAuthorized",
        "verifiedpermissions:IsAuthorizedWithToken",
        "verifiedpermissions:ListIdentitySources",
        "verifiedpermissions:ListPolicies",
        "verifiedpermissions:ListPolicyTemplates"
      ],
      "Resource": [
        "arn:aws:verifiedpermissions::*:policy-store/*"
      ]
    }
  ]
}
```

Atualizações de permissões verificadas para políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas de Permissões Verificadas desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos

sobre alterações nessa página, assine o feed RSS na página Histórico do documento de permissões verificadas.

Alteração	Descrição	Data
AmazonVerifiedPermissionsFullAccess – Nova política	As Permissões Verificadas adicionaram uma nova política para permitir acesso total às Permissões Verificadas.	11 de outubro de 2024
AmazonVerifiedPermissionsReadOnlyAccess – Nova política	As Permissões Verificadas adicionaram uma nova política para permitir o acesso a todas as operações de leitura das Permissões Verificadas da Amazon, incluindo a consulta de autorização APIs `IsAuthorized` `IsAuthorizedWithToken` e.	11 de outubro de 2024
As permissões verificadas começaram a rastrear as alterações	A Verified Permissions começou a rastrear as alterações em suas políticas AWS gerenciadas.	11 de outubro de 2024

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
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha acessem meus Conta da AWS recursos de Permissões Verificadas

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

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a 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

Nossa nova experiência AWS de inscrição não foi projetada para cargas de trabalho regulamentadas. Se você estiver usando nossa nova experiência de AWS inscrição, mas quiser usá-la AWS para cargas de trabalho regulamentadas, você pode se inscrever AWS (avançado) ou ativar recursos avançados para seu AWS ambiente.

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 e escolha o programa de conformidade no qual você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixando relatórios em AWS Artifact .

Sua responsabilidade de conformidade durante o uso Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade da sua empresa e pelas leis e regulamentações aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a Documentação AWS de segurança.

Resiliência no Amazon Verified Permissions

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. 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 altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Quando você cria um repositório de políticas de permissões verificadas, ele é criado dentro de um indivíduo Região da AWS 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 zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte Infraestrutura AWS global.

Monitoramento de chamadas da API de permissões verificadas da Amazon

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho das Permissões Verificadas da Amazon e de suas outras AWS soluções. AWS fornece as seguintes ferramentas para monitorar as Permissões Verificadas, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- AWS CloudTrailcaptura chamadas de API e eventos relacionados feitos por ou em nome da sua AWS conta e entrega os arquivos de log em um Amazon S3 bucket especificado por você. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

Para obter mais informações sobre o monitoramento de permissões verificadas com CloudTrail, consulteRegistrando chamadas da API de permissões verificadas da Amazon usando AWS CloudTrail.

Registrando chamadas da API de permissões verificadas da Amazon usando AWS CloudTrail

As Permissões Verificadas da Amazon são integradas AWS CloudTrail a um serviço que fornece um registro das ações tomadas por um usuário, função ou AWS serviço nas 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 ativar a entrega contínua de CloudTrail eventos para um Amazon S3 bucket, incluindo eventos para permissões verificadas. Se você não configurar uma trilha, ainda poderá visualizar os eventos de ação de gerenciamento mais recentes no CloudTrail console no Histórico de eventos, mas não os eventos para chamadas de API, com`isAuthorized`. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita às Permissões Verificadas, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais CloudTrail, consulte o Guia AWS CloudTrail do usuário.

Informações de permissões verificadas em CloudTrail

CloudTrail é habilitado em você Conta da AWS quando você cria a conta. Quando a atividade ocorre nas 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 Visualizar eventos com o histórico de CloudTrail eventos.

Para um registro contínuo de eventos em seu Conta da AWS, incluindo eventos para permissões verificadas, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log a um Amazon S3 bucket. 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 AWS partição e entrega os arquivos de log ao Amazon S3 bucket que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir mais detalhadamente com base nos dados do evento coletados nos CloudTrail registros. Para saber mais, consulte:

- Visão geral da criação de uma trilha
- CloudTrail serviços e integrações compatíveis
- Configuração de 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 `CreateIdentitySourceDeletePolicy`, e `ListPolicyStores` ações 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 o seguinte:

- Se a solicitação foi feita com credenciais de usuário root ou 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 AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Eventos de dados como IsAuthorized e não IsAuthorizedWithToken são 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 você 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 Amazon S3 bucket 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 de API públicas, portanto, eles não aparecem em nenhuma ordem específica.

Para chamadas de API de autorização, os elementos de resposta, como a decisão, são incluídos em `additionalEventData` vez de`responseElements`.

Tópicos

- IsAuthorized
- IsAuthorizedWithToken
- BatchIsAuthorized
- BatchIsAuthorizedWithToken
- CreatePolicyStore
- ListPolicyStores
- DeletePolicyStore
- PutSchema
- GetSchema
- CreatePolicyTemplate
- DeletePolicyTemplate
- CreatePolicy
- GetPolicy
- CreateIdentitySource
- GetIdentitySource
- ListIdentitySources

- DeleteIdentitySource

> **Note**
>
> Alguns campos foram ocultados nos exemplos por questão de privacidade dos dados.

IsAuthorized

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `IsAuthorized` API:


```
{
    "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"
}
```

IsAuthorizedWithToken

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `IsAuthorizedWithToken` API:

```
{
    "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": "IsAuthorizedWithToken",
    "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-with-token",
```

```
    "requestParameters": {
        "policyStoreId": "PSEXAMPLEabcdefg111111",
        "resource": {
            "entityType": "PhotoFlash::Photo",
            "entityId": "VacationPhoto94.jpg"
        },
        "action": {
            "actionType": "PhotoFlash::Action",
            "actionId": "ViewPhoto"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "decision": "ALLOW",
        "principal": {
            "entityType": "PhotoFlash::User",
            "entityId": "us-east-1_EXAMPLE|EXAMPLE_SUBJECT_ID"
        }
    },
    "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `BatchIsAuthorized` API:

```
{
    "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"
}
```

BatchIsAuthorizedWithToken

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `BatchIsAuthorizedWithToken` API:

```
{
    "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": "BatchIsAuthorizedWithToken",
    "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.batch-is-authorized-with-
token",
    "requestParameters": {
        "policyStoreId": "PSEXAMPLEabcdefg111111",
        "requests": [
            {
                "action": {
                    "actionType": "PhotoFlash::Action",
```

```
                    "actionId": "ViewPhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            },
            {
                "action": {
                    "actionType": "PhotoFlash::Action",
                    "actionId": "DeletePhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            }
        ]
    },
    "responseElements": null,
    "additionalEventData": {
        "principal": {
            "entityType": "PhotoFlash::User",
            "entityId": "us-east-1_EXAMPLE|EXAMPLE_SUBJECT_ID"
        },
        "results": [
            {
                "request": {
                    "action": {
                        "actionType": "PhotoFlash::Action",
                        "actionId": "ViewPhoto"
                    },
                    "resource": {
                        "entityType": "PhotoFlash::Photo",
                        "entityId": "VacationPhoto94.jpg"
                    }
                },
                "decision": "ALLOW"
            },
            {
                "request": {
                    "action": {
                        "actionType": "PhotoFlash::Action",
                        "actionId": "DeletePhoto"
```

```
                    },
                    "resource": {
                        "entityType": "PhotoFlash::Photo",
                        "entityId": "VacationPhoto94.jpg"
                    }
                },
                "decision": "DENY"
            }
        ]
    },
    "requestID": "c67aa1d9-9fd7-45ad-9a23-e8f9f6dd66c8",
    "eventID": "5ed90624-a4f7-4ef6-a07a-0e6486d30fbe",
    "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `CreatePolicyStore` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `ListPolicyStores` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `DeletePolicyStore` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `PutSchema` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `GetSchema` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `CreatePolicyTemplate` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `DeletePolicyTemplate` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `CreatePolicy` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `GetPolicy` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `CreateIdentitySource` API:

```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `GetIdentitySource` API:


```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `ListIdentitySources` API:


```
{
  "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

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `DeleteIdentitySource` API:


```
{
  "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 de permissões verificadas da Amazon com 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 CloudFormation provisiona e configura esses recursos para você.

Ao usar 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 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 Amazon Cognito usuários e o cliente no local geograficamente mais próximo de onde o Amazon Região da AWS Cognito Identity está disponível. Use esse grupo de usuários recém-criado ao criar a origem de identidade do Verified Permissions.

Permissões e CloudFormation modelos verificados

Para provisionar e configurar recursos para o Verified Permissions e serviços relacionados, você deve entender os modelos do CloudFormation. Os modelos são arquivos de texto formatados em JSON ou YAML. Esses modelos descrevem os recursos que você deseja provisionar em suas CloudFormation pilhas. Se você não estiver familiarizado com JSON ou YAML, você pode usar o CloudFormation Designer para ajudá-lo a começar a usar modelos. CloudFormation Para obter mais informações, consulte O que é CloudFormation Designer? no Guia do AWS CloudFormation usuário.

As Permissões Verificadas oferecem suporte à criação de fontes de identidade, políticas, repositórios de políticas, modelos de políticas e aliases de armazenamento de políticas em 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. CloudFormation Construções ou componentes de nuvem reutilizáveis podem ser usados para criar modelos. CloudFormation Esses modelos podem então ser usados para implantar sua infraestrutura de nuvem.

Para saber mais e baixar o AWS CDK, 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

Saiba mais sobre CloudFormation

Para saber mais sobre isso CloudFormation, consulte os seguintes recursos:

- AWS CloudFormation
- AWS CloudFormation Guia do usuário
- CloudFormation API Reference
- AWS CloudFormation Guia do usuário da interface de linha de comando

Acesse as permissões verificadas da Amazon usando AWS PrivateLink

Você pode usar AWS PrivateLink para criar uma conexão privada entre sua VPC e as Permissões Verificadas da Amazon. Você pode acessar as Permissões Verificadas como se estivessem em sua VPC, sem o uso de um gateway de internet, dispositivo NAT, conexão VPN ou conexão. Direct Connect As instâncias na VPC não precisam de endereços IP públicos para acessar o Verified Permissions.

Estabeleça essa conectividade privada criando um endpoint de interface, habilitado pelo AWS PrivateLink. Criaremos um endpoint de interface de rede 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 saber mais, 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 obter 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`

Crie uma política de endpoint para seu endpoint de interface.

Uma política de endpoint é um recurso do IAM que pode ser anexado ao endpoint de interface. A política de endpoint padrão permite acesso total às permissões verificadas por meio do endpoint da interface. Para controlar o acesso permitido às permissões verificadas da sua VPC, anexe uma política de endpoint personalizada ao endpoint da interface.

Uma política de endpoint especifica as seguintes informações:

- As entidades principais que podem realizar ações (Contas da AWS, usuários do IAM e perfis do IAM).
- As ações que podem ser realizadas.
- Os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte Controlar o acesso aos serviços usando políticas de endpoint no Guia do AWS PrivateLink .

Exemplo: política de VPC endpoint para ações de permissões verificadas

Veja a seguir um exemplo de uma política de endpoint personalizado. Quando você anexa essa política ao seu endpoint de interface, ela concede acesso às ações de Permissões Verificadas listadas para todos os diretores em todos os recursos.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "verifiedpermissions:IsAuthorized",
            "verifiedpermissions:IsAuthorizedWithToken",
            "verifiedpermissions:GetPolicy"
```

```
        ],
        "Resource":"*"
      }
   ]
}
```

Cotas do Amazon Verified Permissions

Você Conta da AWS tem cotas padrão, anteriormente chamadas de limites, para cada AWS serviço. Salvo indicação em contrário, cada cota é Region-specific. 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
- Cotas para hierarquias
- Cotas para operações por segundo

Cotas para recursos

Nome	Padrão	Ajustá	Description
Armazenamentos de políticas por região por conta	Cada região suportada: 30.000	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	Description
			um armazenamento de políticas.
Aliases do repositório de políticas por repositório de políticas	10	Sim	O número máximo de aliases do repositório de políticas que você pode associar a um único repositório de políticas.
Tamanho da solicitação de autorização[1]	1 MB	Não	Tamanho máximo de uma solicitação de autorização.
Tamanho da política	10,000 bytes	Sim	Tamanho máximo de uma política individual.
Tamanho do esquema	100,000 bytes	Sim	O tamanho máximo do esquema de um repositór io de políticas.
Namespaces por esquema de armazenamento de políticas	100	Sim	O número máximo de namespaces que você pode definir em um esquema de armazenam ento de políticas.
Tamanho da política por recurso	200.000 bytes[2]	Sim	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 limite padrão para o tamanho total de todas as políticas que fazem referência a um único recurso é de 200.000 bytes. As políticas que não especificam um recurso compartilham um total separado de 200.000 bytes para o `"unspecified"` recurso. Para obter detalhes sobre como as Permissões

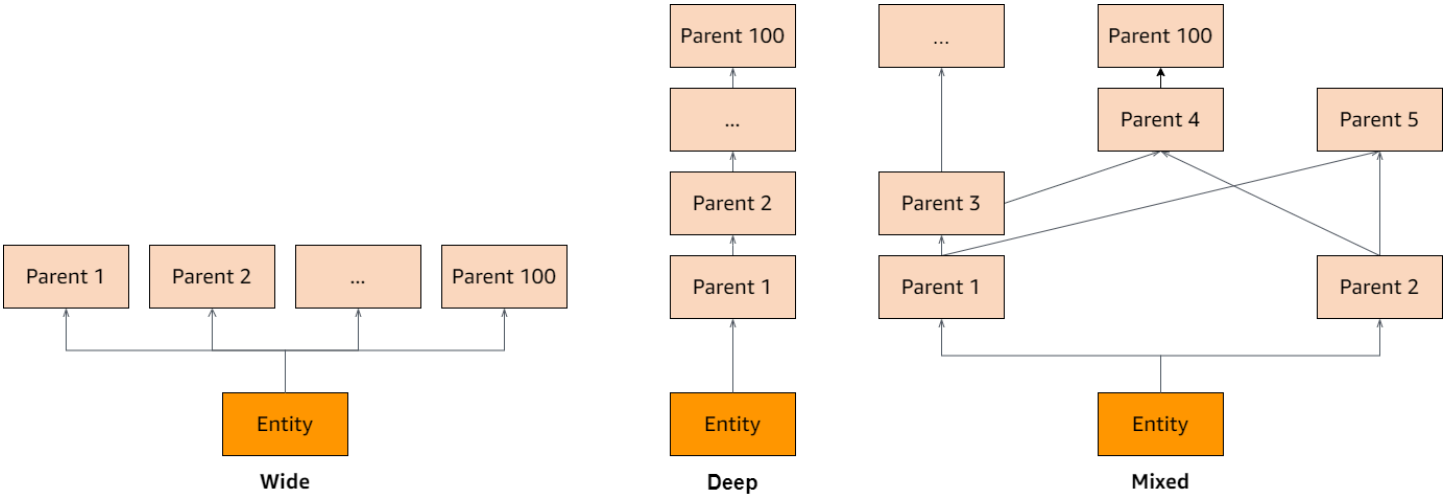
Verificadas calculam essa cota, exemplos e estratégias para permanecer dentro dela, consulte. Tamanho da política por recurso

Cotas para hierarquias

> **Note**
>
> Cada cota parental transitiva se aplica a cada entidade individualmente e conta os pais diretos e indiretos (transitivos) no total. Por exemplo, se o limite de pais transitivos por diretor for 100, um diretor poderá ter 100 grupos de pais diretos sem grupos aninhados. Como alternativa, um diretor pode ter 10 grupos de pais diretos, cada um pertencendo a 9 grupos adicionais de pais. Qualquer combinação que totalize 100 pais para esse diretor é válida.

Nome	Padrão	Ajustá	Description
Pais transitivos por entidade principal	100	Não	Número máximo de pais transitivos para cada entidade principal.
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	Description
BatchGetPolicy solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de BatchGetPolicy solicitaç ões por segundo por repositório de políticas.
BatchIsAuthorized solicitações por segundo por região por repositório de políticas	Cada região compatível: 30	Sim	O número máximo de BatchIsAuthorized solicitações por segundo por repositório de políticas.
BatchIsAuthorizedWithToken solicitaç ões por segundo por região por repositório de políticas	Cada região compatível: 30	Sim	O número máximo de BatchIsAuthorizedW ithToken solicitações por

Nome	Padrão	Ajustá	Description
			segundo por repositório de políticas.
CreateIdentitySource solicitações por segundo por região por repositório de políticas	Cada região compatível: 1	Sim	O número máximo de CreateIdentitySour ce solicitações por segundo por repositório de políticas.
CreatePolicy solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de CreatePolicy solicitações por segundo por repositór io de políticas.
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 repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de CreatePolicyTempla te solicitações por segundo por repositório de políticas.
DeleteIdentitySource solicitações por segundo por região por repositório de políticas	Cada região compatível: 1	Sim	O número máximo de DeleteIdentitySour ce solicitações por segundo por repositório de políticas.
DeletePolicy solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de DeletePolicy solicitações por segundo por repositór io de políticas.

Nome	Padrão	Ajustá	Description
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 repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de DeletePolicyTempla te solicitações por segundo por repositório de políticas.
GetIdentitySource solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de GetIdentitySource solicitações por segundo por repositório de políticas.
GetPolicy solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de GetPolicy solicitações por segundo por repositório de políticas.
GetPolicyStore solicitações por segundo por região por conta	Cada região com suporte: 10	Sim	O número máximo de GetPolicyStore solicitaç ões por segundo.
GetPolicyTemplate solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de GetPolicyTemplate solicitações por segundo por repositório de políticas.
GetSchema solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de GetSchema solicitações por segundo por repositór io de políticas.

Nome	Padrão	Ajustá	Description
IsAuthorized solicitações por segundo por região por repositório de políticas	Cada região compatível: 200	Sim	O número máximo de IsAuthorized solicitações por segundo por repositório de políticas.
IsAuthorizedWithToken solicitações por segundo por região por repositório de políticas	Cada região compatível: 200	Sim	O número máximo de IsAuthorizedWithToken solicitações por segundo por repositório de políticas.
ListIdentitySources solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de ListIdentitySources solicitações por segundo por repositório de políticas.
ListPolicies solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de ListPolicies solicitações por segundo por repositório de políticas.
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 repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de ListPolicyTemplates solicitações por segundo por repositório de políticas.
PutSchema solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de PutSchema solicitações por segundo por repositório de políticas.

Nome	Padrão	Ajustá	Description
UpdateIdentitySource solicitações por segundo por região por repositório de políticas	Cada região compatível: 1	Sim	O número máximo de UpdateIdentitySource solicitações por segundo por repositório de políticas.
UpdatePolicy solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de UpdatePolicy solicitações por segundo por repositório de políticas.
UpdatePolicyStore solicitações por segundo por região por conta	Cada região com suporte: 10	Não	O número máximo de UpdatePolicyStore solicitações por segundo.
UpdatePolicyTemplate solicitações por segundo por região por repositório de políticas	Cada região com suporte: 10	Sim	O número máximo de UpdatePolicyTemplate solicitações por segundo por repositório de políticas.

Permissões verificadas da Amazon e termos e conceitos da linguagem da política Cedar

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
- Apelido do repositório de políticas
- Nome da política
- Nome do modelo de política
- Políticas atendidas
- Diferenças entre as permissões verificadas da Amazon e a linguagem de política do 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. Inclui uma decisão de permitir ou negar, além de informações adicionais, como IDs as 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.

Apelido do repositório de políticas

Um alias de armazenamento de políticas é um nome amigável para um repositório de políticas. Você pode usar um alias de armazenamento de políticas para identificar um repositório de políticas em qualquer operação de Permissões Verificadas que aceite um `policyStoreId` parâmetro. Os aliases do repositório de políticas são AWS recursos independentes com seus próprios. ARNs Cada alias é associado a um repositório de políticas por vez, e vários aliases podem ser associados ao mesmo repositório de políticas. Para obter mais informações, consulte Aliases da loja de políticas de permissões verificadas da Amazon.

Nome da política

Um nome de política é um nome amigável opcional para uma política. Os nomes das políticas devem ser exclusivos para todas as políticas no repositório de políticas e prefixados com`name/`. Você pode usar um nome de política no lugar do ID da política em operações do plano de controle que aceitam um `policyId` parâmetro. Os nomes podem ser definidos ao criar ou atualizar uma política. Somente `GetPolicy` e `ListPolicies` retorne o nome na saída.

Nome do modelo de política

Um nome de modelo de política é um nome amigável opcional para um modelo de política. Os nomes dos modelos de política devem ser exclusivos para todos os modelos de política no repositório de políticas e prefixados com`name/`. Você pode usar um nome de modelo de política no lugar do ID do modelo de política em operações do plano de controle que aceitam um `policyTemplateId` parâmetro. Os nomes podem ser definidos ao criar ou atualizar um modelo de política. Somente `GetPolicyTemplate` e `ListPolicyTemplates` retorne o nome na saída.

Políticas atendidas

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

Diferenças entre as permissões verificadas da Amazon e a linguagem de política do 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.

> **Note**
>
> Em maio de 2026, as Permissões Verificadas estavam alinhadas com o Cedar e agora oferecem suporte a vários namespaces.

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, como para propriedades ou namespaces.

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`.

Formatação de entidades

A formatação JSON de entidades em Permissões verificadas usando o `entityList` parâmetro difere do Cedar das seguintes maneiras:

- 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`

Example- Listas

Os exemplos a seguir mostram como uma lista de entidades é expressa em Cedar e Permissões Verificadas, respectivamente.

Cedar

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

Verified Permissions

```
{
  "Set": [
```

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

Example- Avaliação de políticas

Os exemplos a seguir mostram como as entidades são formatadas para avaliar uma política em uma solicitação de autorização no Cedar e no Verified Permissions, respectivamente.

Cedar

```
[
    {
        "uid": {
            "type": "PhotoApp::User",
            "id": "alice"
        },
        "attrs": {
            "age": 25,
            "name": "alice",
            "userId": "123456789012"
        },
```

```
        "parents": [
            {
                "type": "PhotoApp::UserGroup",
                "id": "alice_friends"
            },
            {
                "type": "PhotoApp::UserGroup",
                "id": "AVTeam"
            }
        ]
    },
    {
        "uid": {
            "type": "PhotoApp::Photo",
            "id": "vacationPhoto.jpg"
        },
        "attrs": {
            "private": false,
            "account": {
                "__entity": {
                    "type": "PhotoApp::Account",
                    "id": "ahmad"
                }
            }
        },
        "parents": []
    },
    {
        "uid": {
            "type": "PhotoApp::UserGroup",
            "id": "alice_friends"
        },
        "attrs": {},
        "parents": []
    },
    {
        "uid": {
            "type": "PhotoApp::UserGroup",
            "id": "AVTeam"
        },
        "attrs": {},
        "parents": []
    }
```

```
]
```

Verified Permissions

```
[
    {
        "Identifier": {
            "EntityType": "PhotoApp::User",
            "EntityId": "alice"
        },
        "Attributes": {
            "age": {
                "Long": 25
            },
            "name": {
                "String": "alice"
            },
            "userId": {
                "String": "123456789012"
            }
        },
        "Parents": [
            {
                "EntityType": "PhotoApp::UserGroup",
                "EntityId": "alice_friends"
            },
            {
                "EntityType": "PhotoApp::UserGroup",
                "EntityId": "AVTeam"
            }
        ]
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::Photo",
            "EntityId": "vacationPhoto.jpg"
        },
        "Attributes": {
            "private": {
                "Boolean": false
            },
            "account": {
                "EntityIdentifier": {
```

```
                    "EntityType": "PhotoApp::Account",
                    "EntityId": "ahmad"
                }
            }
        },
        "Parents": []
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::UserGroup",
            "EntityId": "alice_friends"
        },
        "Parents": []
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::UserGroup",
            "EntityId": "AVTeam"
        },
        "Parents": []
    }
]
```

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

Objeto	Limite do Verified Permissions (em bytes)	Limite do Cedar
Tamanho do esquema	100.000	Nenhum
Tipo de entidade	200	Nenhum
ID da apólice	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. 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.

Perguntas frequentes sobre a atualização de permissões verificadas da Amazon para o Cedar 4

A Amazon Verified Permissions está atualizando a versão do Cedar que usa da versão 2 para a versão 4. O Cedar é a linguagem de código aberto que você usa para escrever políticas, modelos de políticas e esquemas em seus repositórios de políticas. Com o suporte do Cedar 4 nas Permissões Verificadas, você pode usar novos recursos, como as tags de `is` operador e entidade, para criar políticas mais expressivas.

O Amazon Verified Permissions está atualizando automaticamente os repositórios de políticas para o Cedar 4. No entanto, algumas políticas, esquemas e solicitações de autorização escritas para o Cedar 2 são incompatíveis com o Cedar 4. Se esse for o caso do seu repositório de políticas, não o atualizaremos automaticamente. Talvez seja necessário fazer alterações em suas políticas, modelos de políticas, esquemas ou código de aplicativo antes de atualizar para o Cedar 4.

Tópicos

- Por que recebo uma mensagem de erro dizendo que o Cedar 2 não é mais suportado quando ligo para Amazon Verified Permissions?
- Por que algumas políticas, modelos de políticas e esquemas não são compatíveis com o Cedar 4?
- Como posso saber se meu repositório de políticas está usando o Cedar 2 ou o Cedar 4?
- Como faço o upgrade para o Cedar 4?
- Posso fazer o downgrade da minha loja de políticas do Cedar 4 para o Cedar 2?
- Por que estou recebendo uma mensagem de erro dizendo que meu repositório de políticas está configurado para o Cedar 2?
- Como faço para tornar meu esquema compatível com o Cedar 4?
- Como faço para tornar minhas políticas e modelos compatíveis com o Cedar 4?

Por que recebo uma mensagem de erro dizendo que o Cedar 2 não é mais suportado quando ligo para Amazon Verified Permissions?

A partir de abril de 2026, as APIs de autorização (`IsAuthorized`,`BatchIsAuthorized`, `IsAuthorizedWithToken` e`BatchIsAuthorizedWithToken`) estão sendo desativadas

para repositórios de políticas que usam o Cedar 2. Nesse caso, a seguinte mensagem de erro é retornada:

```
Your policy store uses Cedar 2.x, which is no longer supported. See https://
docs.aws.amazon.com/verifiedpermissions/latest/userguide/cedar4-faq.html or contact AWS
 Support for more information.
```

Se você encontrar essa mensagem de erro ao chamar Permissões verificadas, isso significa que seu repositório de políticas é incompatível com o Cedar 4 e, portanto, não pôde ser migrado para a nova versão. Para continuar usando as Permissões Verificadas da Amazon, crie um novo repositório de políticas, que usará o Cedar 4, ou entre em contato Suporte.

Por que algumas políticas, modelos de políticas e esquemas não são compatíveis com o Cedar 4?

A equipe do Cedar fez várias alterações incompatíveis com versões anteriores desde o Cedar 2, para corrigir bugs e simplificar a linguagem. Essas alterações incluem:

- alterações de sintaxe para políticas, modelos de políticas e esquemas
- um validador de políticas mais preciso, que detecta mais erros
- mudanças no comportamento de funções integradas, como `isInRange`

Para obter uma lista completa de alterações incompatíveis com versões anteriores, procure os itens marcados com (*) no changelog do Cedar.

Como posso saber se meu repositório de políticas está usando o Cedar 2 ou o Cedar 4?

Você pode verificar a versão do Cedar que seu repositório de políticas usa usando o console Amazon Verified Permissions ou usando a GetPolicyStore operação.

> **Note**
>
> Todos os repositórios de políticas na mesma Conta da AWS região usam a mesma versão do Cedar.

Console

Para verificar a versão Cedar de um repositório de políticas (console)

1. Faça login no Console de gerenciamento da AWS e abra o console Amazon Verified Permissions em https://console.aws.amazon.com/verifiedpermissions/.
2. No painel de navegação, escolha Armazenamentos de políticas e, em seguida, escolha o repositório de políticas que você deseja verificar.
3. Selecione Configurações no painel de navegação.
4. Na caixa Detalhes, localize o campo Versão do Cedar.

O campo mostra `CEDAR_2` se seu repositório de políticas está usando o Cedar 2 e `CEDAR_4` se usa o Cedar 4.

CLI

Para verificar a versão Cedar de um repositório de políticas (AWS CLI)

1. Instale e configure o AWS Command Line Interface (AWS CLI), caso ainda não tenha feito isso. Para obter informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.
2. Use o comando `get-policy-store`. No exemplo a seguir, *`policy-store-id`* substitua pelo identificador do seu repositório de políticas:

```
aws verifiedpermissions get-policy-store \
  --policy-store-id policy-store-id
```

O `cedarVersion` campo na saída mostra qual versão do Cedar o repositório de políticas está usando. Por exemplo:

```
{
    "policyStoreId": "ABCDEFG12345678abcdefg",
    "arn": "arn:aws:verifiedpermissions::111122223333:policy-store/
ABCDEFG12345678abcdefg",
    "validationSettings": {
        "mode": "STRICT"
    },
    "createdDate": "2025-06-03T13:09:47.752255+00:00",
    "lastUpdatedDate": "2025-06-03T13:09:47.752255+00:00",
```

```
    "deletionProtection": "ENABLED",
    "cedarVersion": "CEDAR_2"
}
```

O campo mostra `CEDAR_2` se seu repositório de políticas está usando o Cedar 2 e `CEDAR_4` se usa o Cedar 4.

Como faço o upgrade para o Cedar 4?

O Amazon Verified Permissions já atualizou a maioria dos clientes para o Cedar 4. Se você nunca criou um repositório de políticas, qualquer novo repositório de políticas que você criar usará o Cedar 4. Se você já é um cliente, provavelmente já o atualizamos para o Cedar 4. Consulte Como posso saber se meu repositório de políticas está usando o Cedar 2 ou o Cedar 4? para verificar qual versão do Cedar suas lojas de políticas usam.

Se você não tiver sido atualizado, as Permissões Verificadas detectaram uma política, modelo de política, esquema ou solicitação de autorização em um de seus repositórios de políticas que é incompatível com o Cedar 4. Enviaremos uma notificação por e-mail descrevendo quais recursos são incompatíveis no final de 2025. Para atualizar mais cedo, abra uma caixa com Suporte.

> **Important**
>
> Todos os repositórios de políticas no mesmo Conta da AWS usam a mesma versão do Cedar. Se um repositório de políticas em sua conta for incompatível com o Cedar 4, você não poderá usar o Cedar 4 em nenhum repositório de políticas dessa conta.

Posso fazer o downgrade da minha loja de políticas do Cedar 4 para o Cedar 2?

Não. Se você tiver problemas depois que seu repositório de políticas for atualizado para o Cedar 4, abra um caso com. Suporte

Por que estou recebendo uma mensagem de erro dizendo que meu repositório de políticas está configurado para o Cedar 2?

Alguns recursos do Amazon Verified Permissions dependem dos novos recursos do Cedar 4. Se seu repositório de políticas não usar o Cedar 4, você não poderá usar os seguintes campos de API:

- Nas BatchIsAuthorizedWithToken operações IsAuthorizedBatchIsAuthorized, IsAuthorizedWithToken e:
  - `datetime`, `decimal` ou `duration` valores nos `context` campos `attributes` ou

Você não pode usar sintaxe ou tipos de dados em políticas, modelos de políticas ou esquemas introduzidos após o Cedar 2 até que seu repositório de políticas seja atualizado.

Como faço para tornar meu esquema compatível com o Cedar 4?

O console de permissões verificadas pode corrigir automaticamente alguns problemas de compatibilidade em seu esquema. Se o esquema não puder ser corrigido automaticamente, o console mostrará uma lista de erros para você corrigir manualmente.

> **Important**
>
> O editor de código no console Amazon Verified Permissions sempre mostra erros e avisos do Cedar 4, mesmo que seu repositório de políticas use o Cedar 2. Você pode continuar fazendo atualizações de esquema que não são compatíveis com o Cedar 4 usando o botão Salvar alterações ou a API de permissões verificadas.

Para corrigir um esquema usando o console

1. Faça login no Console de gerenciamento da AWS e abra o console Amazon Verified Permissions em verifiedpermissions.
2. No painel de navegação, escolha Armazenamentos de políticas e, em seguida, escolha o repositório de políticas que você deseja verificar.
3. Escolha Esquema no painel de navegação.
4. Se seu esquema puder ser corrigido automaticamente, você verá um banner com os dizeres “Clique em 'Corrigir' para visualizar uma versão compatível”. Selecione Corrigir.

5. Revise as alterações feitas em seu esquema e clique em Visualizar esquema atualizado.
6. Revise o esquema atualizado e clique em Salvar alterações.

Se o esquema não puder ser corrigido automaticamente, você poderá ver uma lista de erros para corrigir você mesmo no console.

1. Abra a página Editar esquema conforme descrito acima.
2. Selecione o modo JSON.
3. Passe o mouse sobre o ícone de erro vermelho na medianiz no lado esquerdo do editor de código. A mensagem de erro é exibida em uma dica de ferramenta.

Aqui estão alguns erros comuns que você pode encontrar e como resolvê-los:

falha ao analisar o esquema do JSON: `` ***field-name***

> Com o Cedar 2, você pode incluir campos arbitrários em partes de esquemas, como definições de tipo, mesmo que eles não tenham nenhum significado como parte de um esquema Cedar. No Cedar 4, isso não é mais permitido. Para resolver esse erro, remova o campo chamado *field-name* do seu esquema JSON. Para obter uma lista de campos de esquema válidos, consulte a documentação do Cedar.

tipo de extensão desconhecido `***extension-name***`

> No Cedar 2, ao declarar um atributo cujo `type` é`Extension`, você pode especificar qualquer valor para o `name` campo, independentemente de o valor ser ou não um nome de tipo de extensão válido. Agora isso é um erro com o Cedar 4. Para resolver isso, *extension-name* substitua por um nome de tipo de extensão válido. Você pode encontrar uma lista de nomes de tipos de extensão válidos na documentação do Cedar.

Se você ainda não tiver certeza de como resolver os erros em seu esquema, entre em contato com Suporte

Como faço para tornar minhas políticas e modelos compatíveis com o Cedar 4?

O console de Permissões verificadas mostra quaisquer erros em sua política ou modelo que o tornem incompatível com o Cedar 4.

Para visualizar os erros de uma política ou modelo no console

1. Faça login no Console de gerenciamento da AWS e abra o console Amazon Verified Permissions em verifiedpermissions.
2. No painel de navegação, escolha Armazenamentos de políticas e, em seguida, escolha o repositório de políticas que você deseja verificar.
3. Escolha Políticas ou Modelos de política no painel de navegação, conforme apropriado.
4. Selecione a política ou o modelo incompatível.
5. Selecione Editar
6. Passe o mouse sobre o ícone de erro vermelho na medianiz no lado esquerdo do editor de código. A mensagem de erro é exibida em uma dica de ferramenta.

Aqui estão alguns erros comuns que você pode encontrar e como resolvê-los:

conjuntos literais vazios são proibidos nas políticas

No Cedar 2, você pode usar a sintaxe `mySet == []` para verificar se um conjunto está vazio. Com o Cedar 4, as políticas que usam essa sintaxe não são mais válidas em relação a um esquema. `mySet == []`Substitua sua apólice por`mySet.isEmpty()`.

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
Exclusão definitiva para aliases do repositório de políticas	Agora você pode excluir definitivamente os aliases do repositório de políticas para exclusão imediata, ignorando o período de reserva de 24 horas`PendingDeletion` .	17 de abril de 2026
Nomes de políticas e nomes de modelos de políticas	Agora você pode atribuir nomes a políticas e modelos de políticas para referenciá-las por meio de nomes amigáveis.	4 de março de 2025
Aliases do repositório de políticas	Agora você pode criar aliases de armazenamento de políticas para se referir aos seus repositórios de políticas por nomes amigáveis.	26 de fevereiro de 2025
Novas políticas AWS gerenciadas	Agora você pode usar as políticas `AmazonVerifiedPermissionsFullAccess` `AmazonVerifiedPermissionsReadOnlyAccess` IAM gerenciadas com permissões verificadas.	11 de outubro de 2024
Fontes de identidade do OIDC	Agora você pode autorizar usuários dos provedores de	8 de junho de 2024

	identidade do OpenID Connect (OIDC).	
Autorização em lote com tokens de origem de identidade	Agora você pode autorizar usuários de um grupo de Amazon Cognito usuários em uma única solicitação de `BatchIsAuthorizedWithToken` API.	5 de abril de 2024
Criando um repositório de políticas com API Gateway	Agora você pode criar um repositório de políticas a partir de uma API e de um grupo de Amazon Cognito usuários existentes.	1º de abril de 2024
Conceitos de contexto e exemplo	Foram adicionadas informações sobre o contexto nas solicitações de autorização com permissões verificadas.	1.º de fevereiro de 2024
Conceitos e exemplos de autorização	Foram adicionadas informações sobre solicitações de autorização com permissões verificadas.	1.º de fevereiro de 2024
AWS CloudFormation integração	As permissões verificadas oferecem suporte à criação de fontes de identidade, políticas, repositórios de políticas e modelos de políticas no CloudFormation.	30 de junho de 2023
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.