aws

Guide de l’utilisateur

Amazon Verified Permissions

Amazon Verified Permissions: Guide de l’utilisateur

Table of Contents

Qu'est-ce que les autorisations vérifiées par Amazon ?

Amazon Verified Permissions est un service de gestion et d'autorisation des autorisations évolutif et précis pour les applications personnalisées que vous avez créées. Les autorisations vérifiées permettent à vos développeurs de créer des applications sécurisées plus rapidement en externalisant les autorisations et en centralisant la gestion et l'administration des politiques. Verified Permissions utilise le langage de politique de Cedar pour définir des autorisations précises pour les utilisateurs de l'application.

Rubriques

Autorisation dans les autorisations vérifiées

Les autorisations vérifiées fournissent une autorisation en vérifiant si un principal est autorisé à effectuer une action sur une ressource dans un contexte donné dans une application personnalisée. Les autorisations vérifiées supposent que le principal a déjà été identifié et authentifié par d'autres moyens, par exemple en utilisant des protocoles tels qu'OpenID Connect, un fournisseur d'hébergement tel qu'Amazon Cognito ou une autre solution d'authentification. Les autorisations vérifiées sont indépendantes de l'endroit où l'utilisateur est géré et de la manière dont il a été authentifié.

Verified Permissions est un service qui permet aux clients de créer, de gérer et de tester des politiques dans le AWS Management Console. Les autorisations sont exprimées en utilisant le langage de politique de Cedar. L'application cliente appelle des API d'autorisation pour évaluer les politiques Cedar stockées avec le service et fournir une décision d'accès indiquant si une action est autorisée.

Langage politique du cèdre

Les politiques d'autorisation dans Verified Permissions sont rédigées en utilisant le langage de politique Cedar. Cedar est un langage open source permettant de rédiger des politiques d'autorisation et de prendre des décisions d'autorisation sur la base de ces politiques. Lorsque vous créez une application, vous devez vous assurer que seuls les utilisateurs autorisés peuvent accéder à l'application et ne peuvent faire que ce que chaque utilisateur est autorisé à faire. Avec Cedar, vous pouvez dissocier votre logique métier de la logique d'autorisation. Dans le code de votre application, vous devez préfacer les demandes adressées à vos opérations par un appel au moteur d'autorisation de Cedar, en demandant « Cette demande est-elle autorisée ? ». Ensuite, l'application peut soit effectuer l'opération demandée si la décision est « autoriser », soit renvoyer un message d'erreur si la décision est « refusée ».

Verified Permissions utilise actuellement la version 2.4 de Cedar.

Pour plus d'informations sur Cedar, consultez les pages suivantes :

- Guide de référence sur le langage politique de Cedar
- GitHubDépôt en cèdre

Avantages des autorisations vérifiées

Accélérez le développement des applications

Accélérez le développement des applications en dissociant les autorisations de la logique métier.

Applications plus sécurisées

Les autorisations vérifiées permettent aux développeurs de créer des applications plus sécurisées.

Fonctionnalités destinées à l'utilisateur final

Les autorisations vérifiées vous permettent de proposer aux utilisateurs finaux des fonctionnalités plus riches pour la gestion des autorisations.

Services connexes

- Amazon Cognito — Amazon Cognito est une plateforme d'identité pour les applications Web et mobiles. Il s'agit d'un annuaire d'utilisateurs, d'un serveur d'authentification et d'un service

d'autorisation pour les jetons d'accès OAuth 2.0 et les informations d'identification AWS . Lorsque vous créez un magasin de politiques, vous avez la possibilité de créer vos principaux et vos groupes à partir d'un groupe d'utilisateurs Amazon Cognito. Pour plus d'informations, consultez le Guide du développeur Amazon Cognito.

- Amazon API Gateway — Amazon API Gateway est un AWS service de création, de publication, de maintenance, de surveillance et de sécurisation des API REST, HTTP et des WebSocket API à n'importe quelle échelle. Lorsque vous créez un magasin de politiques, vous avez la possibilité de créer vos actions et vos ressources à partir d'une API dans API Gateway. Pour plus d'informations sur API Gateway, consultez le guide du développeur d'API Gateway.
- AWS IAM Identity Center— Avec IAM Identity Center, vous pouvez gérer la sécurité de connexion pour les identités de vos employés, également appelés utilisateurs du personnel. IAM Identity Center fournit un endroit unique où vous pouvez créer ou connecter les utilisateurs du personnel et gérer de manière centralisée leur accès à toutes leurs Comptes AWS applications. Pour plus d'informations, consultez le Guide de l'utilisateur AWS IAM Identity Center.

Accès aux autorisations vérifiées

Vous pouvez utiliser les autorisations vérifiées d'Amazon de l'une des manières suivantes.

AWS Management Console

La console est une interface basée sur un navigateur permettant de gérer les autorisations et AWS les ressources vérifiées. Pour plus d'informations sur l'accès aux autorisations vérifiées via la console, consultez Comment se connecter AWS dans le guide de Connexion à AWS l'utilisateur.

- Console Amazon Verified Permissions

AWS Outils de ligne de commande

Vous pouvez utiliser les outils de ligne de AWS commande pour émettre des commandes sur la ligne de commande de votre système afin d'effectuer des autorisations et des AWS tâches vérifiées. L'utilisation de la ligne de commande peut être plus rapide et plus pratique que de la console. Les outils de ligne de commande sont également utiles si vous souhaitez créer des scripts exécutant des tâches AWS .

AWS fournit deux ensembles d'outils de ligne de commande : le AWS Command Line Interface(AWS CLI) et le AWS Tools for Windows PowerShell. Pour plus d'informations sur l'installation et l'utilisation du AWS CLI, consultez le guide de AWS Command Line Interface

l'utilisateur. Pour plus d'informations sur l'installation et l'utilisation des outils pour Windows PowerShell, consultez le guide de AWS Tools for Windows PowerShell l'utilisateur.

- autorisations vérifiées dans la référence des commandes AWS CLI
- Autorisations vérifiées par Amazon dans AWS Tools for Windows PowerShell

AWS SDK

AWS fournit des SDK (kits de développement logiciel) composés de bibliothèques et d'exemples de code pour différents langages de programmation et plateformes (Java, Python, Ruby, .NET, iOS, Android, etc.). Les SDK constituent un moyen pratique de créer un accès programmatique aux autorisations vérifiées et. AWS Par exemple, ils automatisent les tâches telles que la signature cryptographique des demandes, la gestion des erreurs et les nouvelles tentatives automatiques de demande.

Pour en savoir plus et télécharger AWS les SDK, consultez la section Outils pour Amazon Web Services.

Vous trouverez ci-dessous des liens vers la documentation relative aux ressources d'autorisations vérifiées dans différents AWS SDK.

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

AWS Constructions CDK

AWS Cloud Development Kit (AWS CDK) Il s'agit d'un framework de développement logiciel open source permettant de définir l'infrastructure cloud dans le code et de la provisionner via ce dernier. AWS CloudFormation Des constructions, ou des composants cloud réutilisables, peuvent être utilisés pour créer des AWS CloudFormation modèles. Ces modèles peuvent ensuite être utilisés pour déployer votre infrastructure cloud.

Pour en savoir plus et télécharger AWS des CDK, consultez AWS Cloud Development Kit.

Vous trouverez ci-dessous des liens vers la documentation relative aux AWS CDK ressources d'autorisations vérifiées, telles que les constructions.

- Autorisations vérifiées par Amazon L2 CDK Construct

API d'autorisations vérifiées

Vous pouvez accéder aux autorisations vérifiées et par AWS programmation à l'aide de l'API Verified Permissions, qui vous permet d'envoyer des requêtes HTTPS directement au service. Lorsque vous utilisez l'API , vous devez inclure un code pour signer numériquement les demandes à l'aide de vos informations d'identification.

- Guide de référence de l'API Amazon Verified Permissions

Tarification des autorisations vérifiées

Verified Permissions propose une tarification échelonnée basée sur le nombre de demandes d'autorisation par mois effectuées par vos applications auprès de Verified Permissions. La tarification des actions de gestion des politiques est également basée sur le nombre de demandes d'API de politique cURL (URL client) adressées par mois par vos applications aux autorisations vérifiées.

Pour une liste complète des frais et des prix des autorisations vérifiées, consultez les tarifs des autorisations vérifiées sur Amazon.

Pour consulter votre facture, dirigez-vous vers le Tableau de bord de gestion des coûts et de la facturation dans la console AWS Billing and Cost Management. Votre facture contient des liens vers les rapports d'utilisation qui fournissent des détails sur votre facture. Pour en savoir plus sur Compte AWS la facturation, consultez le guide de AWS Billing l'utilisateur.

Si vous avez des questions concernant AWS la facturation, les comptes et les événements, contactez AWS Support.

Termes et concepts relatifs aux autorisations vérifiées par Amazon

Vous devez comprendre les concepts suivants pour utiliser Amazon Verified Permissions.

Concepts d'autorisations vérifiés

- Modèle d'autorisation
- Demande d'autorisation
- Réponse d'autorisation
- Politiques prises en compte
- Données contextuelles
- Déterminer les politiques
- Données sur les entités
- Permissions, autorisations et principes
- Application des politiques
- Boutique Policy
- Politiques satisfaites
- Différences entre les autorisations vérifiées et Cedar

Concepts linguistiques relatifs à la politique du

- Autorisation
- Entité
- Groupes et hiérarchies
- Espaces de noms
- Stratégie
- Modèle de politique
- Schema (Schéma)

Modèle d'autorisation

Le modèle d'autorisation décrit l'étendue des demandes d'autorisation effectuées par l'application et la base d'évaluation de ces demandes. Il est défini en fonction des différents types de ressources, des actions entreprises sur ces ressources et des types de principes qui prennent ces mesures. Il prend également en compte le contexte dans lequel ces mesures sont prises.

Le contrôle d'accès basé sur les rôles (RBAC) est une base d'évaluation dans laquelle les rôles sont définis et associés à un ensemble d'autorisations. Ces rôles peuvent ensuite être attribués à une ou plusieurs identités. L'identité attribuée acquiert les autorisations associées au rôle. Si les autorisations associées au rôle sont modifiées, la modification a automatiquement un impact sur toute identité à laquelle le rôle a été attribué. Cedar peut soutenir les décisions du RBAC en faisant appel à des groupes principaux.

Le contrôle d'accès basé sur les attributs (ABAC) est une base d'évaluation dans laquelle les autorisations associées à une identité sont déterminées par les attributs de cette identité. Cedar peut soutenir les décisions de l'ABAC en utilisant des conditions politiques qui font référence aux attributs du principal.

Le langage de politique Cedar permet de combiner RBAC et ABAC dans une seule politique en permettant de définir des autorisations pour un groupe d'utilisateurs soumis à des conditions basées sur des attributs.

Demande d'autorisation

Une demande d'autorisation est une demande d'autorisations vérifiées faite par une application pour évaluer un ensemble de politiques afin de déterminer si un principal peut effectuer une action sur une ressource dans un contexte donné.

Réponse d'autorisation

La réponse d'autorisation est la réponse à la demande d'autorisation. Il inclut une décision d'autorisation ou de refus, ainsi que des informations supplémentaires, telles que les identifiants des politiques déterminantes.

Politiques prises en compte

Les politiques considérées sont l'ensemble complet des politiques sélectionnées par Verified Permissions pour inclusion lors de l'évaluation d'une demande d'autorisation.

Données contextuelles

Les données contextuelles sont des valeurs d'attributs qui fournissent des informations supplémentaires à évaluer.

Déterminer les politiques

Les politiques de détermination sont les politiques qui déterminent la réponse d'autorisation. Par exemple, si deux politiques sont satisfaites, l'une étant un refus et l'autre une autorisation, la politique de refus sera la politique déterminante. Si plusieurs politiques d'autorisation sont satisfaites et qu'aucune politique d'interdiction n'est satisfaite, alors il existe plusieurs politiques déterminantes. Dans le cas où aucune politique ne correspond et que la réponse est refusée, il n'existe aucune politique déterminante.

Données sur les entités

Les données d'entité sont des données relatives au principal, à l'action et à la ressource. Les données d'entité pertinentes pour l'évaluation des politiques sont l'appartenance à un groupe tout au long de la hiérarchie des entités et les valeurs d'attribut du principal et de la ressource.

Permissions, autorisations et principes

Verified Permissions gère les autorisations détaillées et les autorisations au sein des applications personnalisées que vous créez.

Un mandant est l'utilisateur d'une application, qu'elle soit humaine ou machine, dont l'identité est liée à un identifiant tel qu'un nom d'utilisateur ou un identifiant de machine. Le processus d'authentification détermine si le mandant est réellement l'identité qu'il prétend être.

À cette identité est associé un ensemble d'autorisations d'application qui déterminent ce que le principal est autorisé à faire au sein de cette application. L'autorisation est le processus qui consiste

à évaluer ces autorisations afin de déterminer si un principal est autorisé à effectuer une action particulière dans l'application. Ces autorisations peuvent être exprimées sous forme de politiques.

Application des politiques

L'application des politiques est le processus qui consiste à appliquer la décision d'évaluation au sein de l'application en dehors des autorisations vérifiées. Si l'évaluation des autorisations vérifiées aboutit à un refus, l'application empêchera le principal d'accéder à la ressource.

Boutique Policy

Un magasin de politiques est un conteneur pour les politiques et les modèles. Chaque magasin contient un schéma qui est utilisé pour valider les politiques ajoutées au magasin. Par défaut, chaque application possède son propre magasin de politiques, mais plusieurs applications peuvent partager un seul magasin de politiques. Lorsqu'une application fait une demande d'autorisation, elle identifie le magasin de politiques utilisé pour évaluer cette demande. Les magasins de politiques permettent d'isoler un ensemble de politiques et peuvent donc être utilisés dans une application multi-locataires pour contenir les schémas et les politiques de chaque locataire. Une seule application peut disposer de magasins de politiques distincts pour chaque locataire.

Lors de l'évaluation d'une demande d'autorisation, Verified Permissions ne prend en compte que le sous-ensemble des politiques du magasin de politiques qui sont pertinentes pour la demande. La pertinence est déterminée en fonction de la portée de la politique. Le champ d'application identifie le principal et la ressource spécifiques auxquels la politique s'applique, ainsi que les actions que le principal peut effectuer sur la ressource. La définition du périmètre permet d'améliorer les performances en réduisant l'ensemble des politiques envisagées.

Politiques satisfaites

Les politiques satisfaisantes sont celles qui correspondent aux paramètres de la demande d'autorisation.

Différences entre les autorisations vérifiées et Cedar

Amazon Verified Permissions utilise le moteur de langage de politique Cedar pour effectuer ses tâches d'autorisation. Cependant, il existe certaines différences entre l'implémentation native de Cedar et l'implémentation de Cedar trouvée dans les autorisations vérifiées. Cette rubrique met en évidence ces différences.

Définition de l'espace de noms

L'implémentation des autorisations vérifiées de Cedar présente les différences suivantes par rapport à l'implémentation native de Cedar :

- Les autorisations vérifiées ne prennent en charge qu'une seuleespace de noms dans un schémadéfini dans un magasin de politiques.
- Les autorisations vérifiées ne vous permettent pas de créeresespace de nomsavec les valeurs suivantes :`aws`,`amazon`, ou`cedar`.

Support pour les modèles de politique

Verified Permissions et Cedar autorisent tous deux des espaces réservés uniquement aux`principal`et`resource`. Toutefois, les autorisations vérifiées exigent également que ni le`principal`et`resource`ne sont pas soumis à des contraintes.

La politique suivante est valide dans Cedar mais elle est rejetée par Verified Permissions car`principal`n'est pas contraint.

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

Les deux exemples suivants sont valides à la fois dans Cedar et dans Verified Permissions car`principal`et`resource`ont des contraintes.

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

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

Support de schéma

Les autorisations vérifiées nécessitent que tous les noms de clé JSON du schéma soient des chaînes non vides. Cedar autorise les chaînes vides dans certains cas, par exemple pour les propriétés.

Support de type d'extension

Les autorisations vérifiées sont compatibles avec Cedartypes de rallongesdans les politiques, mais ne permet pas actuellement de les inclure dans la définition d'un schéma ou dans le cadre du`entities`paramètre du`IsAuthorized`et`IsAuthorizedWithToken`opérations.

Les types d'extension incluent le point fixe (`decimal`) et adresse IP (`ipaddr`) types de données.

Format Cedar JSON pour les entités

À l'heure actuelle, les autorisations vérifiées vous obligent à transmettre la liste des entités à prendre en compte dans le cadre d'une demande d'autorisation en utilisant la structure définie pourEntitiesDefinition, qui est un tableau deEntityIteméléments. Verified Permissions ne prend actuellement pas en charge la transmission de la liste des entités à prendre en compte dans une demande d'autorisation dansformat Cedar JSON. Pour connaître les exigences spécifiques relatives au formatage de vos entités à utiliser dans les autorisations vérifiées, voirFormatage des entités dans Amazon Verified Permissions.

Définition des groupes d'action

Les méthodes d'autorisation de Cedar nécessitent une liste des entités à prendre en compte lors de l'évaluation d'une demande d'autorisation par rapport aux politiques.

Vous pouvez définir les actions et les groupes d'actions utilisés par votre application dans le schéma. Toutefois, Cedar n'inclut pas le schéma dans le cadre d'une demande d'évaluation. Cedar utilise plutôt le schéma uniquement pour valider les politiques et les modèles de politiques que vous soumettez. Comme Cedar ne fait pas référence au schéma lors des demandes d'évaluation, même si vous avez défini des groupes d'actions dans le schéma, vous devez également inclure la liste de tous les groupes d'actions dans la liste des entités que vous devez transmettre aux opérations de l'API d'autorisation.

Verified Permissions le fait pour vous. Tous les groupes d'actions que vous définissez dans votre schéma sont automatiquement ajoutés à la liste des entités que vous transmettez en tant que paramètre au`IsAuthorized`ou`IsAuthorizedWithToken`opérations.

Limites de longueur et de taille

Verified Permissions prend en charge le stockage sous forme de magasins de politiques pour stocker vos schémas, politiques et modèles de politiques. Ce stockage oblige les autorisations vérifiées à imposer des limites de longueur et de taille qui ne sont pas pertinentes pour Cedar.

Objet	Limite d'autorisations vérifiées (en octets)	Limite de cèdre
Taille de la politique[1]	10 000	Aucune

Objet	Limite d'autorisations vérifiées (en octets)	Limite de cèdre
Description de la politique en ligne	150	ne s'applique pas au cèdre
Taille du modèle de politique	10 000	Aucune
Taille du schéma	10 000	Aucune
Type d'entité	200	Aucune
ID de stratégie	64	Aucune
ID de modèle de politique	64	Aucune
ID de l'entité	200	Aucune
ID du magasin Policy	64	ne s'applique pas au cèdre

[1] Il existe une limite de politiques par magasin de politiques dans les autorisations vérifiées en fonction de la taille combinée des principes, des actions et des ressources des politiques créées dans le magasin de politiques. La taille totale de toutes les politiques relatives à une seule ressource ne peut pas dépasser 200 000 octets. Pour les politiques liées à un modèle, la taille du modèle de stratégie n'est comptée qu'une seule fois, plus la taille de chaque ensemble de paramètres utilisé pour instancier chaque politique liée au modèle.

Commencer avec les autorisations vérifiées

Utilisez ce didacticiel pour démarrer avec Amazon Verified Permissions.

Rubriques

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique en matière de sécurité consiste à attribuer un accès administratif à un utilisateur et à n'utiliser que l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. Vous pouvez afficher l'activité en cours de votre compte et gérer votre compte à tout moment en accédant à https://aws.amazon.com/ et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'IAM utilisateur.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, accordez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connectez-vous en tant qu'utilisateur disposant d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribuer l'accès à des utilisateurs supplémentaires

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme aux meilleures pratiques en matière d'application des autorisations du moindre privilège.

   Pour obtenir des instructions, voir Création d'un ensemble d'autorisations dans le guide de AWS IAM Identity Center l'utilisateur.
2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

   Pour obtenir des instructions, consultez la section Ajouter des groupes dans le guide de AWS IAM Identity Center l'utilisateur.

IAM politiques relatives aux autorisations vérifiées

Verified Permissions gère les autorisations des utilisateurs au sein de votre application. Pour que votre application puisse appeler les API d'autorisations vérifiées ou pour que AWS Management Console les utilisateurs soient autorisés à gérer les politiques Cedar dans un magasin de politiques d'autorisations vérifiées, vous devez ajouter les IAM autorisations nécessaires.

Les politiques basées sur l'identité sont des documents de politique d'autorisation JSON que vous pouvez associer à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, consultez la section Création de IAM politiques dans le Guide de l' IAM utilisateur.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier les actions et les ressources autorisées ou refusées ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées (répertoriées ci-dessous). Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour en savoir plus sur tous les éléments que vous pouvez utiliser dans une politique JSON, consultez la référence des éléments de stratégie IAM JSON dans le guide de IAM l'utilisateur.

Action	Description
CreatePolicyStore	Action visant à créer un nouveau magasin de politiques.
DeletePolicyStore	Action visant à supprimer un magasin de politiques.
ListPolicyStores	Action visant à répertorier tous les magasins de politiques dans le Compte AWS.
CreatePolicy	Action visant à créer une politique Cedar dans un magasin de politiques. Vous pouvez créer une politique statique ou une politique liée à un modèle de stratégie.
DeletePolicy	Action visant à supprimer une politique d'un magasin de politiques.
GetPolicy	Action permettant de récupérer des informati ons relatives à une politique spécifiée.
ListPolicies	Action permettant de répertorier toutes les politiques d'un magasin de politiques.
IsAuthorized	Action permettant d'obtenir une réponse d'autorisation en fonction des paramètres décrits dans la demande d'autorisation.

Exemple IAM de politique d'autorisation pour l' CreatePolicy action :

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

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

Créez votre premier magasin de politiques d'autorisations vérifiées

Lorsque vous vous connectez à la console Verified Permissions pour la première fois, vous pouvez choisir comment créer votre premier magasin de politiques et votre première politique Cedar. Suivez la procédure de connexion correspondant à votre type d'utilisateur, comme décrit dans la rubrique Connexion à AWS du Guide de l'utilisateur Connexion àAWS . Sur la page d'accueil de la console, sélectionnez le service Amazon Verified Permissions. Choisissez Démarrer.

Création d'un exemple de magasin de politiques

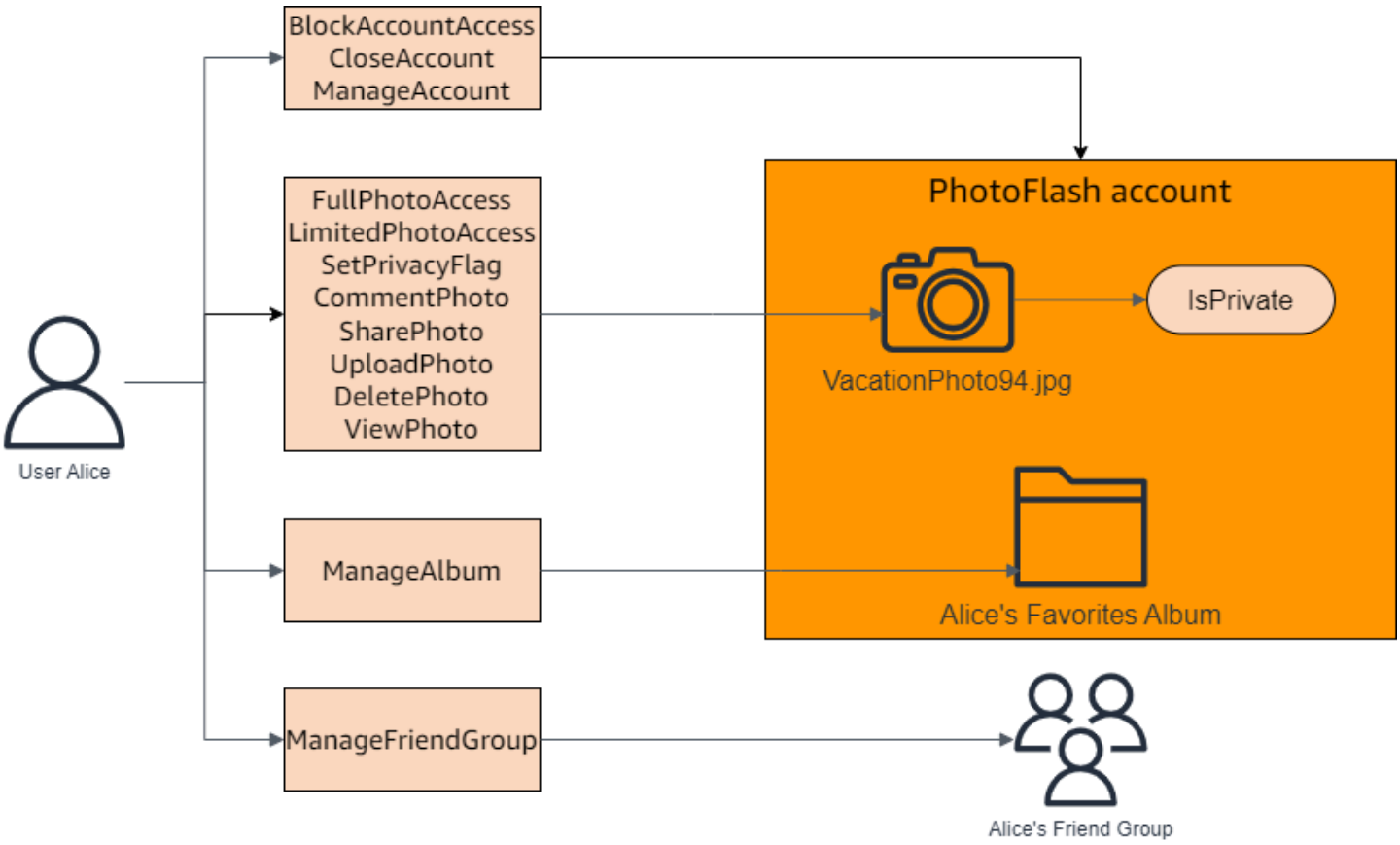
Si c'est la première fois que vous utilisez les autorisations vérifiées, nous vous recommandons d'utiliser l'un des exemples de magasins de politiques pour vous familiariser avec le fonctionnement des autorisations vérifiées. Les exemples de magasins de politiques fournissent des politiques prédéfinies et un schéma.

Pour créer un magasin de politiques à l'aide de la méthode de configuration Sample Policy Store

1. Dans la console des autorisations vérifiées, sélectionnez Créer un nouveau magasin de politiques.
2. Dans la section Options de démarrage, sélectionnez Sample policy store.
3. Dans la section Exemple de projet, choisissez le type d'exemple d'application d'autorisations vérifiées à utiliser. Pour ce didacticiel, choisissez le magasin PhotoFlashde politiques.
4. Un espace de noms pour le schéma de votre exemple de magasin de politiques est automatiquement généré en fonction de l'exemple de projet que vous avez choisi.
5. Choisissez Create Policy Store.

   Votre magasin de politiques est créé avec des politiques, des modèles de politiques et un schéma pour l'exemple de magasin de politiques.

Le schéma ci-dessous illustre les relations entre les PhotoFlash exemples d'actions du magasin de politiques et les types de ressources auxquels elles s'appliquent.



Création de politiques liées à un modèle pour un exemple de magasin de politiques

L' PhotoFlash exemple de magasin de politiques inclut des politiques, des modèles de politiques et un schéma. Vous pouvez créer des politiques liées à des modèles sur la base des modèles de politiques inclus dans l'exemple de magasin de politiques.

Pour créer des politiques liées à un modèle pour le magasin d'exemples de politiques

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le panneau de navigation de gauche, sélectionnez Policies (Politiques).
3. Choisissez Créer une politique, puis choisissez Créer une politique liée à un modèle.
4. Cliquez sur le bouton radio à côté du modèle de politique avec la description Accorder un accès complet aux photos partagées non privées, puis cliquez sur Suivant.
5. Pour Principal, entrez`PhotoFlash::User::"Alice"`. Pour Ressource, entrez`PhotoFlash::Album::"Bob-Vacation-Album"`.

6. Choisissez Créer une politique liée à un modèle.

   La nouvelle politique liée au modèle est affichée sous Politiques.

7. Créez une autre politique liée à un modèle pour le magasin d' PhotoFlash exemples de politiques. Choisissez Créer une politique, puis choisissez Créer une politique liée à un modèle.

8. Cliquez sur le bouton radio à côté du modèle de politique avec la description Accorder un accès limité aux photos partagées non privées, puis cliquez sur Suivant.

9. Pour Principal, entrez`PhotoFlash::FriendGroup::"MySchoolFriends"`. Pour Ressource, entrez`PhotoFlash::Album::"Alice's favorite album"`.

10. Choisissez Créer une politique liée à un modèle.

    La nouvelle politique liée au modèle est affichée sous Politiques.

Nous testerons les nouvelles politiques liées aux modèles dans la section suivante du didacticiel. Pour plus d'exemples de valeurs que vous pouvez utiliser pour créer une politique liée à un modèle, consultez. PhotoFlash PhotoFlashexemples de politiques liés à un modèle

Tester un exemple de magasin de politiques

Après avoir créé votre banque d'exemples de politiques et vos politiques liées à des modèles, vous pouvez tester les exemples de politiques statiques d'autorisations vérifiées et vos nouvelles politiques liées à des modèles en exécutant une demande d'autorisation simulée à l'aide du banc de test des autorisations vérifiées.

Selon la date à laquelle vous avez créé votre exemple de magasin de politiques, vos modèles de politiques peuvent différer des références de cette procédure. Avant de commencer cette partie du didacticiel, vérifiez que vous disposez de chaque modèle de stratégie qui suit dans votre PhotoFlash exemple de magasin de politiques. Si votre politique ne correspond pas à ces politiques, modifiez les politiques existantes ou créez un nouveau magasin de politiques à partir de l'option Exemple de projet PhotoFlash.

Accorder un accès complet aux photos partagées non privées

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

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

Accorder un accès limité aux photos partagées non privées

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

Pour tester des exemples de politiques de magasin de politiques

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Test bench.
3. Choisissez le mode visuel.
4. Dans la section Principal, choisissez PhotoFlash: :User parmi les principaux types de votre schéma. Entrez un identifiant pour l'utilisateur dans la zone de texte. Par exemple, `Alice`.
5. Ne choisissez pas Ajouter un parent pour le directeur.
6. Pour l'attribut Account : Entity, assurez-vous que l'entité PhotoFlash::Account est sélectionnée. Entrez un identifiant pour le compte. Par exemple, `Alice-account`.
7. Dans la section Ressource, choisissez le type de ressource PhotoFlash::Photo. Entrez un identifiant pour la photo dans la zone de texte. Par exemple, `photo.jpeg`.
8. Choisissez Ajouter un parent et choisissez PhotoFlash: :Account pour le type d'entité. Entrez le même identifiant pour le compte parent associé à la photo que vous avez indiqué dans le champ Compte : Entité pour l'utilisateur. Par exemple, `Alice-account`.
9. Dans la section Action, choisissez PhotoFlash: :Action : : » ViewPhoto "dans la liste des actions valides.
10. Dans la section Entités supplémentaires, choisissez Ajouter cette entité pour ajouter l'entité de compte suggérée.
11. Choisissez Exécuter une demande d'autorisation en haut de la page pour simuler la demande d'autorisation pour les politiques Cedar dans l'exemple de magasin de politiques. Le banc de test doit afficher la décision d'autoriser la demande.

Le tableau suivant fournit des valeurs supplémentaires pour le principal, la ressource et l'action que vous pouvez tester avec le banc de test des autorisations vérifiées. Le tableau inclut la décision de demande d'autorisation basée sur les politiques statiques incluses dans l' PhotoFlash exemple de magasin de politiques et les politiques liées à un modèle que vous avez créées dans la section précédente.

Valeur principale	Compte principal : valeur de l'entité	Valeur des ressources	Valeur parente de la ressource	Action	Décision d'autorisation
PhotoFlash: :Utilisateur | Alice	PhotoFlash: :Compte | Compte Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Compte | Compte Bob	PhotoFlash: :Action : : » » ViewPhoto	Refuser
PhotoFlash: :Utilisateur | Alice	PhotoFlash: :Compte | Compte Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Compte | Compte Alice	PhotoFlash: :Action : : » » ViewPhoto	Autorisation
PhotoFlash: :Utilisateur | Alice	PhotoFlash: :Compte | Compte Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash: :Album | Bob Vacation-Album	PhotoFlash: :Action : : » » ViewPhoto	Autorisation
PhotoFlash: :Utilisateur | Alice	PhotoFlash: :Compte | Compte Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash: :Album | Bob Vacation-Album	PhotoFlash: :Action : : » » DeletePhoto	Refuser
PhotoFlash: :Utilisateur | Alice	PhotoFlash: :Compte | Compte Alice	PhotoFlash: :Photo | Bob-photo .jpeg, IsPrivate : Boolean | true	PhotoFlash: :Album | Bob Vacation-Album	PhotoFlash: :Action : : » » ViewPhoto	Refuser

Valeur principale	Compte principal : valeur de l'entité	Valeur des ressources	Valeur parente de la ressource	Action	Décision d'autorisation
PhotoFlash: :Utilisateur | Jane, PhotoFlash: : FriendGroup | MySchoolFriends	PhotoFlash: :Compte | Compte Jane	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | L'album préféré d'Alice	PhotoFlash: :Action : : » » ViewPhoto	Autorisation
PhotoFlash: :Utilisateur | Jane, PhotoFlash: : FriendGroup | MySchoolFriends	PhotoFlash: :Compte | Compte Jane	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | L'album préféré d'Alice	PhotoFlash: :Action : : » » DeletePhoto	Refuser

Créez un magasin de politiques avec une API et un fournisseur d'identité connectés

Un cas d'utilisation courant des autorisations Amazon Verified consiste à autoriser les demandes d'un client d'application à une API principale. AWS dispose d'un service d'authentification des utilisateurs de l'application : Amazon Cognito. AWS dispose également d'un service pour les API hébergées sécurisées : Amazon API Gateway. Lorsque vous associez un magasin de politiques d'autorisations vérifiées à ces deux Services AWS options, vous pouvez associer l'authentification du groupe d'utilisateurs et l'autorisation d'API dans votre application grâce à un ensemble de politiques cohérent et centralisé. Les magasins de politiques d'autorisations vérifiées prennent en charge de manière intégrée les sources d'identité du pool d'utilisateurs Amazon Cognito et les API API Gateway.

Pour créer un magasin de politiques lié à un groupe d'utilisateurs et à une API existants, choisissez Configurer avec Cognito et API Gateway lorsque vous créez un nouveau magasin de politiques.

Un magasin de politiques lié à une API provisionne automatiquement votre modèle d'autorisation et vos ressources pour les demandes d'autorisation. Le processus de création de Set up with Cognito et API Gateway génère un magasin de politiques avec une source d'identité pour le groupe d'utilisateurs et un autorisateur Lambda qui connecte API Gateway aux autorisations vérifiées. Dans un premier temps, vous pouvez autoriser les demandes d'API en fonction de l'appartenance aux groupes des utilisateurs. Par exemple, les autorisations vérifiées ne peuvent accorder l'accès qu'aux utilisateurs membres du `Directors` groupe.

Au fur et à mesure que votre application se développe, vous pouvez implémenter une autorisation précise avec des attributs utilisateur et des étendues OAuth 2.0. Par exemple, les autorisations vérifiées ne peuvent accorder l'accès qu'aux utilisateurs possédant un `email` attribut dans le domaine`mycompany.co.uk`.

Une fois que vous avez automatisé le modèle d'autorisation pour votre API, il ne vous reste plus qu'à authentifier les utilisateurs, à générer des demandes d'API dans votre application et à gérer votre magasin de politiques.

Pour en savoir plus, consultez Magasins de politiques liés à l'API.

Boutiques relatives à la politique d'autorisations vérifiées d'Amazon

Un magasin de politiques est un conteneur pour les politiques et les modèles de politiques. Chaque magasin de politiques contient un schéma qui est utilisé pour valider les politiques ajoutées au magasin de politiques. Nous recommandons de créer un magasin de politiques par application ou un magasin de politiques par locataire pour les applications à locataires multiples. Vous devez spécifier un magasin de politiques lorsque vous faites une demande d'autorisation.

Nous vous recommandons d'utiliser des espaces de noms pour les entités Cedar dans vos magasins de polices afin d'éviter toute ambiguïté. Un espace de noms est un préfixe de chaîne pour un type, séparé par une paire de deux-points (`::`) en tant que délimiteur. Verified Permissions prend en charge un espace de noms par magasin de politiques. Pour plus d'informations, voir Namespaces dans le Guide de référence du langage politique de Cedar.

Rubriques

Création de magasins de politiques d'autorisations vérifiées

Vous pouvez créer un magasin de règles en utilisant les méthodes suivantes :

- Suivez une configuration guidée : vous allez définir un type de ressource avec des actions valides et un type principal avant de créer votre première politique.
- Configuration avec API Gateway et une source d'identité : définissez vos entités principales avec les utilisateurs qui se connectent avec un fournisseur d'identité (IdP), et vos actions et entités de ressources à partir d'une API Amazon API Gateway. Nous recommandons cette option si vous souhaitez que votre application autorise les demandes d'API impliquant l'appartenance à un groupe d'utilisateurs.
- Commencez par un exemple de magasin de politiques : choisissez un exemple de magasin de politiques de projet prédéfini. Nous recommandons cette option si vous souhaitez en savoir plus sur les autorisations vérifiées et si vous souhaitez consulter et tester des exemples de politiques.

- Créez un magasin de politiques vide : vous définirez vous-même le schéma et toutes les politiques d'accès. Nous recommandons cette option si vous êtes déjà familiarisé avec la configuration d'un magasin de politiques.

Guided setup

Pour créer un magasin de politiques à l'aide de la méthode de configuration guidée

L'assistant de configuration guidée vous guide tout au long du processus de création de la première itération de votre magasin de politiques. Vous allez créer un schéma pour votre premier type de ressource, décrire les actions applicables à ce type de ressource et le type principal pour lequel vous accordez des autorisations. Vous allez ensuite créer votre première politique. Une fois que vous aurez terminé cet assistant, vous pourrez ajouter des éléments à votre magasin de politiques, étendre le schéma pour décrire d'autres types de ressources et de principaux types, et créer des politiques et des modèles supplémentaires.

1. Dans la console des autorisations vérifiées, sélectionnez Créer un nouveau magasin de politiques.
2. Dans la section Options de démarrage, choisissez Configuration guidée.
3. Entrez une description du Policy Store. Ce texte peut être celui qui convient à votre organisation comme référence conviviale au fonctionnement du magasin de politiques actuel, par exemple les mises à jour météorologiques.
4. Dans la section Détails, saisissez un espace de noms pour votre schéma.
5. Choisissez Suivant.
6. Dans la fenêtre Type de ressource, saisissez le nom de votre type de ressource.
7. (Facultatif) Choisissez Ajouter un attribut pour ajouter des attributs de ressource. Tapez le nom de l'attribut et choisissez un type d'attribut pour chaque attribut de la ressource. Choisissez si chaque attribut est obligatoire. Verified Permissions utilise les valeurs d'attribut spécifiées lors de la vérification des politiques par rapport au schéma. Pour supprimer un attribut qui a été ajouté pour le type de ressource, choisissez Supprimer à côté de l'attribut.
8. Dans le champ Actions, saisissez les actions à autoriser pour le type de ressource spécifié. Pour ajouter des actions supplémentaires pour le type de ressource, choisissez Ajouter une action. Pour supprimer une action qui a été ajoutée pour le type de ressource, choisissez Supprimer à côté de l'action.
9. Dans le champ Nom du type principal, tapez le nom d'un type de principal qui utilisera les actions spécifiées pour votre type de ressource.

10. Choisissez Suivant.
11. Dans la fenêtre Type principal, choisissez la source d'identité pour votre type principal.

    - Choisissez Personnalisé si l'identifiant et les attributs du principal seront fournis directement par votre application d'autorisations vérifiées. Choisissez Ajouter un attribut pour ajouter des attributs principaux. Tapez le nom de l'attribut et choisissez un type d'attribut pour chaque attribut du principal. Verified Permissions utilise les valeurs d'attribut spécifiées lors de la vérification des politiques par rapport au schéma. Pour supprimer un attribut qui a été ajouté pour le type principal, choisissez Supprimer à côté de l'attribut.
    - Choisissez le groupe d'utilisateurs Cognito si l'identifiant et les attributs du principal seront fournis à partir d'un identifiant ou d'un jeton d'accès généré par Amazon Cognito. Choisissez Connect user pool. Sélectionnez Région AWSet saisissez l'ID du groupe d'utilisateurs Amazon Cognito auquel vous souhaitez vous connecter. Choisissez Se connecter. Pour plus d'informations, consultez la section Autorisation avec autorisations vérifiées par Amazon dans le guide du développeur Amazon Cognito.
12. Choisissez Suivant.
13. Dans la section Détails du contrat, saisissez une description facultative du contrat pour votre premier contrat Cedar.
14. Dans le champ Champ d'application des principes, choisissez les principaux auxquels la politique accordera des autorisations.

    - Choisissez Spécific principal pour appliquer la politique à un principal spécifique. Choisissez le principal dans le champ Principal qui sera autorisé à prendre des mesures et saisissez un identifiant d'entité pour le principal.
    - Choisissez Tous les principaux pour appliquer la politique à tous les principaux de votre magasin de polices.
15. Dans le champ Champ d'application des ressources, choisissez les ressources sur lesquelles les principaux spécifiés seront autorisés à agir.

    - Choisissez Ressource spécifique pour appliquer la politique à une ressource spécifique. Choisissez la ressource dans le champ Ressource à laquelle cette politique doit s'appliquer et saisissez un identifiant d'entité pour la ressource.
    - Choisissez Toutes les ressources pour appliquer la politique à toutes les ressources de votre magasin de politiques.
16. Dans le champ Champ d'application des actions, choisissez les actions que les principaux spécifiés seront autorisés à effectuer.

- Choisissez un ensemble d'actions spécifique pour appliquer la politique à des actions spécifiques. Cochez les cases à côté des actions dans le champ Actions auxquelles cette politique doit s'appliquer.
- Choisissez Toutes les actions pour appliquer la politique à toutes les actions de votre magasin de politiques.

17. Consultez la politique dans la section Aperçu de la politique. Choisissez Create Policy Store.

Set up with API Gateway and an identity source

Pour créer un magasin de politiques à l'aide de la méthode Set up with API Gateway et d'une source d'identité

L'option API Gateway sécurise les API grâce à des politiques d'autorisations vérifiées conçues pour prendre des décisions d'autorisation à partir de groupes ou de rôles d'utilisateurs. Cette option crée un magasin de politiques pour tester l'autorisation avec des groupes de sources d'identité et une API avec un autorisateur Lambda.

Les utilisateurs et leurs groupes dans un IdP deviennent soit vos principaux (jetons d'identification), soit votre contexte (jetons d'accès). Les méthodes et les chemins d'une API API Gateway deviennent les actions autorisées par vos politiques. Votre application devient la ressource. À la suite de ce flux de travail, Verified Permissions crée un magasin de politiques, une fonction Lambda et un autorisateur Lambda d'API. Vous devez attribuer l'autorisateur Lambda à votre API une fois ce flux de travail terminé.

1. Dans la console des autorisations vérifiées, sélectionnez Créer un nouveau magasin de politiques.
2. Dans la section Options de démarrage, choisissez Set up with API Gateway and an identity source, puis sélectionnez Next.
3. À l'étape Importer des ressources et des actions, sous API, choisissez une API qui servira de modèle aux ressources et aux actions de votre magasin de politiques.

    a. Choisissez une étape de déploiement parmi les étapes configurées dans votre API et sélectionnez Importer l'API. Pour plus d'informations sur les étapes d'API, consultez la section Configuration d'une étape pour une API REST dans le manuel Amazon API Gateway Developer Guide.

    b. Prévisualisez votre carte des ressources et des actions importées.

c. Pour mettre à jour les ressources ou les actions, modifiez vos chemins ou méthodes d'API et sélectionnez Importer l'API.

d. Lorsque vous êtes satisfait de vos choix, choisissez Next.

4. Dans Source d'identité, choisissez un type de fournisseur d'identité. Vous pouvez choisir un groupe d'utilisateurs Amazon Cognito ou un type d'IdP OpenID Connect (OIDC).

5. Si vous avez choisi Amazon Cognito :

   a. Choisissez un groupe d'utilisateurs identique à celui Région AWS de Compte AWS votre magasin de polices.

   b. Choisissez le type de jeton à transmettre à l'API que vous souhaitez soumettre pour autorisation. L'un ou l'autre type de jeton contient des groupes d'utilisateurs, qui constituent la base de ce modèle d'autorisation lié à l'API.

   c. Dans le cadre de la validation du client d'application, vous pouvez limiter l'étendue d'un magasin de politiques à un sous-ensemble des clients de l'application Amazon Cognito d'un groupe d'utilisateurs multi-locataires. Pour demander à l'utilisateur de s'authentifier auprès d'un ou de plusieurs clients d'applications spécifiques de votre groupe d'utilisateurs, sélectionnez Accepter uniquement les jetons dont les identifiants de client d'application sont attendus. Pour accepter tout utilisateur qui s'authentifie auprès du groupe d'utilisateurs, sélectionnez Ne pas valider les identifiants des clients de l'application.

   d. Choisissez Suivant.

6. Si vous avez choisi le fournisseur OIDC :

   a. Dans URL de l'émetteur, entrez l'URL de votre émetteur OIDC. Il s'agit du point de terminaison du service qui fournit le serveur d'autorisation, les clés de signature et d'autres informations sur votre fournisseur, par exemple`https://auth.example.com`. L'URL de votre émetteur doit héberger un document de découverte OIDC à l'adresse. `/.well-known/openid-configuration`

   b. Dans Type de jeton, choisissez le type de JWT OIDC que vous souhaitez que votre demande soumette pour autorisation. Pour plus d'informations, consultez Utilisation des sources d'identité dans les schémas et les politiques.

   c. Dans Token claims, choisissez la manière dont vous souhaitez configurer les attributs utilisateur dans votre magasin de polices. Ces attributs définissent les demandes auxquelles vos politiques peuvent faire référence.

i. Choisissez une source de réclamation.

A. Pour fournir un exemple de jeton, choisissez Extraire de la charge utile JWT et collez la charge utile d'un JWT du type de jeton que vous avez choisi. Les JWT contiennent un en-tête, une charge utile et une signature. Votre échantillon JWT doit être décodé et réservé à la charge utile. Pour analyser la charge utile, sélectionnez Extraire.

B. Pour saisir votre propre ensemble d'attributs, choisissez Saisir les demandes manuellement.

ii. Entrez ou confirmez le nom de chaque demande de jeton et le type de valeur de réclamation que vous souhaitez ajouter aux attributs du principal utilisateur ou au contexte d'action de votre schéma.

d. Dans Réclamations d'utilisateur et de groupe, choisissez une réclamation d'utilisateur pour la source d'identité. Il s'agit généralement `sub` d'une réclamation provenant de votre identifiant ou de votre jeton d'accès contenant l'identifiant unique de l'entité à évaluer. Les identités de l'IdP OIDC connecté seront mappées au type d'utilisateur dans votre magasin de politiques.

e. Dans Réclamations d'utilisateurs et de groupes, choisissez une réclamation de groupe pour la source d'identité. Il s'agit généralement `groups` d'une réclamation provenant de votre identifiant ou de votre jeton d'accès qui contient une liste des groupes d'utilisateurs. Votre magasin de polices autorisera les demandes en fonction de l'appartenance au groupe.

f. Dans Validation d'audience ou ID client, entrez les identifiants clients ou les URL d'audience que vous souhaitez que votre magasin de politiques accepte dans les demandes d'autorisation, le cas échéant. Pour les jetons d'accès, entrez une valeur de réclamation d'audience telle que`https://myapp.example.com`. Pour les jetons d'identification, entrez un identifiant client tel que`1example23456789`.

g. Choisissez Suivant.

7. Si vous avez choisi Amazon Cognito, Verified Permissions interroge votre groupe d'utilisateurs pour trouver des groupes. Pour les fournisseurs OIDC, entrez les noms des groupes manuellement. L'étape Attribuer des actions aux groupes crée des politiques pour votre magasin de politiques qui permettent aux membres du groupe d'effectuer des actions.

a. Choisissez ou ajoutez les groupes que vous souhaitez inclure dans vos politiques.

b. Attribuez des actions à chacun des groupes que vous avez sélectionnés.

c. Choisissez Suivant.

8. Dans Déployer l'intégration de l'application, passez en revue les étapes que Verified Permissions effectuera pour créer votre magasin de politiques et votre autorisateur Lambda.
9. Lorsque vous êtes prêt à créer les nouvelles ressources, choisissez Créer et déployer.
10. Gardez l'étape d'état du magasin Policy ouverte dans votre navigateur pour suivre la progression de la création des ressources par le biais d'autorisations vérifiées.
11. Après un certain temps, généralement environ une heure, ou lorsque l'étape Déployer l'autorisateur Lambda indique Success, configurez votre autorisateur.

    Les autorisations vérifiées auront créé une fonction Lambda et un autorisateur Lambda dans votre API. Choisissez Open API pour accéder à votre API.

    Pour savoir comment attribuer un autorisateur Lambda, consultez la section Utiliser les autorisateurs Lambda d'API Gateway dans le manuel du développeur Amazon API Gateway.

    a. Accédez à Autorisateurs pour votre API et notez le nom de l'autorisateur créé par Verified Permissions.
    b. Accédez à Ressources et sélectionnez une méthode de haut niveau dans votre API.
    c. Sélectionnez Modifier dans les paramètres de demande de méthode.
    d. Définissez le nom de l'autorisateur comme indiqué précédemment.
    e. Développez les en-têtes de requête HTTP, entrez un nom ou`AUTHORIZATION`, puis sélectionnez Obligatoire.
    f. Déployez l'étape de l'API.
    g. Enregistrez vos modifications.
12. Testez votre autorisateur avec un jeton de groupe d'utilisateurs du type de jeton que vous avez sélectionné à l'étape Choisir une source d'identité. Pour plus d'informations sur la connexion au groupe d'utilisateurs et la récupération de jetons, consultez le flux d'authentification du groupe d'utilisateurs dans le manuel Amazon Cognito Developer Guide.
13. Testez à nouveau l'authentification avec un jeton de pool d'utilisateurs dans l'`AUTHORIZATION`en-tête d'une demande adressée à votre API.
14. Examinez votre nouveau magasin de polices. Ajoutez et affinez des politiques.

Sample policy store

Pour créer un magasin de politiques à l'aide de la méthode de configuration Sample Policy Store

1. Dans la section Options de démarrage, sélectionnez Sample policy store.
2. Dans la section Exemple de projet, choisissez le type d'exemple d'application d'autorisations vérifiées à utiliser.

   - PhotoFlashest un exemple d'application Web destinée aux clients qui permet aux utilisateurs de partager des photos et des albums individuels avec des amis. Les utilisateurs peuvent définir des autorisations précises sur les personnes autorisées à voir, à commenter et à partager à nouveau leurs photos. Les titulaires de comptes peuvent également créer des groupes d'amis et organiser les photos dans des albums.
   - DigitalPetStore est un exemple d'application où tout le monde peut s'inscrire et devenir client. Les clients peuvent ajouter des animaux de compagnie à vendre, rechercher des animaux de compagnie et passer des commandes. Les clients qui ont ajouté un animal de compagnie sont enregistrés en tant que propriétaire de l'animal. Les propriétaires d'animaux peuvent mettre à jour les informations de leur animal, télécharger une photo de l'animal ou supprimer la liste des animaux. Les clients qui ont passé une commande sont enregistrés en tant que propriétaires de la commande. Les propriétaires de la commande peuvent obtenir des informations sur la commande ou l'annuler. Les gérants des animaleries ont un accès administratif.

     > **Note**
     >
     > Le DigitalPetmagasin d'exemples de politiques Store n'inclut pas de modèles de politiques. Les magasins de politiques PhotoFlashet TinyTodod'exemples incluent des modèles de politiques.

   - TinyTodoest un exemple d'application qui permet aux utilisateurs de créer des tâches et des listes de tâches. Les propriétaires de listes peuvent gérer et partager leurs listes et spécifier qui peut consulter ou modifier leurs listes.
3. Un espace de noms pour le schéma de votre exemple de magasin de politiques est automatiquement généré en fonction de l'exemple de projet que vous avez choisi.
4. Choisissez Create Policy Store.

Votre magasin de politiques est créé avec des politiques et un schéma pour l'exemple de magasin de politiques que vous avez choisi. Pour plus d'informations sur les politiques liées à des modèles que vous pouvez créer pour les exemples de magasins de politiques, consultez. Exemples de politiques liées à un modèle pour les autorisations vérifiées, exemples de magasins de politiques

Empty policy store

Pour créer un magasin de politiques à l'aide de la méthode de configuration Empty policy store

1. Dans la section Options de démarrage, choisissez Empty policy store.
2. Choisissez Create Policy Store.

Un magasin de politiques vide est créé sans schéma, ce qui signifie que les politiques ne sont pas validées. Pour plus d'informations sur la mise à jour du schéma de votre magasin de politiques, consultezSchéma de la boutique Amazon Verified Permissions Policy.

Pour plus d'informations sur la création de politiques pour votre magasin de politiques, consultez Création de politiques statiques relatives aux autorisations vérifiées par Amazon etCréation de politiques liées à un modèle.

AWS CLI

Pour créer un magasin de politiques vide à l'aide du AWS CLI.

Vous pouvez créer un magasin de politiques à l'aide de cette `create-policy-store` opération.

> **Note**
>
> Un magasin de politiques que vous créez à l'aide du AWS CLI est vide.
>
> - Pour ajouter un schéma, voirSchéma de la boutique Amazon Verified Permissions Policy.
> - Pour ajouter des politiques, consultezCréation de politiques statiques relatives aux autorisations vérifiées par Amazon.
> - Pour ajouter des modèles de politique, consultezCréation de modèles de politiques.

```
$ 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

Vous pouvez créer un magasin de politiques à l'aide de l'`CreatePolicyStore`API. Pour plus d'informations, consultez CreatePolicyStore dans le guide de référence de l'API Amazon Verified Permissions.

Magasins de politiques liés à l'API

Lorsque vous créez un nouveau magasin de politiques dans la console Amazon Verified Permissions, vous pouvez choisir l'option Configurer avec API Gateway et une source d'identité. Avec cette option, vous créez un magasin de politiques lié à une API, un modèle d'autorisation pour les applications qui s'authentifient auprès des groupes d'utilisateurs Amazon Cognito ou auprès d'un fournisseur d'identité OIDC (IdP) et obtiennent des données à partir des API Amazon API Gateway. Consultez Créez un magasin de politiques avec une API et un fournisseur d'identité connectés pour démarrer.

Rubriques

- Comment les autorisations vérifiées autorisent-elles les demandes d'API
- Ajout d'un contrôle d'accès basé sur les attributs (ABAC)
- Considérations relatives aux magasins de politiques liés aux API
- Résolution des problèmes liés aux magasins de politiques liés à l'API

> **Important**
>
> Les magasins de politiques que vous créez à l'aide de l'option Set up with API Gateway et d'une source d'identité dans la console Verified Permissions ne sont pas destinés à un déploiement immédiat en production. Avec votre magasin de politiques initial,

finalisez votre modèle d'autorisation et exportez les ressources du magasin de politiques vers CloudFormation. Déployez les autorisations vérifiées en production de manière programmatique avec le AWS Cloud Development Kit (CDK). Pour plus d'informations, consultez Passage à la production avec AWS CloudFormation.

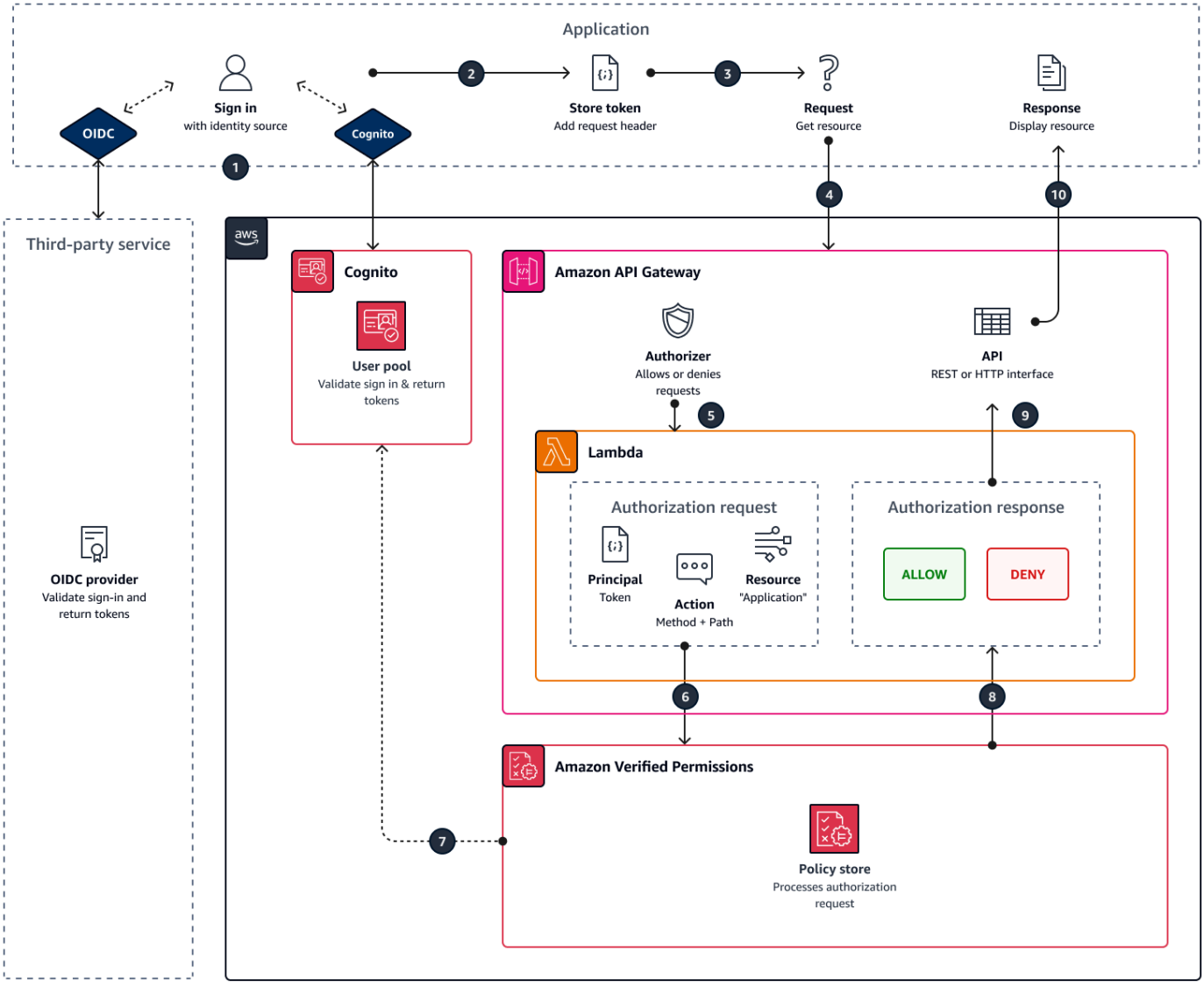
Dans un magasin de politiques lié à une API et à une source d'identité, votre application présente un jeton de pool d'utilisateurs dans un en-tête d'autorisation lorsqu'elle envoie une demande à l'API. La source d'identité de votre magasin de politiques fournit une validation par jeton pour les autorisations vérifiées. Le jeton forme `principal` les demandes d'autorisation internes avec l'IsAuthorizedWithTokenAPI. Verified Permissions élabore des politiques relatives à l'appartenance à un groupe d'utilisateurs, comme indiqué dans une réclamation de groupe sous forme de jetons d'identité (ID) et d'accès, par exemple `cognito:groups` pour les groupes d'utilisateurs. Votre API traite le jeton de votre application dans un autorisateur Lambda et le soumet à Verified Permissions pour une décision d'autorisation. Lorsque votre API reçoit la décision d'autorisation de la part de l'autorisateur Lambda, elle transmet la demande à votre source de données ou refuse la demande.

Composants de la source d'identité et de l'autorisation API Gateway avec autorisations vérifiées

- Un groupe d'utilisateurs Amazon Cognito ou un IdP OIDC qui authentifie et regroupe les utilisateurs. Les jetons des utilisateurs renseignent l'appartenance au groupe et le principal ou le contexte que Verified Permissions évalue dans votre magasin de politiques.
- Une API REST API Gateway. Les autorisations vérifiées définissent les actions à partir des chemins d'API et des méthodes d'API, par exemple`MyAPI::Action::get /photo`.
- Une fonction Lambda et un autorisateur Lambda pour votre API. La fonction Lambda reçoit les jetons porteurs de votre groupe d'utilisateurs, demande l'autorisation à Verified Permissions et renvoie une décision à API Gateway. Le flux de travail Set up with Cognito and API Gateway crée automatiquement cet autorisateur Lambda pour vous.
- Un magasin de politiques d'autorisations vérifiées. La source d'identité du Policy Store est votre groupe d'utilisateurs. Le schéma du magasin de politiques reflète la configuration de votre API, et les politiques lient les groupes d'utilisateurs aux actions d'API autorisées.
- Une application qui authentifie les utilisateurs auprès de votre IdP et ajoute des jetons aux demandes d'API.

Comment les autorisations vérifiées autorisent-elles les demandes d'API

Lorsque vous créez un nouveau magasin de politiques et que vous sélectionnez l'option Configurer avec Cognito et API Gateway, Verified Permissions crée le schéma et les politiques du magasin de politiques. Le schéma et les politiques reflètent les actions de l'API et les groupes d'utilisateurs que vous souhaitez autoriser à effectuer ces actions. Verified Permissions crée également la fonction Lambda et l'autorisateur. Vous devez configurer le nouvel autorisateur sur une méthode de votre API.



1. Votre utilisateur se connecte à votre application via Amazon Cognito ou un autre IdP OIDC. L'IdP émet des identifiants et des jetons d'accès contenant les informations de l'utilisateur.

2. Votre application stocke les JWT. Pour plus d'informations, consultez la section Utilisation de jetons avec des groupes d'utilisateurs dans le manuel Amazon Cognito Developer Guide.
3. Votre utilisateur demande des données que votre application doit récupérer à partir d'une API externe.
4. Votre application demande des données à une API REST dans API Gateway. Il ajoute un identifiant ou un jeton d'accès en tant qu'en-tête de demande.
5. Si votre API dispose d'un cache pour la décision d'autorisation, elle renvoie la réponse précédente. Si la mise en cache est désactivée ou si l'API n'a pas de cache actuel, API Gateway transmet les paramètres de la demande à un autorisateur Lambda basé sur des jetons.
6. La fonction Lambda envoie une demande d'autorisation à un magasin de politiques d'autorisations vérifiées avec l'IsAuthorizedWithTokenAPI. La fonction Lambda transmet les éléments d'une décision d'autorisation :
   a. Le jeton de l'utilisateur en tant que principal.
   b. La méthode API combinée au chemin de l'API, par exemple`GetPhoto`, en tant qu'action.
   c. Le terme `Application` en tant que ressource.
7. Les autorisations vérifiées valident le jeton. Pour plus d'informations sur la manière dont les jetons Amazon Cognito sont validés, consultez la section Autorisation avec autorisations vérifiées par Amazon dans le guide du développeur Amazon Cognito.
8. Verified Permissions évalue la demande d'autorisation par rapport aux politiques de votre magasin de politiques et renvoie une décision d'autorisation.
9. L'autorisateur Lambda renvoie une `Deny` réponse `Allow` OR à API Gateway.
10. L'API renvoie des données ou une `ACCESS_DENIED` réponse à votre application. Votre application traite et affiche les résultats de la demande d'API.

Ajout d'un contrôle d'accès basé sur les attributs (ABAC)

Une session d'authentification typique avec un IdP renvoie un identifiant et des jetons d'accès. Vous pouvez transmettre l'un ou l'autre de ces types de jetons en tant que jeton porteur dans les demandes d'application adressées à votre API. En fonction de vos choix lors de la création de votre magasin de politiques, Verified Permissions attend l'un des deux types de jetons. Les deux types contiennent des informations sur l'appartenance au groupe de l'utilisateur. Pour plus d'informations sur les types de jetons dans Amazon Cognito, consultez la section Utilisation de jetons avec des groupes d'utilisateurs dans le manuel Amazon Cognito Developer Guide.

Après avoir créé un magasin de politiques, vous pouvez ajouter et étendre des politiques. Par exemple, vous pouvez ajouter de nouveaux groupes à vos politiques au fur et à mesure que vous les ajoutez à votre groupe d'utilisateurs. Étant donné que votre magasin de politiques connaît déjà la manière dont votre groupe d'utilisateurs présente les groupes sous forme de jetons, vous pouvez autoriser un ensemble d'actions pour tout nouveau groupe doté d'une nouvelle politique.

Vous souhaiterez peut-être également étendre le modèle d'évaluation des politiques basé sur les groupes à un modèle plus précis basé sur les propriétés des utilisateurs. Les jetons du pool d'utilisateurs contiennent des informations supplémentaires sur les utilisateurs qui peuvent contribuer aux décisions d'autorisation.

Jetons d'identification

Les jetons d'identification représentent les attributs d'un utilisateur et offrent le plus haut niveau de contrôle d'accès précis. Pour évaluer les adresses e-mail, les numéros de téléphone ou les attributs personnalisés tels que le service et le responsable, évaluez le jeton d'identification.

Jetons d'accès

Les jetons d'accès représentent les autorisations d'un utilisateur avec les étendues OAuth 2.0. Pour ajouter une couche d'autorisation ou pour configurer des demandes de ressources supplémentaires, évaluez le jeton d'accès. Par exemple, vous pouvez vérifier qu'un utilisateur appartient aux groupes appropriés et qu'il possède un champ d'application tel `PetStore.read` que celui qui autorise généralement l'accès à l'API. Les groupes d'utilisateurs peuvent ajouter des étendues personnalisées aux jetons grâce aux serveurs de ressources et à la personnalisation des jetons lors de l'exécution.

Voir par Utilisation des sources d'identité dans les schémas et les politiques exemple les politiques qui traitent les demandes sous forme de jetons d'identification et d'accès.

Considérations relatives aux magasins de politiques liés aux API

Lorsque vous créez un magasin de politiques lié à une API dans la console Verified Permissions, vous créez un test pour un éventuel déploiement en production. Avant de passer à la production, établissez une configuration fixe pour votre API et votre groupe d'utilisateurs. Tenez compte des facteurs suivants :

API Gateway met en cache les réponses

Dans les magasins de politiques liés à l'API, Verified Permissions crée un autorisateur Lambda avec un TTL de mise en cache d'autorisation de 120 secondes. Vous pouvez ajuster cette valeur ou désactiver la mise en cache dans votre autorisateur. Dans un autorisateur dont la mise en cache est activée, votre autorisateur renvoie la même réponse à chaque fois jusqu'à l'expiration du TTL. Cela peut prolonger la durée de vie effective des jetons du pool d'utilisateurs d'une durée égale au TTL de mise en cache de l'étape demandée.

Les groupes Amazon Cognito peuvent être réutilisés

Amazon Verified Permissions détermine l'appartenance à un groupe pour les utilisateurs du groupe d'utilisateurs à partir de la `cognito:groups` réclamation contenue dans l'identifiant ou le jeton d'accès d'un utilisateur. La valeur de cette réclamation est un tableau des noms conviviaux des groupes d'utilisateurs auxquels appartient l'utilisateur. Vous ne pouvez pas associer des groupes de groupes d'utilisateurs à un identifiant unique.

Les groupes de groupes d'utilisateurs que vous supprimez et recréez sous le même nom sont présents dans votre magasin de politiques en tant que même groupe. Lorsque vous supprimez un groupe d'un groupe d'utilisateurs, supprimez toutes les références au groupe de votre magasin de politiques.

L'espace de noms et le schéma dérivés de l'API sont point-in-time

Verified Permissions capture votre API à un moment donné : il interroge votre API uniquement lorsque vous créez votre magasin de politiques. Lorsque le schéma ou le nom de votre API change, vous devez mettre à jour votre magasin de politiques et votre autorisateur Lambda, ou créer un nouveau magasin de politiques lié à l'API. Verified Permissions dérive l'espace de noms du magasin de politiques à partir du nom de votre API.

La fonction Lambda n'a pas de configuration VPC

La fonction Lambda créée par Verified Permissions pour votre autorisateur d'API n'est pas connectée à un VPC. Par défaut. Les API dont l'accès au réseau est limité aux VPC privés ne peuvent pas communiquer avec la fonction Lambda qui autorise les demandes d'accès avec des autorisations vérifiées.

Verified Permissions déploie les ressources d'autorisation dans CloudFormation

Pour créer un magasin de politiques lié à une API, vous devez connecter un AWS principal hautement privilégié à la console Verified Permissions. Cet utilisateur déploie une AWS CloudFormation pile qui crée des ressources entre plusieurs Services AWS. Ce principal doit être

autorisé à ajouter et à modifier des ressources dans Verified Permissions IAM, Lambda et API Gateway. Il est recommandé de ne pas partager ces informations d'identification avec les autres administrateurs de votre organisation.

Consultez Passage à la production avec AWS CloudFormation pour un aperçu des ressources créées par Verified Permissions.

Passage à la production avec AWS CloudFormation

Les magasins de politiques liés aux API permettent de créer rapidement un modèle d'autorisation pour une API API Gateway. Ils sont conçus pour servir d'environnement de test pour le composant d'autorisation de votre application. Après avoir créé votre magasin de politiques de test, passez du temps à affiner les politiques, le schéma et l'autorisateur Lambda.

Vous pouvez ajuster l'architecture de votre API, ce qui nécessite des ajustements équivalents au schéma et aux politiques de votre magasin de politiques. Les magasins de politiques liés à l'API ne mettent pas automatiquement à jour leur schéma à partir de l'architecture d'API. Les autorisations vérifiées interrogent uniquement l'API au moment de la création d'un magasin de politiques. Si votre API change suffisamment, vous devrez peut-être répéter le processus avec un nouveau magasin de politiques.

Lorsque votre application et votre modèle d'autorisation sont prêts à être déployés en production, intégrez le magasin de politiques lié à l'API que vous avez développé à vos processus d'automatisation. Il est recommandé d'exporter le schéma et les politiques du magasin de politiques dans un AWS CloudFormation modèle que vous pouvez déployer sur d'autres Comptes AWS et Régions AWS.

Les résultats du processus de magasin de politiques lié à l'API sont un magasin de politiques initial et un autorisateur Lambda. L'autorisateur Lambda dispose de plusieurs ressources dépendantes. Verified Permissions déploie ces ressources dans une pile générée automatiquement CloudFormation . Pour effectuer un déploiement en production, vous devez collecter les ressources du magasin de politiques et de l'autorisateur Lambda dans un modèle. Un magasin de politiques lié à une API est composé des ressources suivantes :

1. AWS::VerifiedPermissions::PolicyStore: Copiez votre schéma dans l'`SchemaDefinition`objet. Escapez " les personnages en tant que\".
2. AWS::VerifiedPermissions::IdentitySource: Copiez les valeurs à partir de la sortie GetIdentitySource de votre magasin de politiques de test et modifiez-les si nécessaire.

3. Une ou plusieurs des options AWS::VerifiedPermissions::Policysuivantes : Copiez votre déclaration de politique dans l'`Definition`objet. Escapez " les personnages en tant que\".
4. AWS: :Lambda : :Function, AWS: : : :Role,IAM : : :Policy,AWS :IAM : : :Authorizer,AWS ApiGateway AWS::Lambda::Permission: Copiez le modèle depuis l'onglet Modèle de la pile que Verified Permissions a déployée lorsque vous avez créé votre magasin de politiques.

Le modèle suivant est un exemple de magasin de politiques. Vous pouvez ajouter les ressources de l'autorisateur Lambda de votre pile existante à ce modèle.

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

Résolution des problèmes liés aux magasins de politiques liés à l'API

Utilisez les informations fournies ici pour vous aider à diagnostiquer et à résoudre les problèmes courants lorsque vous créez des boutiques de politiques liées à l'API Amazon Verified Permissions.

Rubriques

J'ai mis à jour ma politique, mais la décision d'autorisation n'a pas changé

Par défaut, Verified Permissions configure l'autorisateur Lambda pour mettre en cache les décisions d'autorisation pendant 120 secondes. Réessayez au bout de deux minutes ou désactivez le cache sur votre système d'autorisation. Pour plus d'informations, consultez la section Activer la mise en cache des API pour améliorer la réactivité dans le manuel Amazon API Gateway Developer Guide.

J'ai joint l'autorisateur Lambda à mon API mais il ne génère pas de demandes d'autorisation

Pour commencer à traiter les demandes, vous devez déployer le stage d'API auquel vous avez associé votre autorisateur. Pour plus d'informations, consultez la section Déploiement d'une API REST dans le manuel Amazon API Gateway Developer Guide.

J'ai reçu une décision d'autorisation inattendue et je souhaite revoir la logique d'autorisation

Le processus de magasin de politiques lié à l'API crée une fonction Lambda pour votre autorisateur. Verified Permissions intègre automatiquement la logique de vos décisions d'autorisation dans la

fonction d'autorisation. Après avoir créé votre magasin de règles, vous pouvez revenir en arrière pour revoir et mettre à jour la logique de la fonction.

Pour localiser votre fonction Lambda depuis la AWS CloudFormation console, cliquez sur le bouton Vérifier le déploiement sur la page de présentation de votre nouveau magasin de politiques.

Vous pouvez également localiser votre fonction dans la AWS Lambda console. Accédez à la console dans votre magasin Région AWS de politiques et recherchez le nom d'une fonction avec le préfixe de`AVPAuthorizerLambda`. Si vous avez créé plusieurs magasins de politiques liés à l'API, utilisez l'heure de dernière modification de vos fonctions pour les corréler avec la création du magasin de politiques.

Je souhaite trouver les journaux de mon autorisateur Lambda

Les fonctions Lambda collectent des métriques et enregistrent les résultats de leurs invocations sur Amazon. CloudWatch Pour consulter vos journaux, localisez votre fonction dans la console Lambda et choisissez l'onglet Monitor. Sélectionnez Afficher CloudWatch les journaux et passez en revue les entrées du groupe de journaux.

Pour plus d'informations sur les journaux de fonctions Lambda, consultez la section Utilisation d'Amazon CloudWatch Logs AWS Lambda dans le manuel du AWS Lambda développeur.

Mon autorisateur Lambda n'existe pas

Une fois que vous avez terminé la configuration d'un magasin de politiques lié à une API, vous devez associer l'autorisateur Lambda à votre API. Si vous ne trouvez pas votre autorisateur dans la console API Gateway, les ressources supplémentaires pour votre magasin de politiques ont peut-être échoué ou n'ont pas encore été déployées. Les magasins de politiques liés à des API déploient ces ressources dans une AWS CloudFormation pile.

Les autorisations vérifiées affichent un lien avec le libellé Vérifier le déploiement à la fin du processus de création. Si vous avez déjà quitté cet écran, accédez à la CloudFormation console et recherchez dans les piles récentes un nom préfixé par. `AVPAuthorizer-<policy store ID>` CloudFormation fournit des informations de dépannage précieuses dans le résultat d'un déploiement en pile.

Pour obtenir de l'aide pour résoudre les problèmes liés aux CloudFormation piles, consultez la section Résolution des problèmes CloudFormation dans le Guide de AWS CloudFormation l'utilisateur.

Mon API se trouve dans un VPC privé et ne peut pas invoquer l'autorisateur

Les autorisations vérifiées ne prennent pas en charge l'accès aux autorisateurs Lambda via les points de terminaison VPC. Vous devez ouvrir un chemin réseau entre votre API et la fonction Lambda qui vous sert d'autorisateur.

Je souhaite traiter des attributs utilisateur supplémentaires dans mon modèle d'autorisation

Le processus de stockage des politiques lié à l'API déduit les politiques d'autorisations vérifiées à partir des revendications des groupes dans les jetons des utilisateurs. Pour mettre à jour votre modèle d'autorisation afin de prendre en compte des attributs utilisateur supplémentaires, intégrez ces attributs dans vos politiques.

Vous pouvez associer de nombreuses demandes d'identification et de jetons d'accès provenant des groupes d'utilisateurs Amazon Cognito aux déclarations de politique relatives aux autorisations vérifiées. Par exemple, la plupart des utilisateurs ont une `email` réclamation dans leur jeton d'identification. Pour plus d'informations sur l'ajout de réclamations provenant de votre source d'identité aux politiques, consultezUtilisation des sources d'identité dans les schémas et les politiques.

Je souhaite ajouter de nouvelles actions, de nouveaux attributs de contexte d'action ou de nouvelles ressources

Un magasin de politiques lié à une API et l'autorisateur Lambda qu'il crée constituent une ressource. point-in-time Ils reflètent l'état de votre API au moment de sa création. Le schéma du magasin de politiques n'attribue aucun attribut de contexte aux actions, ni aucun attribut ou parent à la `Application` ressource par défaut.

Lorsque vous ajoutez des actions (chemins et méthodes) à votre API, vous devez mettre à jour votre magasin de règles pour être au courant des nouvelles actions. Vous devez également mettre à jour votre autorisateur Lambda pour traiter les demandes d'autorisation pour les nouvelles actions. Vous pouvez recommencer avec un nouveau magasin de politiques ou mettre à jour votre magasin de politiques existant.

Pour mettre à jour votre magasin de polices existant, localisez votre fonction. Examinez la logique de la fonction générée automatiquement et mettez-la à jour pour traiter les nouvelles actions, les nouveaux attributs ou le nouveau contexte. Modifiez ensuite votre schéma pour inclure les nouvelles actions et les nouveaux attributs.

Changer de magasin de politiques d'autorisations vérifiées

AWS Management Console

Pour changer de magasin de politiques ou créer des magasins de politiques supplémentaires

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Switch à côté de Current Policy Store.
3. Vous pouvez passer d'un magasin de politiques existant à un autre ou créer des magasins de politiques supplémentaires.
   - Pour changer de magasin de politiques, choisissez l'ID du magasin de politiques vers lequel passer.
   - Pour créer un nouveau magasin de politiques, choisissez Créer un nouveau magasin de politiques. Suivez les instructions de la section Création de magasins de politiques d'autorisations vérifiées.

AWS CLI

Pour changer de magasin de politiques ou créer des magasins de politiques supplémentaires

Le AWS CLI ne gère pas de magasin de politiques « par défaut ». La plupart des AWS CLI commandes utilisent plutôt le `--policy-store-id` pour spécifier le magasin de politiques à utiliser pour chaque commande.

Pour créer un nouveau magasin de politiques, utilisez la create-policy-storecommande.

Suppression des magasins de politiques d'autorisations vérifiées

AWS Management Console

Pour supprimer un magasin de politiques

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Settings.
3. Choisissez Supprimer ce magasin de règles.

4. Tapez `delete` dans la zone de texte et choisissez Supprimer.

AWS CLI

Pour supprimer un magasin de politiques

Vous pouvez supprimer un magasin de politiques à l'aide de cette `delete-policy-store` opération.

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

Cette commande ne produit aucune sortie en cas de réussite.

Schéma de la boutique Amazon Verified Permissions Policy

Un schéma est une déclaration de la structure des types d'entités pris en charge par votre application et des actions que votre application peut fournir dans les demandes d'autorisation.

Pour plus d'informations, consultez le format du schéma Cedar dans le Guide de référence du langage politique de Cedar.

> **Note**
>
> L'utilisation de schémas dans Verified Permissions est facultative, mais elle est vivement recommandée pour les logiciels de production. Lorsque vous créez une nouvelle politique, Verified Permissions peut utiliser le schéma pour valider les entités et les attributs référencés dans le champ d'application et les conditions afin d'éviter les fautes de frappe et les erreurs dans les politiques susceptibles d'entraîner un comportement confus du système. Si vous activez la validation des politiques, toutes les nouvelles politiques doivent être conformes au schéma.

AWS Management Console

Pour créer un schéma

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices d'assurance.
2. Dans le volet de navigation de gauche, choisissez Schema.
3. Choisissez Create schema (Créer un schéma).

AWS CLI

Pour soumettre un nouveau schéma ou remplacer un schéma existant à l'aide du AWS CLI.

Vous pouvez créer un magasin de politiques en exécutant une AWS CLI commande similaire à l'exemple suivant.

Prenons l'exemple d'un schéma contenant le contenu Cedar suivant :

```
{
```

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

Vous devez d'abord transformer le JSON en une chaîne d'une seule ligne, et le préfacer avec une déclaration de son type de données :`cedarJson`. L'exemple suivant utilise le contenu suivant d'un `schema.json` fichier qui contient la version échappée du schéma JSON.

> **Note**
>
> L'exemple ci-dessous est encadré par des lignes pour plus de lisibilité. Vous devez avoir le fichier entier sur une seule ligne pour que la commande l'accepte.

```
{"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

Vous pouvez créer un magasin de politiques à l'aide de l'`PutSchema`API. Pour plus d'informations, consultez le guide PutSchemade référence de l'API Amazon Verified Permissions.

Modification de schémas en mode visuel

Lorsque vous sélectionnez Schéma dans la console des autorisations vérifiées, le mode visuel affiche les types d'entités et les actions qui constituent votre schéma. Dans cette vue de haut niveau ou dans les détails de n'importe quelle entité, vous pouvez choisir Modifier le schéma pour commencer à mettre à jour votre schéma. Le mode visuel n'est pas disponible avec certains formats de schéma tels que les enregistrements imbriqués.

L'éditeur de schéma visuel commence par une série de diagrammes illustrant les relations entre les entités de votre schéma. Choisissez Expand pour optimiser votre vision des relations entre les entités de votre schéma.

Schéma des actions

La vue du diagramme des actions répertorie les types de principes que vous avez configurés dans votre magasin de politiques, les actions qu'ils peuvent effectuer et les ressources sur lesquelles ils peuvent effectuer des actions. Les lignes entre les entités indiquent votre capacité à créer une politique qui permet au principal d'agir sur une ressource. Si votre diagramme d'actions n'indique aucune relation entre deux entités, vous devez créer cette relation entre elles avant de pouvoir l'autoriser ou la refuser dans les politiques. Sélectionnez une entité pour obtenir une vue d'ensemble des propriétés et effectuez un défilement vers le bas pour afficher tous les détails. Choisissez Filtrer selon cette [action | type de ressource | type principal] pour afficher une entité dans une vue avec uniquement ses propres connexions.

Schéma des types d'entités

Le diagramme des types d'entités met l'accent sur les relations entre les principaux et les ressources. Pour comprendre les relations parentales imbriquées complexes dans votre schéma, consultez ce diagramme. Passez le pointeur de la souris sur une entité pour examiner les relations parentales qu'elle entretient.

Sous les diagrammes se trouvent des vues de liste des types d'entités et des actions de votre schéma. L'affichage par liste est utile lorsque vous souhaitez afficher immédiatement les détails d'une action ou d'un type d'entité spécifique. Sélectionnez n'importe quelle entité pour afficher les détails.

Pour modifier un schéma d'autorisations vérifiées en mode visuel

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices d'assurance.
2. Dans le volet de navigation de gauche, choisissez Schema.
3. Choisissez le mode visuel. Passez en revue les diagrammes entité-relation et planifiez les modifications que vous souhaitez apporter à votre schéma. Vous pouvez éventuellement filtrer par une entité pour examiner ses connexions individuelles avec d'autres entités.
4. Choisissez Edit schema (Modifier le schéma).
5. Dans la section Détails, saisissez un espace de noms pour votre schéma.
6. Dans la section Types d'entités, choisissez Ajouter un nouveau type d'entité.
7. Entrez le nom de l'entité.
8. (Facultatif) Choisissez Ajouter un parent pour ajouter les entités parents dont la nouvelle entité est membre. Pour supprimer un parent qui a été ajouté à l'entité, choisissez Supprimer à côté du nom du parent.
9. Choisissez Ajouter un attribut pour ajouter des attributs à l'entité. Tapez le nom de l'attribut et choisissez le type d'attribut pour chaque attribut de l'entité. Verified Permissions utilise les valeurs d'attribut spécifiées lors de la vérification des politiques par rapport au schéma. Indiquez si chaque attribut est obligatoire. Pour supprimer un attribut qui a été ajouté à l'entité, choisissez Supprimer à côté de l'attribut.
10. Choisissez Ajouter un type d'entité pour ajouter l'entité au schéma.
11. Dans la section Actions, choisissez Ajouter une nouvelle action.
12. Entrez le nom de l'action.
13. (Facultatif) Choisissez Ajouter une ressource pour ajouter les types de ressources auxquels l'action s'applique. Pour supprimer un type de ressource qui a été ajouté à l'action, choisissez Supprimer à côté du nom du type de ressource.

14. (Facultatif) Choisissez Ajouter un principal pour ajouter un type de principal auquel l'action s'applique. Pour supprimer un type principal qui a été ajouté à l'action, choisissez Supprimer à côté du nom du type principal.
15. Choisissez Ajouter un attribut pour ajouter des attributs qui peuvent être ajoutés au contexte d'une action dans vos demandes d'autorisation. Entrez le nom de l'attribut et choisissez le type d'attribut pour chaque attribut. Verified Permissions utilise les valeurs d'attribut spécifiées lors de la vérification des politiques par rapport au schéma. Indiquez si chaque attribut est obligatoire. Pour supprimer un attribut qui a été ajouté à l'action, choisissez Supprimer à côté de l'attribut.
16. Choisissez Add action.
17. Une fois que tous les types d'entités et actions ont été ajoutés au schéma, choisissez Enregistrer les modifications.

Modification de schémas en mode JSON

Pour modifier un schéma d'autorisations vérifiées en mode JSON

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices d'assurance.
2. Dans le volet de navigation de gauche, choisissez Schema.
3. Choisissez le mode JSON, puis sélectionnez Modifier le schéma.
4. Entrez le contenu de votre schéma JSON dans le champ Contenu. Vous ne pouvez pas enregistrer les mises à jour de votre schéma tant que vous n'avez pas résolu toutes les erreurs de syntaxe. Vous pouvez choisir Format JSON pour formater la syntaxe JSON de votre schéma avec l'espacement et l'indentation recommandés.
5. Sélectionnez Enregistrer les modifications.

Suppression d'un schéma

AWS Management Console

Pour supprimer un schéma d'autorisations vérifiées

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices d'assurance.
2. Dans le volet de navigation de gauche, choisissez Schema.

3. Choisissez Supprimer le schéma.

AWS CLI

Pour supprimer un schéma d'autorisations vérifiées

Il n'existe pas de commande de suppression du schéma. Vous pouvez supprimer le schéma dans un magasin de politiques à l'aide de la `put-schema` commande avec un schéma vide dans `cedarJson` le champ. Un schéma vide est représenté par une paire d'accolades « {} ».

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

Mode de validation de la politique d'autorisations Amazon Verified

Vous pouvez définir le mode de validation des politiques dans Permissions vérifiées pour contrôler si les modifications de politique sont validées par rapport au schéma de votre magasin de politiques.

> **Important**
>
> Lorsque vous activez la validation des politiques, toutes les tentatives de création ou de mise à jour d'une politique ou d'un modèle de stratégie sont validées par rapport au schéma dans le magasin de politiques. Verified Permissions rejette la demande en cas d'échec de la validation.

AWS Management Console

Pour définir le mode de validation des politiques pour un magasin de politiques

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Sélectionnez Settings (Paramètres).
3. Dans la section Mode de validation des politiques, choisissez Modifier.
4. Effectuez l'une des actions suivantes :
   - Pour activer la validation des politiques et faire en sorte que toutes les modifications de politique soient validées par rapport à votre schéma, cliquez sur le bouton radio Strict (recommandé).
   - Pour désactiver la validation des politiques en cas de modification des politiques, cliquez sur le bouton radio Désactivé. Tapez `confirm` pour confirmer que les mises à jour des politiques ne seront plus validées par rapport à votre schéma.
5. Sélectionnez Enregistrer les modifications.

AWS CLI

Pour définir le mode de validation d'un magasin de politiques

Vous pouvez modifier le mode de validation d'un magasin de politiques en utilisant l'UpdatePolicyStoreopération et en spécifiant une valeur différente pour le ValidationSettingsparamètre.

```
$ 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"
    }
}
```

Pour plus d'informations, consultez la section Validation des politiques dans le Guide de référence du langage des politiques de Cedar.

Politiques relatives aux autorisations vérifiées par Amazon

Une politique est une déclaration qui permet ou interdit à un mandant d'effectuer une ou plusieurs actions sur une ressource. Chaque politique est évaluée indépendamment de toute autre politique. Pour plus d'informations sur la manière dont les politiques Cedar sont structurées et évaluées, consultez la section Validation des politiques Cedar par rapport au schéma dans le Guide de référence du langage de politique Cedar.

Important

Lorsque vous rédigez des politiques Cedar qui font référence à des principes, à des ressources et à des actions, vous pouvez définir les identifiants uniques utilisés pour chacun de ces éléments. Nous vous recommandons vivement de suivre les meilleures pratiques suivantes :

- Utilisez des valeurs telles que les identificateurs uniques universels (UUID) pour tous les identificateurs principaux et de ressources.

  Par exemple, si un utilisateur `jane` quitte l'entreprise et que vous autorisez ensuite quelqu'un d'autre à utiliser le nom`jane`, ce nouvel utilisateur a automatiquement accès à tout ce qui est accordé par les politiques qui font toujours référence`User::"jane"`. Cedar ne fait pas la distinction entre le nouvel utilisateur et l'ancien. Cela s'applique à la fois aux identificateurs principaux et aux identificateurs de ressources. Utilisez toujours des identifiants dont l'unicité est garantie et qui ne sont jamais réutilisés afin de ne pas autoriser l'accès par inadvertance en raison de la présence d'un ancien identifiant dans une politique.

  Lorsque vous utilisez un UUID pour une entité, nous vous recommandons de le suivre avec le spécificateur//comment et le nom « convivial » de votre entité. Cela permet de faciliter la compréhension de vos politiques. Par exemple : principal == User : « A1B2C3D4-E5F6-A1B2-C3D4-Example11111",//alice
- N'incluez pas d'informations d'identification personnelle, confidentielles ou sensibles dans l'identifiant unique de vos mandants ou de vos ressources. Ces identifiants sont inclus dans les entrées de journal partagées dans les AWS CloudTrail sentiers.

Formatage des entités dans Amazon Verified Permissions

Amazon Verified Permissions utilise le langage de politique de Cedar pour créer des politiques. La syntaxe des politiques et les types de données pris en charge correspondent à la syntaxe et aux types de données décrits dans les rubriques Construction de politiques de base dans Cedar et Types de données pris en charge par Cedar dans le Guide de référence du langage de politique Cedar. Cependant, il existe des différences entre Verified Permissions et Cedar dans le formatage des entités lors d'une demande d'autorisation.

Le formatage JSON des entités dans Verified Permissions diffère de celui de Cedar pour les raisons suivantes :

- Dans Verified Permissions, toutes les paires clé-valeur d'un objet JSON doivent être encapsulées dans un objet JSON portant le nom de. `Record`
- Une liste JSON dans Verified Permissions doit être encapsulée dans une paire clé-valeur JSON où le nom de la clé est `Set` et la valeur est la liste JSON originale de Cedar.
- Pour`String`,`Long`, et les noms `Boolean` de type, chaque paire clé-valeur de Cedar est remplacée par un objet JSON dans Verified Permissions. Le nom de l'objet est le nom de clé d'origine. À l'intérieur de l'objet JSON, il existe une paire clé-valeur dont le nom de clé est le nom de type de la valeur scalaire (`String`,`Long`, ou`Boolean`) et la valeur est la valeur de l'entité Cedar.
- Le formatage syntaxique des entités Cedar et des entités Verified Permissions diffère des manières suivantes :

Format cèdre	Format des autorisations vérifiées
`uid`	`Identifier`
`type`	`EntityType`
`id`	`EntityId`
`attrs`	`Attributes`
`parents`	`Parents`

L'exemple suivant montre comment les entités d'une liste sont mises en forme à l'aide de Cedar.

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

L'exemple suivant montre comment les mêmes entités que celles de l'exemple précédent de liste Cedar sont formatées dans Permissions vérifiées.

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

L'exemple suivant montre comment les entités Cedar sont formatées pour évaluer une politique dans une demande d'autorisation.

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

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

L'exemple suivant montre comment les mêmes entités de l'exemple précédent de Cedar sont formatées dans les autorisations vérifiées.

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

Création de politiques statiques relatives aux autorisations vérifiées par Amazon

Vous pouvez créer une politique statique Cedar pour autoriser ou interdire aux principaux d'effectuer des actions spécifiques sur des ressources spécifiques pour votre application.

AWS Management Console

Pour créer une politique statique

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices d'assurance.
2. Dans le panneau de navigation de gauche, sélectionnez Policies (Politiques).
3. Choisissez Créer une politique, puis sélectionnez Créer une politique statique.
4. Dans la section Effet de la politique, choisissez si la politique autorisera ou interdira lorsqu'une demande correspond à la politique.
5. Dans le champ Champ d'application des principes, choisissez le champ d'application des principes auxquels la politique s'appliquera.
   - Choisissez Spécific principal pour appliquer la politique à un principal spécifique. Spécifiez le type d'entité et l'identifiant du principal qui sera autorisé ou interdit à effectuer les actions spécifiées dans la politique.
   - Choisissez Groupe de directeurs pour appliquer la politique à un groupe de directeurs. Tapez le nom du groupe principal dans le champ Groupe de principaux.
   - Choisissez Tous les principaux pour appliquer la politique à tous les principaux de votre magasin de polices.
6. Dans le champ Champ d'application des ressources, choisissez l'étendue des ressources auxquelles la politique s'appliquera.
   - Choisissez Ressources spécifiques pour appliquer la politique à une ressource spécifique. Spécifiez le type d'entité et l'identifiant de la ressource à laquelle la politique doit s'appliquer.
   - Choisissez Groupe de ressources pour appliquer la politique à un groupe de ressources. Tapez le nom du groupe de ressources dans le champ Groupe de ressources.
   - Choisissez Toutes les ressources pour appliquer la politique à toutes les ressources de votre magasin de politiques.

7. Dans la section Champ d'application des actions, choisissez l'étendue des ressources auxquelles la politique s'appliquera.
   - Choisissez un ensemble d'actions spécifique pour appliquer la politique à un ensemble d'actions. Cochez les cases situées à côté des actions pour appliquer la politique.
   - Choisissez Toutes les actions pour appliquer la politique à toutes les actions de votre magasin de politiques.
8. Choisissez Suivant.
9. Dans la section Politique, passez en revue votre politique Cedar. Vous pouvez choisir Format pour formater la syntaxe de votre politique avec l'espacement et l'indentation recommandés. Pour plus d'informations, voir Construction de politiques de base dans Cedar dans le Guide de référence du langage politique de Cedar.
10. Dans la section Détails, saisissez une description facultative de la politique.
11. Choisissez Créer une politique.

AWS CLI

Pour créer une politique statique

Vous pouvez créer une politique statique à l'aide de l'CreatePolicyopération. L'exemple suivant crée une politique statique simple.

```
$ 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"
}
```

Modification des politiques statiques d'Amazon Verified Permissions

Vous pouvez modifier une politique statique Cedar existante dans votre magasin de politiques. Vous ne pouvez mettre à jour directement que les politiques statiques. Vous ne pouvez modifier que certains éléments d'une politique statique :

- Le `action` référencé par la politique.
- Une clause conditionnelle, telle que `when` et`unless`.

Vous ne pouvez pas modifier les éléments suivants d'une politique statique :

- Modification d'une politique statique en une politique liée à un modèle.
- Modification de l'effet d'une politique statique à partir de `permit` ou`forbid`.
- Le `principal` référencé par une politique statique.
- Le `resource` référencé par une politique statique.

Pour modifier une politique liée à un modèle, vous devez plutôt mettre à jour le modèle. Pour plus d'informations, consultez Modification de modèles de politiques.

AWS Management Console

Pour modifier une politique statique

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices d'assurance.
2. Dans le panneau de navigation de gauche, sélectionnez Policies (Politiques).
3. Cliquez sur le bouton radio situé à côté de la politique statique à modifier, puis sélectionnez Modifier.
4. Dans la section Corps de la politique, mettez à jour la clause `action` or condition de votre politique statique. Vous ne pouvez pas mettre à jour l'effet de la politique`principal`, ou `resource` de la politique.
5. Choisissez Mettre à jour une politique.

> **Note**
>
> Si la validation des politiques est activée dans le magasin de politiques, la mise à jour d'une politique statique amène Verified Permissions à valider la politique par rapport au schéma du magasin de politiques. Si la politique statique mise à jour n'est pas validée, l'opération échoue et la mise à jour n'est pas enregistrée.

AWS CLI

Pour modifier une politique statique

Vous pouvez modifier une politique statique à l'aide de l'UpdatePolicyopération. L'exemple suivant modifie une politique statique simple.

L'exemple utilise le fichier `definition.txt` pour contenir la définition de la politique.

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

La commande suivante fait référence à ce fichier.

```
$ 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"
    }
}
```

Politiques d'affichage

AWS Management Console

Pour consulter vos politiques d'autorisations vérifiées

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices d'assurance.
2. Dans le panneau de navigation de gauche, sélectionnez Policies (Politiques). Toutes les politiques que vous avez créées sont affichées.
3. Choisissez la zone de texte Rechercher pour filtrer les politiques par principal ou par ressource.
4. Cliquez sur le bouton radio à côté d'une stratégie pour afficher les détails de celle-ci, tels que la date de création, de mise à jour et le contenu de la stratégie.
5. Vous pouvez supprimer une politique en cliquant sur le bouton radio situé à côté d'une politique, puis en choisissant Supprimer. Choisissez Supprimer la politique pour confirmer la suppression de la politique.

AWS CLI

Pour répertorier toutes les politiques disponibles dans un magasin de politiques

Vous pouvez consulter la liste des politiques à l'aide de l'GetPolicyopération. L'exemple suivant extrait une liste qui inclut une politique statique et une politique liée à un modèle.

```
$ aws verifiedpermissions list-policies \
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "Policies": [
        {
            "createdDate": "2023-05-17T18:38:31.359864+00:00",
```

```
            "definition": {
                "static": {
                    "Description": "Grant everyone of janeFriends UserGroup access
 to the vacationFolder Album"
                }
            },
            "lastUpdatedDate": "2023-05-18T16:15:04.366237+00:00",
            "policyId": "SPEXAMPLEabcdefg111111",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "STATIC",
            "resource": {
                "entityId": "publicFolder",
                "entityType": "Album"
            }
        },
        {
            "createdDate": "2023-05-22T18:57:53.298278+00:00",
            "definition": {
                "templateLinked": {
                    "policyTemplateId": "PTEXAMPLEabcdefg111111"
                }
            },
            "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",
            "policyId": "TPEXAMPLEabcdefg111111",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "TEMPLATELINKED",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
            },
            "resource": {
                "entityId": "VacationPhoto94.jpg",
                "entityType": "Photo"
            }
        }
    ]
}
```

Pour consulter les détails d'une police individuelle

Vous pouvez récupérer les détails d'une politique à l'aide de l'GetPolicyopération. L'exemple suivant permet de récupérer les détails d'une politique liée à un modèle.

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

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

Exemples de politiques relatives aux autorisations vérifiées par Amazon

Les exemples de politique d'autorisations vérifiées suivants sont basés sur le schéma défini pour l'application hypothétique appelée PhotoFlash décrite dans la section Exemple de schéma du Guide

de référence du langage de politique Cedar. Pour plus d'informations sur la syntaxe des politiques Cedar, consultez la section Construction de base des politiques dans Cedar dans le Guide de référence du langage de politique Cedar.

Exemples de politiques

- Permet l'accès à des entités individuelles
- Permet l'accès à des groupes d'entités
- Permet l'accès à n'importe quelle entité
- Autorise l'accès aux attributs d'une entité (ABAC)
- Refuse l'accès

Permet l'accès à des entités individuelles

Cet exemple montre comment créer une politique permettant à l'utilisateur `alice` de voir la photo`VacationPhoto94.jpg`.

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

Permet l'accès à des groupes d'entités

Cet exemple montre comment créer une politique permettant à tous les membres du groupe `alice_friends` de voir la photo`VacationPhoto94.jpg`.

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

Cet exemple montre comment créer une politique permettant à l'utilisateur de `alice` visualiser n'importe quelle photo de l'album`alice_vacation`.

```
permit(
  principal == User::"alice",
```

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

Cet exemple montre comment créer une politique permettant à l'utilisateur d'`alice`afficher, de modifier ou de supprimer n'importe quelle photo de l'album`alice_vacation`.

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

Cet exemple montre comment vous pouvez créer une politique qui autorise les utilisateurs de l'album `alice` `alice_vacation` à accéder à un groupe défini dans la hiérarchie du schéma qui contient les autorisations nécessaires pour afficher, modifier et supprimer une photo. `admin`

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

Cet exemple montre comment vous pouvez créer une politique qui autorise l'utilisateur `alice` à accéder à l'album`alice_vacation`, lorsqu'un groupe `viewer` est défini dans la hiérarchie du schéma et qui contient l'autorisation de visualiser et de commenter une photo. L'utilisateur `alice` obtient également l'`edit`autorisation par la deuxième action répertoriée dans la politique.

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

Permet l'accès à n'importe quelle entité

Cet exemple montre comment créer une politique permettant à n'importe quel principal authentifié de consulter l'album`alice_vacation`.

```
permit(
```

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

Cet exemple montre comment créer une politique permettant à l'utilisateur de `alice` répertorier tous les albums du `jane` compte, de répertorier les photos de chaque album et de visualiser les photos du compte.

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

Cet exemple montre comment créer une politique permettant à l'utilisateur d'`alice`effectuer n'importe quelle action sur les ressources de l'album`jane_vaction`.

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

Autorise l'accès aux attributs d'une entité (ABAC)

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Les autorisations vérifiées permettent d'associer des attributs aux principes, aux actions et aux ressources. Ces attributs peuvent ensuite être référencés dans les `unless` clauses `when` et des politiques qui évaluent les attributs des principes, des actions et des ressources qui constituent le contexte de la demande.

Les exemples suivants utilisent les attributs définis dans l'application hypothétique appelée PhotoFlash décrite dans la section Exemple de schéma du Guide de référence du langage de politique Cedar.

Cet exemple montre comment créer une politique permettant à tout directeur `HardwareEngineering` du département dont le niveau de poste est supérieur ou égal à 5 de consulter et de répertorier les photos de l'album`device_prototypes`.

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

Cet exemple montre comment créer une politique permettant à l'utilisateur de `alice` visualiser n'importe quelle ressource de type fichier`JPEG`.

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

Les actions possèdent des attributs contextuels. Vous devez transmettre ces attributs dans le cadre `context` d'une demande d'autorisation. Cet exemple montre comment créer une politique permettant à l'utilisateur d'`alice`effectuer n'importe quelle `readOnly` action. Vous pouvez également définir une `appliesTo` propriété pour les actions de votre schéma. Cela spécifie des actions valides pour une ressource lorsque vous voulez vous assurer que, par exemple, les utilisateurs ne peuvent tenter `ViewPhoto` d'autoriser qu'une ressource de type`PhotoFlash::Photo`.

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

Une meilleure façon de définir les propriétés des actions dans votre schéma consiste toutefois à les organiser en groupes d'actions fonctionnels. Par exemple, vous pouvez créer une action nommée `ReadOnlyPhotoAccess` et définie `PhotoFlash::Action::"ViewPhoto"` pour être membre `ReadOnlyPhotoAccess` en tant que groupe d'actions. Cet exemple montre comment créer une politique qui accorde à Alice l'accès aux actions en lecture seule de ce groupe.

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

Cet exemple montre comment vous pouvez créer une politique qui autorise tous les principaux à effectuer n'importe quelle action sur les ressources pour lesquelles ils ont un `owner` attribut.

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

Cet exemple montre comment créer une politique permettant à n'importe quel principal de consulter n'importe quelle ressource si l'`department`attribut du principal correspond à l'`department`attribut de la ressource.

> **Note**
>
> Si aucun attribut d'une entité n'est mentionné dans une condition de stratégie, la politique sera ignorée lors de la prise d'une décision d'autorisation et l'évaluation de cette politique échouera pour cette entité. Par exemple, aucun principal qui n'a pas d'`department`attribut ne peut accéder à aucune ressource en vertu de cette politique.

```
permit(
```

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

Cet exemple montre comment créer une politique permettant à n'importe quel principal d'effectuer n'importe quelle action sur une ressource si le principal est le responsable `owner` de la ressource OU si le principal fait partie du `admins` groupe de la ressource.

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

Refuse l'accès

Si une politique contient des `forbid` informations relatives à son effet, elle limite les autorisations au lieu de les accorder.

> **Important**
>
> Lors de l'autorisation, si une `forbid` politique `permit` et une politique sont appliquées à la fois, elles `forbid` ont la priorité.

Les exemples suivants utilisent les attributs définis dans l'application hypothétique appelée PhotoFlash décrite dans la section Exemple de schéma du Guide de référence du langage de politique Cedar.

Cet exemple montre comment vous pouvez créer une politique qui empêche l'utilisateur d'`alice`effectuer toutes les actions sauf `readOnly` sur une ressource.

```
forbid (
```

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

Cet exemple montre comment vous pouvez créer une politique qui refuse l'accès à toutes les ressources dotées d'un `private` attribut, sauf si le principal possède l'`owner`attribut correspondant à la ressource.

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

Modèles de politiques Amazon Verified Permissions

Vous pouvez créer des modèles de politique Cedar dans Verified Permissions afin de définir une règle de contrôle d'accès pour votre système. Les modèles de politiques sont des politiques Cedar avec des espaces réservés pour`principal`,`resource`, ou les deux. Les modèles de politique permettent de définir une politique une seule fois, puis de l'associer à plusieurs principes et ressources. Les mises à jour du modèle de politique sont répercutées sur tous les principes et ressources qui utilisent le modèle. Pour plus d'informations, veuillez consulter la rubriqueModèles de politiques Cedardans le Guide de référence du langage politique de Cedar.

Nous vous recommandons d'utiliser des modèles de politiques pour créer des politiques qui peuvent être partagées dans l'ensemble de votre application. Par exemple, vous pouvez créer un modèle de stratégie pour un éditeur qui fournit des autorisations de lecture, de modification et de commentaire au principal et à la ressource qui utilisent le modèle de stratégie.

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

Lorsqu'un principal est désigné comme éditeur d'une ressource, votre application peut instancier une politique à l'aide du modèle pour autoriser le principal à effectuer les actions de lecture, de modification et de commentaire sur la ressource.

Création de modèles de politiques

AWS Management Console

Pour créer un modèle de politique

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, sélectionnez Modèles de politiques.
3. Choisissez Créer un modèle de politique.
4. Dans la section Détails, saisissez une description du modèle de politique.

5. Dans le corps du modèle de politique, utilisez des espaces réservés `?principal` et autorisez les politiques créées `?resource` à partir de ce modèle à personnaliser les autorisations qu'elles accordent. Vous pouvez choisir Format pour mettre en forme la syntaxe de votre modèle de politique avec l'espacement et l'indentation recommandés.
6. Choisissez Créer un modèle de politique.

AWS CLI

Pour créer un modèle de politique

Vous pouvez créer un modèle de politique à l'aide de cette CreatePolicyTemplateopération. L'exemple suivant crée un modèle de politique avec un espace réservé pour le principal.

Le fichier `template1.txt` contient les éléments suivants.

```
"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"
}
```

Création de politiques liées à un modèle

Vous pouvez créer des politiques liées à un modèle pour créer un lien vers un modèle de stratégie. Les politiques liées à des modèles restent liées à leurs modèles de politiques. Si vous modifiez la déclaration de politique dans le modèle de stratégie, toutes les politiques liées à ce modèle utilisent

automatiquement la nouvelle déclaration pour toutes les décisions d'autorisation prises à partir de ce moment.

AWS Management Console

Pour créer une politique liée à un modèle en instanciant un modèle de stratégie

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le panneau de navigation de gauche, sélectionnez Policies (Politiques).
3. Choisissez Créer une politique, puis choisissez Créer une politique liée à un modèle.
4. Cliquez sur le bouton radio situé à côté du modèle de politique à utiliser, puis sur Suivant.
5. Tapez le principal et la ressource à utiliser pour cette instance spécifique de la politique liée au modèle. Les valeurs spécifiées sont affichées dans le champ d'aperçu de la déclaration de politique.

   > **Note**
   >
   > Les valeurs Principal et Resource doivent avoir le même format que les politiques statiques. Par exemple, pour spécifier le `AdminUsers` groupe du principal, tapez`Group::"AdminUsers"`. Si vous tapez`AdminUsers`, une erreur de validation s'affiche.

6. Choisissez Créer une politique liée à un modèle.

   La nouvelle politique liée au modèle est affichée sous Politiques.

AWS CLI

Pour créer une politique liée à un modèle en instanciant un modèle de stratégie

Vous pouvez créer une politique liée à un modèle qui fait référence à un modèle de politique existant et qui spécifie des valeurs pour tous les espaces réservés utilisés par le modèle.

L'exemple suivant crée une politique liée à un modèle qui utilise un modèle avec l'instruction suivante :

```
permit(
```

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

Il utilise également le `definition.txt` fichier suivant pour fournir la valeur du `definition` paramètre :

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

La sortie montre à la fois la ressource, qu'elle obtient à partir du modèle, et le principal, qu'elle obtient à partir du paramètre de définition

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

Modification de modèles de politiques

AWS Management Console

Pour modifier vos modèles de politiques

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices d'assurance.
2. Dans le volet de navigation de gauche, sélectionnez Modèles de politiques. La console affiche tous les modèles de politiques que vous avez créés dans le magasin de politiques actuel.
3. Cliquez sur le bouton radio situé à côté d'un modèle de stratégie pour afficher des détails sur le modèle de stratégie, tels que la date de création et de mise à jour du modèle de stratégie, ainsi que le contenu du modèle de stratégie.
4. Choisissez Modifier pour modifier votre modèle de politique. Mettez à jour la description de la politique et le corps de la stratégie si nécessaire, puis choisissez Mettre à jour le modèle de politique.
5. Vous pouvez supprimer un modèle de stratégie en cliquant sur le bouton radio situé à côté d'un modèle de stratégie, puis en choisissant Supprimer. Cliquez sur OK pour confirmer la suppression du modèle de politique.

AWS CLI

Pour mettre à jour un modèle de politique

Vous pouvez créer une politique statique à l'aide de l'UpdatePolicyopération. L'exemple suivant met à jour le modèle de stratégie spécifié en remplaçant son corps de stratégie par une nouvelle stratégie définie dans un fichier.

Contenu du fichier `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"
}
```

Exemples de politiques liées à un modèle pour les autorisations vérifiées, exemples de magasins de politiques

Lorsque vous créez un magasin de politiques dans Verified Permissions à l'aide de la méthode Sample policy store, votre magasin de politiques est créé avec des politiques prédéfinies, des modèles de politiques et un schéma pour l'exemple de projet que vous avez choisi. Les exemples de politiques liés au modèle d'autorisations vérifiées suivants peuvent être utilisés avec les exemples de magasins de politiques et leurs politiques, modèles de politiques et schémas respectifs.

PhotoFlashexemples de politiques liés à un modèle

Cet exemple montre comment créer une politique liée à un modèle qui utilise le modèle de politique Accorder un accès limité aux photos partagées non privées avec un utilisateur et une photo individuels.

> **Note**
>
> Le langage politique de Cedar considère qu'une entité `in` est elle-même. Par conséquent, `principal in User::"Alice"` est équivalent à`principal == User::"Alice"`.

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

Cet exemple montre comment créer une politique liée à un modèle qui utilise le modèle de politique Accorder un accès limité aux photos partagées non privées avec un utilisateur et un album individuels.

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

Cet exemple montre comment créer une politique liée à un modèle qui utilise le modèle de politique Accorder un accès limité aux photos partagées non privées avec un groupe d'amis et à une photo individuelle.

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

Cet exemple montre comment créer une politique liée à un modèle qui utilise le modèle de politique Accorder un accès limité aux photos partagées non privées avec un groupe d'amis et un album.

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

Cet exemple montre comment créer une politique liée à un modèle qui utilise le modèle de politique Accorder un accès complet aux photos partagées non privées avec un groupe d'amis et à une photo individuelle.

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

Cet exemple montre comment créer une politique liée à un modèle qui utilise le modèle de politique Bloquer l'accès d'un utilisateur à un compte.

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

DigitalPetStore

Le magasin DigitalPetStore d'exemples de politiques n'inclut aucun modèle de politique. Vous pouvez consulter les politiques incluses dans le magasin de politiques en choisissant Politiques dans le volet de navigation de gauche après avoir créé l'DigitalPetStoreexemple de magasin de politiques.

TinyToDo exemples de politiques liés à un modèle

Cet exemple montre comment créer une politique liée à un modèle qui utilise le modèle de politique qui permet aux utilisateurs d'accéder à une liste de tâches et à un utilisateur individuels.

```
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"
);
```

Cet exemple montre comment créer une politique liée à un modèle qui utilise le modèle de politique qui donne l'accès à l'éditeur à un utilisateur individuel et à une liste de tâches.

```
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"
);
```

Utilisation des autorisations Amazon Verified avec les fournisseurs d'identité

Une source d'identité est une représentation d'un fournisseur d'identité externe (IdP) dans Amazon Verified Permissions. Les sources d'identité fournissent des informations provenant d'un utilisateur qui s'est authentifié auprès d'un IdP ayant une relation de confiance avec votre magasin de politiques. Lorsque votre application fait une demande d'autorisation à l'aide d'un jeton provenant d'une source d'identité, votre magasin de politiques peut prendre des décisions d'autorisation à partir des propriétés des utilisateurs et des autorisations d'accès. Les sources d'identité Verified Permissions améliorent l'autorisation grâce à une connexion directe à votre magasin d'identité central et à votre service d'authentification.

Vous pouvez utiliser les fournisseurs d'identité OpenID Connect (OIDC) (IdPs) dotés d'autorisations vérifiées. Votre application peut générer des demandes d'autorisation avec une identité OIDC (ID) ou accéder à des jetons Web JSON (JWT). Avec les jetons d'identification, Verified Permissions lit les identifiants utilisateur et les demandes d'attributs en tant que principes du contrôle d'accès basé sur les attributs (ABAC). Avec les jetons d'accès, Verified Permissions lit les identifiants utilisateur en tant que principaux et les autres demandes en tant que contexte. Avec les deux types de jetons, vous pouvez associer une réclamation `groups` à un groupe principal et élaborer des politiques qui évaluent le contrôle d'accès basé sur les rôles (RBAC).

Vous pouvez ajouter un groupe d'utilisateurs Amazon Cognito ou un IdP OpenID Connect (OIDC) personnalisé comme source d'identité.

Rubriques

Utilisation des sources d'identité Amazon Cognito

Verified Permissions travaille en étroite collaboration avec les groupes d'utilisateurs d'Amazon Cognito. Les JWT Amazon Cognito ont une structure prévisible. Verified Permissions reconnaît cette structure et tire le meilleur parti des informations qu'elle contient. Par exemple, vous pouvez implémenter un modèle d'autorisation de contrôle d'accès basé sur les rôles (RBAC) avec des jetons d'identification ou des jetons d'accès.

Une nouvelle source d'identité de pool d'utilisateurs Amazon Cognito a besoin des informations suivantes :

- Le Région AWS.
- ID du groupe d'utilisateurs.
- Le type d'entité utilisateur que vous souhaitez associer à votre source d'identité, par exemple`MyCorp::User`.
- Le type d'entité de groupe que vous souhaitez associer à votre source d'identité, par exemple`MyCorp::UserGroup`.
- (Facultatif) Les identifiants clients de votre groupe d'utilisateurs que vous souhaitez autoriser à envoyer des demandes à votre magasin de politiques.

Comme les autorisations vérifiées ne fonctionnent qu'avec les groupes d'utilisateurs Amazon Cognito appartenant au même groupe Compte AWS, vous ne pouvez pas spécifier de source d'identité dans un autre compte. Les autorisations vérifiées définissent le préfixe d'entité (identifiant de source d'identité auquel vous devez faire référence dans les politiques qui agissent sur les principes du groupe d'utilisateurs) à l'ID de votre groupe d'utilisateurs, par exemple. `us-west-2_EXAMPLE`

Les réclamations relatives aux jetons du pool d'utilisateurs peuvent contenir des attributs, des étendues, des groupes, des identifiants clients et des données personnalisées. Les JWT Amazon Cognito peuvent inclure diverses informations susceptibles de contribuer aux décisions d'autorisation dans les autorisations vérifiées. Il s'agit des licences suivantes :

1. Nom d'utilisateur et réclamations groupées avec un `cognito:` préfixe
2. Attributs utilisateur personnalisés avec `custom: prefix`
3. Réclamations personnalisées ajoutées au moment de l'exécution
4. Les allégations standard de l'OIDC telles que `sub` et `email`

Nous abordons ces réclamations en détail et expliquons comment les gérer dans les politiques d'autorisations vérifiées, dansUtilisation des sources d'identité dans les schémas et les politiques.

> **Important**
>
> Bien que vous puissiez révoquer les jetons Amazon Cognito avant leur expiration, les JWT sont considérés comme des ressources apatrides dotées d'une signature et d'une validité autonomes. Les services conformes au jeton Web JSON RFC 7519 sont censés valider les jetons à distance et ne sont pas tenus de les valider auprès de l'émetteur. Cela signifie qu'il est possible pour les autorisations vérifiées d'accorder l'accès en fonction d'un jeton révoqué ou émis pour l'utilisateur qui a ensuite été supprimé. Pour atténuer ce risque, nous vous recommandons de créer vos jetons avec la durée de validité la plus courte possible et de révoquer les jetons d'actualisation lorsque vous souhaitez supprimer l'autorisation de poursuivre la session d'un utilisateur.

Les politiques de Cedar relatives aux sources d'identité des groupes d'utilisateurs dans Verified Permissions utilisent une syntaxe spéciale pour les noms de demandes contenant des caractères autres que des caractères alphanumériques et un trait de soulignement ()_. Cela inclut les revendications de préfixes du groupe d'utilisateurs qui contiennent un `:` caractère, comme `cognito:username` et`custom:department`. Pour rédiger une condition de politique qui fait référence à la `custom:department` réclamation `cognito:username` ou, écrivez-les respectivement sous la forme `principal["cognito:username"]` et`principal["custom:department"]`.

> **Note**
>
> Si un jeton contient une réclamation avec le `custom:` préfixe `cognito:` ou et un nom de réclamation avec la valeur littérale `cognito` ou`custom`, une demande d'autorisation avec un IsAuthorizedWithTokenéchouera avec un. `ValidationException`

Cet exemple montre comment vous pouvez créer une politique qui fait référence à certaines réclamations des groupes d'utilisateurs Amazon Cognito associées à un mandant.

```
permit(
     principal == ExampleCo::User::"us-east-1_example|4fe90f4a-ref8d9-4033-
a750-4c8622d62fb6",
```

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

Pour plus d'informations sur le mappage des réclamations, consultezAssocier les jetons d'identification au schéma. Pour plus d'informations sur l'autorisation des utilisateurs d'Amazon Cognito, consultez la section Autorisation avec autorisations vérifiées par Amazon dans le guide du développeur Amazon Cognito.

Travailler avec des sources d'identité OIDC

Vous pouvez également configurer n'importe quel IdP OpenID Connect (OIDC) conforme comme source d'identité d'un magasin de politiques. Les fournisseurs OIDC sont similaires aux groupes d'utilisateurs d'Amazon Cognito : ils produisent des JWT comme produit d'authentification. Pour ajouter un fournisseur OIDC, vous devez fournir l'URL de l'émetteur

Une nouvelle source d'identité OIDC nécessite les informations suivantes :

- URL de l'émetteur. Les autorisations vérifiées doivent être en mesure de découvrir un `.well-known/openid-configuration` point de terminaison à cette URL.
- Type de jeton que vous souhaitez utiliser dans les demandes d'autorisation. Dans ce cas, vous avez choisi le jeton d'identité.
- Le type d'entité utilisateur que vous souhaitez associer à votre source d'identité, par exemple`MyCorp::User`.
- Le type d'entité de groupe que vous souhaitez associer à votre source d'identité, par exemple`MyCorp::UserGroup`.
- Exemple de jeton d'identification ou définition des revendications contenues dans le jeton d'identification.
- Préfixe que vous souhaitez appliquer aux ID d'entité d'utilisateur et de groupe. Dans la CLI et l'API, vous pouvez choisir ce préfixe. Dans les magasins de politiques que vous créez à l'aide de l'option Set up with API Gateway and an identity source ou de l'option de configuration guidée, Verified Permissions attribue un préfixe au nom de l'émetteur moins`https://`, par exemple. `MyCorp::User::"auth.example.com|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"`

L'autorisation avec les sources d'identité OIDC utilise les mêmes opérations d'API que les sources d'identité du pool d'utilisateurs : IsAuthorizedWithTokenet BatchIsAuthorizedWithToken.

Cet exemple montre comment vous pouvez créer une politique qui autorise l'accès aux rapports de fin d'année aux employés du service de comptabilité, qui ont une classification confidentielle et qui ne travaillent pas dans un bureau satellite. Verified Permissions dérive ces attributs des allégations contenues dans le jeton d'identification du principal.

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

Validation du client et du public

Lorsque vous ajoutez une source d'identité à un magasin de politiques, Verified Permissions propose des options de configuration qui vérifient que les jetons d'identification et d'accès sont utilisés comme prévu. Cette validation intervient lors du traitement `IsAuthorizedWithToken` des demandes `BatchIsAuthorizedWithToken` d'API. Le comportement diffère entre les jetons d'identification et d'accès, et entre les sources d'identité Amazon Cognito et OIDC. Avec les fournisseurs de groupes d'utilisateurs Amazon Cognito, Verified Permissions peut valider l'ID client à la fois sous forme d'ID et de jetons d'accès. Avec les fournisseurs OIDC, Verified Permissions peut valider l'identifiant du client sous forme de jetons d'identification et l'audience sous forme de jetons d'accès.

Un ID client est un identifiant associé à une application OAuth ou OIDC configurée avec le fournisseur, par exemple. `1example23456789` Une audience est un chemin d'URL associé à la partie utilisatrice prévue, ou à la destination, de l'application cible, par exemple`https://myapplication.example.com`. La `aud` réclamation n'est pas toujours associée au public.

Verified Permissions effectue la validation de l'identité, de la source, de l'audience et du client comme suit :

Amazon Cognito

Les jetons Amazon Cognito ID comportent une `aud` réclamation contenant l'ID client de l'application. Les jetons d'accès ont une `client_id` réclamation qui contient également l'ID du client de l'application.

Lorsque vous entrez une ou plusieurs valeurs pour la validation d'une application cliente dans votre source d'identité, Verified Permissions compare cette liste d'identifiants clients d'applications à la demande de jeton d'identification ou à la `aud` demande de jeton `client_id` d'accès. Les autorisations vérifiées ne valident pas l'URL d'une audience tierce pour les sources d'identité Amazon Cognito.

OIDC

Les jetons d'identification OIDC ont une `aud` réclamation qui contient une liste d'identifiants clients. Les jetons d'accès ont une `aud` réclamation qui contient l'URL d'audience associée au jeton. Les jetons d'accès comportent également une `client_id` réclamation contenant l'ID client prévu.

Vous pouvez saisir une ou plusieurs valeurs pour la validation de l'audience auprès d'un fournisseur OIDC. Lorsque vous choisissez un jeton d'identification de type jeton, Verified Permissions valide l'identifiant du client, en vérifiant qu'au moins un des identifiants client figurant dans la `aud` réclamation correspond à une valeur de validation d'audience.

Les autorisations vérifiées valident l'audience pour les jetons d'accès, en vérifiant que la `aud` réclamation correspond à une valeur de validation de l'audience. La valeur du jeton d'accès provient principalement de la `aud` réclamation, mais peut provenir de la `client_id` réclamation `cid` ou si aucune `aud` réclamation n'existe. Vérifiez auprès de votre IdP la bonne déclaration d'audience et le bon format.

Un exemple de valeur de validation d'audience du jeton d'identification est`1example23456789`.

Un exemple de valeur de validation d'audience du jeton d'accès est`https://myapplication.example.com`.

Autorisation côté client pour les JWT

Vous souhaiterez peut-être traiter les jetons Web JSON dans votre application et transmettre leurs demandes à Verified Permissions sans utiliser de source d'identité de magasin de politiques.

Vous pouvez extraire les attributs de votre entité d'un jeton Web JSON (JWT) et les analyser en autorisations vérifiées.

Cet exemple montre comment vous pouvez appeler Verified Permissions depuis un IDC. [1]

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

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

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

    return authResult;
```

```
}

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

  });

  let entityItem = {
    attributes: attributes,
    identifier: {
      entityType: "PhotoFlash::User",
```

```
      entityId: payload["sub"], // the application need to use the claim that
 represents the user-id
    }
  };
  return [entityItem];
}
```

[1] Cet exemple de code utilise la bibliothèque aws-jwt-verify pour vérifier les JWT signés par OIDC compatible. IdPs

Création de sources d'identité Amazon Verified Permissions

La procédure suivante ajoute une source d'identité à un magasin de politiques existant. Après avoir ajouté votre source d'identité, vous devez ajouter des attributs à votre schéma.

Vous pouvez également créer une source d'identité lorsque vous créez un nouveau magasin de politiques dans la console Verified Permissions. Au cours de ce processus, vous pouvez importer automatiquement les revendications contenues dans vos jetons de source d'identité dans les attributs des entités. Choisissez l'option Configuration guidée ou Configuration avec API Gateway et un fournisseur d'identité. Ces options créent également des politiques initiales.

> **Note**
>
> Les sources d'identité ne sont pas disponibles dans le volet de navigation de gauche tant que vous n'avez pas créé un magasin de politiques. Les sources d'identité que vous créez sont associées au magasin de politiques actuel.

Vous pouvez omettre le type d'entité principal lorsque vous créez une source d'identité avec create-identity-source dans l'API AWS CLI ou CreateIdentity Source dans l'API Verified Permissions. Cependant, un type d'entité vide crée une source d'identité avec un type d'entité de`AWS::Cognito`. Ce nom d'entité n'est pas compatible avec le schéma Policy Store. Pour intégrer les identités Amazon Cognito à votre schéma de magasin de politiques, vous devez définir le type d'entité principal sur une entité de magasin de politiques prise en charge.

Rubriques

- Source d'identité Amazon Cognito
- Source d'identité OIDC

Source d'identité Amazon Cognito

AWS Management Console

Pour créer une source d'identité pour un pool d'utilisateurs Amazon Cognito

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Identity sources.
3. Choisissez Créer une source d'identité.
4. Dans Détails du groupe d'utilisateurs Cognito, sélectionnez Région AWS et entrez l'ID du groupe d'utilisateurs pour votre source d'identité.
5. Dans Configuration principale, choisissez un type principal pour la source d'identité. Les identités issues des groupes d'utilisateurs Amazon Cognito connectés seront mappées au type principal sélectionné.
6. Dans Configuration du groupe, sélectionnez Utiliser le groupe Cognito si vous souhaitez mapper la réclamation du groupe d'utilisateurs. `cognito:groups` Choisissez un type d'entité parent du type principal.
7. Dans Validation des applications clientes, indiquez si vous souhaitez valider les identifiants des applications clientes.
   - Pour valider les identifiants d'applications clientes, sélectionnez Accepter uniquement les jetons dont les identifiants d'application client correspondent. Choisissez Ajouter un nouvel ID d'application client pour chaque ID d'application client à valider. Pour supprimer un ID d'application client qui a été ajouté, choisissez Supprimer à côté de l'ID d'application client.
   - Choisissez Ne pas valider les identifiants des applications clientes si vous ne souhaitez pas valider les identifiants des applications clientes.
8. Choisissez Créer une source d'identité.
9. Avant de pouvoir référencer les attributs que vous extrayez des jetons d'identité ou d'accès dans vos politiques Cedar, vous devez mettre à jour votre schéma pour informer Cedar du type de principal créé par votre source d'identité. Cet ajout au schéma doit inclure les attributs auxquels vous souhaitez faire référence dans vos politiques Cedar. Pour plus d'informations sur le mappage des attributs du jeton Amazon Cognito avec les attributs principaux de Cedar, consultez. Utilisation des sources d'identité dans les schémas et les politiques

Lorsque vous créez un magasin de politiques lié à une API, Verified Permissions interroge votre groupe d'utilisateurs pour les attributs utilisateur et crée un schéma dans lequel votre type principal est renseigné avec les attributs du groupe d'utilisateurs.

AWS CLI

Pour créer une source d'identité pour un pool d'utilisateurs Amazon Cognito

Vous pouvez créer une source d'identité à l'aide de l'opération CreateIdentitySource. L'exemple suivant crée une source d'identité qui peut accéder aux identités authentifiées à partir d'un groupe d'utilisateurs Amazon Cognito.

Le `config.txt` fichier suivant contient les détails du groupe d'utilisateurs Amazon Cognito à utiliser par le paramètre --configuration dans la commande. `create-identity-source`

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

Commande :

```
$ 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"
}
```

Avant de pouvoir référencer les attributs que vous extrayez des jetons d'identité ou d'accès dans vos politiques Cedar, vous devez mettre à jour votre schéma pour informer Cedar du type de principal créé par votre source d'identité. Cet ajout au schéma doit inclure les attributs auxquels vous souhaitez faire référence dans vos politiques Cedar. Pour plus d'informations sur le mappage des attributs du jeton Amazon Cognito avec les attributs principaux de Cedar, consultez. Utilisation des sources d'identité dans les schémas et les politiques

Lorsque vous créez un magasin de politiques lié à une API, Verified Permissions interroge votre groupe d'utilisateurs pour les attributs utilisateur et crée un schéma dans lequel votre type principal est renseigné avec les attributs du groupe d'utilisateurs.

Pour plus d'informations sur l'utilisation des jetons d'accès et d'identité Amazon Cognito pour les utilisateurs authentifiés dans Verified Permissions, consultez la section Autorisation avec Amazon Verified Permissions dans le guide du développeur Amazon Cognito.

Source d'identité OIDC

AWS Management Console

Pour créer une source d'identité OpenID Connect (OIDC)

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Identity sources.
3. Choisissez Créer une source d'identité.
4. Choisissez un fournisseur OIDC externe.
5. Dans URL de l'émetteur, entrez l'URL de votre émetteur OIDC. Il s'agit du point de terminaison du service qui fournit le serveur d'autorisation, les clés de signature et d'autres informations sur votre fournisseur, par exemple`https://auth.example.com`. L'URL de votre émetteur doit héberger un document de découverte OIDC à l'adresse. `/.well-known/openid-configuration`
6. Dans Type de jeton, choisissez le type de JWT OIDC que vous souhaitez que votre demande soumette pour autorisation. Pour plus d'informations, consultez Utilisation des sources d'identité dans les schémas et les politiques.
7. Dans Réclamations d'utilisateur et de groupe, choisissez une entité utilisateur et une réclamation utilisateur pour la source d'identité. L'entité utilisateur est une entité de votre

magasin de politiques à laquelle vous souhaitez faire référence aux utilisateurs de votre fournisseur OIDC. La réclamation de l'utilisateur est généralement `sub` une réclamation provenant de votre identifiant ou de votre jeton d'accès qui contient l'identifiant unique de l'entité à évaluer. Les identités de l'IdP OIDC connecté seront mappées au type principal sélectionné.

8. Dans Réclamations d'utilisateur et de groupe, choisissez une entité de groupe et une réclamation de groupe pour la source d'identité. L'entité Group est le parent de l'entité User. Les revendications de groupe sont mappées à cette entité. La réclamation de groupe est généralement `groups` une réclamation provenant de votre identifiant ou de votre jeton d'accès qui contient une chaîne, un JSON ou une chaîne de noms de groupes d'utilisateurs délimitée par des espaces pour l'entité à évaluer. Les identités de l'IdP OIDC connecté seront mappées au type principal sélectionné.
9. Dans Validation de l'audience, entrez les identifiants clients ou les URL d'audience que vous souhaitez que votre magasin de politiques accepte dans les demandes d'autorisation, le cas échéant.
10. Choisissez Créer une source d'identité.
11. Mettez à jour votre schéma pour que Cedar connaisse le type de principal créé par votre source d'identité. Cet ajout au schéma doit inclure les attributs auxquels vous souhaitez faire référence dans vos politiques Cedar. Pour plus d'informations sur le mappage des attributs du jeton Amazon Cognito avec les attributs principaux de Cedar, consultez. Utilisation des sources d'identité dans les schémas et les politiques

    Lorsque vous créez un magasin de politiques lié à une API, Verified Permissions interroge votre groupe d'utilisateurs pour les attributs utilisateur et crée un schéma dans lequel votre type principal est renseigné avec les attributs du groupe d'utilisateurs.

AWS CLI

Pour créer une source d'identité OIDC

Vous pouvez créer une source d'identité à l'aide de l'opération CreateIdentitySource. L'exemple suivant crée une source d'identité qui peut accéder aux identités authentifiées à partir d'un groupe d'utilisateurs Amazon Cognito.

Le `config.txt` fichier suivant contient les détails d'un IdP OIDC à utiliser par `--configuration` le paramètre de `create-identity-source` la commande. Cet exemple crée une source d'identité OIDC pour les jetons d'identification.

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

Le `config.txt` fichier suivant contient les détails d'un IdP OIDC à utiliser par `--configuration` le paramètre de `create-identity-source` la commande. Cet exemple crée une source d'identité OIDC pour les jetons d'accès.

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

Commande :

```
$ 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"
}
```

Avant de pouvoir référencer les attributs que vous extrayez des jetons d'identité ou d'accès dans vos politiques Cedar, vous devez mettre à jour votre schéma pour informer Cedar du type de principal créé par votre source d'identité. Cet ajout au schéma doit inclure les attributs auxquels vous souhaitez faire référence dans vos politiques Cedar. Pour plus d'informations sur le mappage des attributs du jeton Amazon Cognito avec les attributs principaux de Cedar, consultez. Utilisation des sources d'identité dans les schémas et les politiques

Lorsque vous créez un magasin de politiques lié à une API, Verified Permissions interroge votre groupe d'utilisateurs pour les attributs utilisateur et crée un schéma dans lequel votre type principal est renseigné avec les attributs du groupe d'utilisateurs.

Modification des sources d'identité Amazon Verified Permissions

Vous pouvez modifier certains paramètres de votre source d'identité après l'avoir créée. Si le schéma de votre magasin de politiques correspond aux attributs de votre source d'identité, notez que vous devez mettre à jour votre schéma séparément pour refléter les modifications que vous apportez à votre source d'identité.

Rubriques

- Source d'identité des groupes d'utilisateurs Amazon Cognito
- Source d'identité OpenID Connect (OIDC)

Source d'identité des groupes d'utilisateurs Amazon Cognito

AWS Management Console

Pour mettre à jour la source d'identité d'un groupe d'utilisateurs Amazon Cognito

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Identity sources.
3. Choisissez l'ID de la source d'identité à modifier.
4. Choisissez Modifier.
5. Dans Détails du groupe d'utilisateurs Cognito, sélectionnez Région AWS et saisissez l'ID du groupe d'utilisateurs pour votre source d'identité.
6. Dans Détails du principal, vous pouvez mettre à jour le type principal pour la source d'identité. Les identités issues des groupes d'utilisateurs Amazon Cognito connectés seront mappées au type principal sélectionné.
7. Dans Configuration du groupe, sélectionnez Utiliser le groupe Cognito si vous souhaitez mapper la réclamation du groupe d'utilisateurs. `cognito:groups` Choisissez un type d'entité parent du type principal.
8. Dans Validation des applications clientes, indiquez si vous souhaitez valider les identifiants des applications clientes.
   - Pour valider les identifiants d'applications clientes, sélectionnez Accepter uniquement les jetons dont les identifiants d'application client correspondent. Choisissez Ajouter un nouvel ID d'application client pour chaque ID d'application client à valider. Pour supprimer un ID d'application client qui a été ajouté, choisissez Supprimer à côté de l'ID d'application client.
   - Choisissez Ne pas valider les identifiants des applications clientes si vous ne souhaitez pas valider les identifiants des applications clientes.
9. Sélectionnez Enregistrer les modifications.
10. Si vous avez modifié le type principal de la source d'identité, vous devez mettre à jour votre schéma pour qu'il reflète correctement le type principal mis à jour.

Vous pouvez supprimer une source d'identité en cliquant sur le bouton radio à côté d'une source d'identité, puis en choisissant Supprimer la source d'identité. Tapez `delete` dans la zone de

texte, puis choisissez Supprimer la source d'identité pour confirmer la suppression de la source d'identité.

AWS CLI

Pour mettre à jour la source d'identité d'un groupe d'utilisateurs Amazon Cognito

Vous pouvez mettre à jour une source d'identité à l'aide de l'opération UpdateIdentitySource. L'exemple suivant met à jour la source d'identité spécifiée pour utiliser un autre groupe d'utilisateurs Amazon Cognito.

Le `config.txt` fichier suivant contient les détails du groupe d'utilisateurs Amazon Cognito à utiliser par le paramètre --configuration dans la commande. `create-identity-source`

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

Commande :

```
$ 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"
}
```

Si vous modifiez le type principal de la source d'identité, vous devez mettre à jour votre schéma pour qu'il reflète correctement le type principal mis à jour.

Source d'identité OpenID Connect (OIDC)

AWS Management Console

Pour mettre à jour une source d'identité OIDC

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Identity sources.
3. Choisissez l'ID de la source d'identité à modifier.
4. Choisissez Modifier.
5. Dans les détails du fournisseur OIDC, modifiez l'URL de l'émetteur selon vos besoins.
6. Dans Map token claims to schema attributes, modifiez les associations entre les revendications d'utilisateur et de groupe et les types d'entités du policy store, selon les besoins. Après avoir modifié les types d'entités, vous devez mettre à jour vos politiques et les attributs de schéma pour les appliquer aux nouveaux types d'entités.
7. Dans Validation de l'audience, ajoutez ou supprimez les valeurs d'audience que vous souhaitez appliquer.
8. Sélectionnez Enregistrer les modifications.

Vous pouvez supprimer une source d'identité en cliquant sur le bouton radio à côté d'une source d'identité, puis en choisissant Supprimer la source d'identité. Tapez `delete` dans la zone de texte, puis choisissez Supprimer la source d'identité pour confirmer la suppression de la source d'identité.

AWS CLI

Pour mettre à jour une source d'identité OIDC

Vous pouvez mettre à jour une source d'identité à l'aide de l'opération UpdateIdentitySource. L'exemple suivant met à jour la source d'identité spécifiée pour utiliser un autre fournisseur OIDC.

Le `config.txt` fichier suivant contient les détails du groupe d'utilisateurs Amazon Cognito à utiliser par le paramètre --configuration dans la commande. `create-identity-source`

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

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

Commande :

```
$ 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"
}
```

Si vous modifiez le type principal de la source d'identité, vous devez mettre à jour votre schéma pour qu'il reflète correctement le type principal mis à jour.

Utilisation des sources d'identité dans les schémas et les politiques

Vous souhaiterez peut-être ajouter une source d'identité à un magasin de politiques et mapper les revendications du fournisseur dans le schéma de votre magasin de politiques. Vous pouvez automatiser ce processus ou mettre à jour votre schéma manuellement. Cette section du guide de l'utilisateur contient les informations suivantes :

- Quand vous pouvez renseigner automatiquement les attributs d'un schéma de magasin de politiques
- Comment utiliser les demandes de jetons Amazon Cognito et OIDC dans vos politiques d'autorisations vérifiées

- Comment créer manuellement un schéma pour une source d'identité

Les magasins de politiques liés à l'API et les magasins de politiques dotés d'une source d'identité via la configuration guidée ne nécessitent pas de mappage manuel des attributs des jetons d'identité (ID) avec le schéma. Vous pouvez fournir des autorisations vérifiées avec les attributs de votre groupe d'utilisateurs ou des jetons OIDC et créer un schéma rempli d'attributs utilisateur. Dans l'autorisation par jeton d'identification, Verified Permissions associe les revendications aux attributs d'une entité principale. Vous devrez peut-être mapper manuellement les jetons Amazon Cognito à votre schéma dans les conditions suivantes :

- Vous avez créé un magasin de politiques vide ou un magasin de politiques à partir d'un échantillon.
- Vous souhaitez étendre votre utilisation des jetons d'accès au-delà du contrôle d'accès basé sur les rôles (RBAC).
- Vous créez des magasins de politiques à l'aide de l'API REST Verified Permissions, d'un AWS SDK ou du AWS CDK.

Pour utiliser Amazon Cognito ou un fournisseur d'identité OIDC (IdP) comme source d'identité dans votre magasin de politiques d'autorisations vérifiées, vous devez avoir des attributs de fournisseur dans votre schéma. Si vous avez créé votre magasin de politiques de manière à remplir automatiquement votre schéma à partir des informations du fournisseur contenues dans un jeton d'identification, vous êtes prêt à écrire des politiques. Si vous créez un magasin de politiques sans schéma pour votre source d'identité, vous devez ajouter des attributs de fournisseur au schéma. Votre schéma doit correspondre aux entités créées par les jetons du fournisseur IsAuthorizedWithTokenou dans les demandes d'API BatchIsAuthorizedWithToken. Vous pouvez ensuite écrire des politiques à l'aide des attributs du jeton du fournisseur.

Pour plus d'informations sur l'utilisation de l'identifiant Amazon Cognito et des jetons d'accès pour les utilisateurs authentifiés dans le cadre des autorisations vérifiées, consultez la section Autorisation avec autorisations vérifiées Amazon dans le guide du développeur Amazon Cognito.

Rubriques

- Ce qu'il faut savoir sur le mappage de schémas
- Associer les jetons d'identification au schéma
- Cartographie des jetons d'accès
- Notation alternative pour les demandes délimitées par des deux-points sur Amazon Cognito

Ce qu'il faut savoir sur le mappage de schémas

Le mappage des attributs diffère selon les types de jetons

Dans l'autorisation du jeton d'accès, Verified Permissions associe les revendications au contexte. Dans l'autorisation par jeton d'identification, Verified Permissions associe les revendications aux attributs principaux. Pour les magasins de politiques que vous créez dans la console Verified Permissions, seuls les magasins de politiques vides et d'exemple ne vous laissent aucune source d'identité et vous obligent à renseigner votre schéma avec les attributs du groupe d'utilisateurs pour l'autorisation par jeton d'identification. L'autorisation des jetons d'accès est basée sur le contrôle d'accès basé sur les rôles (RBAC) avec les demandes d'adhésion à un groupe et n'associe pas automatiquement les autres revendications au schéma du magasin de politiques.

Les attributs de source d'identité ne sont pas obligatoires

Lorsque vous créez une source d'identité dans la console Verified Permissions, aucun attribut n'est marqué comme obligatoire. Cela évite que les demandes manquantes ne provoquent des erreurs de validation dans les demandes d'autorisation. Vous pouvez définir les attributs comme obligatoires selon vos besoins, mais ils doivent être présents dans toutes les demandes d'autorisation.

Le RBAC ne nécessite pas d'attributs dans le schéma

Les schémas des sources d'identité dépendent des associations d'entités que vous créez lorsque vous ajoutez votre source d'identité. Une source d'identité associe une réclamation à un type d'entité utilisateur et une réclamation à un type d'entité de groupe. Ces mappages d'entités sont au cœur d'une configuration identité-source. Avec ces informations minimales, vous pouvez rédiger des politiques qui exécutent des actions d'autorisation pour des utilisateurs spécifiques et des groupes spécifiques dont les utilisateurs peuvent être membres, dans un modèle de contrôle d'accès basé sur les rôles (RBAC). L'ajout de revendications de jetons au schéma étend le champ d'autorisation de votre magasin de politiques. Les attributs utilisateur issus des jetons d'identification contiennent des informations sur les utilisateurs qui peuvent contribuer à l'autorisation du contrôle d'accès basé sur les attributs (ABAC). Les attributs contextuels des jetons d'accès contiennent des informations telles que les étendues OAuth 2.0 qui peuvent fournir des informations de contrôle d'accès supplémentaires de la part de votre fournisseur, mais nécessitent des modifications de schéma supplémentaires.

Les options Set up with API Gateway and a identity source et Guided setup de la console Verified Permissions attribuent des droits de jeton d'identification au schéma. Ce n'est pas le cas pour les demandes de jetons d'accès. Pour ajouter des revendications de jeton d'accès non groupé à votre

schéma, vous devez modifier votre schéma en mode JSON et ajouter des attributs CommonTypes. Pour plus d'informations, consultez Cartographie des jetons d'accès.

Les groupes OIDC affirment prendre en charge plusieurs formats

Lorsque vous ajoutez un fournisseur OIDC, vous pouvez choisir le nom de la réclamation du groupe sous forme d'identifiant ou de jetons d'accès que vous souhaitez associer à l'appartenance au groupe d'un utilisateur dans votre magasin de politiques. Les autorisations vérifiées reconnaissent les demandes de groupes dans les formats suivants :

1. Chaîne sans espaces : `"groups": "MyGroup"`
2. Liste délimitée par des espaces :. `"groups": "MyGroup1 MyGroup2 MyGroup3"` Chaque chaîne est un groupe.
3. Liste JSON (séparée par des virgules) : `"groups": ["MyGroup1", "MyGroup2", "MyGroup3"]`

> **Note**
>
> Verified Permissions interprète chaque chaîne d'une réclamation de groupe séparée par des espaces comme un groupe distinct. Pour interpréter un nom de groupe comportant un espace comme un groupe unique, remplacez ou supprimez l'espace dans la réclamation. Par exemple, formatez un groupe nommé `My Group` comme`MyGroup`.

Choisissez un type de jeton

La façon dont votre magasin de politiques fonctionne avec votre source d'identité dépend d'une décision clé en matière de configuration de la source d'identité : traiter les identifiants ou les jetons d'accès. Avec un fournisseur d'identité Amazon Cognito, vous pouvez choisir le type de jeton lorsque vous créez un magasin de politiques lié à une API. Lorsque vous créez un magasin de politiques lié à une API, vous devez choisir si vous souhaitez configurer l'autorisation pour les jetons d'identification ou d'accès. Ces informations affectent les attributs de schéma que Verified Permissions applique à votre magasin de politiques, ainsi que la syntaxe de l'autorisateur Lambda pour votre API API Gateway. Avec un fournisseur OIDC, vous devez choisir un type de jeton lorsque vous ajoutez la source d'identité. Vous pouvez choisir un identifiant ou un jeton d'accès, et votre choix exclut le type de jeton non choisi du traitement dans votre magasin de polices. En particulier, si vous souhaitez bénéficier du mappage automatique des demandes de jeton d'identification aux attributs

dans la console Verified Permissions, déterminez rapidement le type de jeton que vous souhaitez traiter avant de créer votre source d'identité. La modification du type de jeton nécessite des efforts considérables pour refactoriser vos politiques et votre schéma. Les rubriques suivantes décrivent l'utilisation des jetons d'identification et d'accès avec les magasins de politiques.

L'analyseur Cedar nécessite des crochets pour certains caractères

Les politiques font généralement référence aux attributs du schéma dans un format tel que`principal.username`. Dans le cas de la plupart des caractères non alphanumériques tels `/` que `:.`, ou susceptibles d'apparaître dans les noms de réclamation de jetons, Verified Permissions ne peut pas analyser une valeur de condition telle que ou. `principal.cognito:groups` `context.ip-address` Vous devez plutôt mettre en forme ces conditions avec une notation entre crochets dans le format `principal["cognito:username"]` ou`context["ip-address"]`, respectivement. Le caractère de soulignement _ est un caractère valide dans les noms des demandes et constitue la seule exception non alphanumérique à cette exigence.

Voici un exemple de schéma partiel pour un attribut principal de ce type :

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

Voici un exemple de schéma partiel pour un attribut de contexte de ce type :

```
"GetOrder": {
```

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

Voici un exemple de politique pour les attributs qui seront validés par rapport à ce schéma :

```
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.*"
};
```

Associer les jetons d'identification au schéma

Verified Permissions traite les demandes de jetons d'identification en tant qu'attributs de l'utilisateur : ses noms et titres, son appartenance à un groupe, ses coordonnées. Les jetons d'identification sont particulièrement utiles dans un modèle d'autorisation de contrôle d'accès basé sur les attributs (ABAC). Lorsque vous souhaitez que les autorisations vérifiées analysent l'accès aux ressources en fonction de l'auteur de la demande, choisissez des jetons d'identification pour votre source d'identité.

Jetons d'identification Amazon Cognito

Les jetons d'identification Amazon Cognito fonctionnent avec la plupart des bibliothèques dépendantes OIDC. Ils étendent les fonctionnalités de l'OIDC avec des allégations supplémentaires. Votre application peut authentifier l'utilisateur à l'aide des opérations de l'API d'authentification des groupes d'utilisateurs Amazon Cognito ou à l'aide de l'interface utilisateur hébergée par les groupes d'utilisateurs. Pour plus d'informations, consultez la section Utilisation de l'API et des points de terminaison dans le manuel Amazon Cognito Developer Guide.

Réclamations utiles concernant les jetons d'identification Amazon Cognito

`cognito:username` et *`preferred_username`*

Variantes du nom d'utilisateur de l'utilisateur.

`sub`

Identifiant utilisateur unique (UUID) de l'utilisateur

Réclamations comportant un *`custom:`* préfixe

Un préfixe pour les attributs personnalisés du groupe d'utilisateurs tels que`custom:employmentStoreCode`.

Réclamations standard

Les allégations standard de l'OIDC telles que `email` et. `phone_number` Pour plus d'informations, consultez la section Réclamations standard d'OpenID Connect Core 1.0 incorporant le jeu d'errata 2.

`cognito:groups`

Appartenances à un groupe d'utilisateurs. Dans un modèle d'autorisation basé sur le contrôle d'accès basé sur les rôles (RBAC), cette affirmation présente les rôles que vous pouvez évaluer dans vos politiques.

Réclamations transitoires

Réclamations qui ne sont pas une propriété de l'utilisateur, mais qui sont ajoutées au moment de l'exécution par un groupe d'utilisateurs. Déclencheur Lambda pré-génération de jetons. Les réclamations transitoires ressemblent aux réclamations standard mais ne sont pas conformes à la norme, par exemple `tenant` ou`department`.

Dans les politiques qui font référence à des attributs Amazon Cognito dotés d'un `:` séparateur, référencez les attributs au format. `principal["cognito:username"]` La revendication des rôles `cognito:groups` constitue une exception à cette règle. Verified Permissions associe le contenu de cette réclamation aux entités parentes de l'entité utilisateur.

Pour plus d'informations sur la structure des jetons d'identification issus des groupes d'utilisateurs Amazon Cognito, consultez la section Utilisation du jeton d'identification dans le guide du développeur Amazon Cognito.

L'exemple de jeton d'identification suivant possède chacun des quatre types d'attributs. Elle inclut la réclamation spécifique à Amazon Cognito`cognito:username`, la réclamation personnalisée`custom:employmentStoreCode`, la réclamation standard et la réclamation `email` transitoire. `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"
}
```

Lorsque vous créez une source d'identité avec votre groupe d'utilisateurs Amazon Cognito, vous spécifiez le type d'entité principale que Verified Permissions génère dans les demandes d'autorisation. `IsAuthorizedWithToken` Vos politiques peuvent ensuite tester les attributs de ce

principal dans le cadre de l'évaluation de cette demande. Votre schéma définit le type et les attributs principaux d'une source d'identité, puis vous pouvez les référencer dans vos politiques Cedar.

Vous spécifiez également le type d'entité de groupe que vous souhaitez obtenir à partir de la réclamation des groupes de jetons d'identification. Dans les demandes d'autorisation, Verified Permissions associe chaque membre du groupe à ce type d'entité de groupe. Dans les politiques, vous pouvez faire référence à cette entité de groupe en tant que principale.

L'exemple suivant montre comment refléter les attributs de l'exemple de jeton d'identité dans votre schéma d'autorisations vérifiées. Pour plus d'informations sur la modification de votre schéma, consultezModification de schémas en mode JSON. Si la configuration de votre source d'identité spécifie le type principal`User`, vous pouvez inclure quelque chose de similaire à l'exemple suivant pour mettre ces attributs à la disposition de 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
         }
      }
   }
}
```

Après avoir mis à jour votre schéma pour refléter les attributs Amazon Cognito, vous pouvez créer des politiques qui font référence à ces attributs.

```
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"
};
```

Jetons d'identification OIDC

L'utilisation des jetons d'identification d'un fournisseur OIDC est similaire à celle des jetons d'identification Amazon Cognito. La différence réside dans les réclamations. Votre IdP peut présenter des attributs OIDC standard ou avoir un schéma personnalisé. Lorsque vous créez un nouveau magasin de politiques dans la console Verified Permissions, vous pouvez ajouter une source d'identité OIDC avec un exemple de jeton d'identification, ou vous pouvez associer manuellement les demandes de jeton aux attributs utilisateur. Comme Verified Permissions ne connaît pas le schéma d'attributs de votre IdP, vous devez fournir ces informations.

Pour plus d'informations, consultez Création de magasins de politiques d'autorisations vérifiées.

Voici un exemple de schéma pour un magasin de politiques avec une source d'identité 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"
         }
      }
   }
}
```

La politique suivante s'applique aux membres d'un groupe de votre fournisseur OIDC.

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

Cartographie des jetons d'accès

Verified Permissions traite les demandes de jetons d'accès autres que celles revendiquées par les groupes en tant qu'attributs de l'action ou en tant qu'attributs contextuels. Outre l'appartenance à un groupe, les jetons d'accès de votre IdP peuvent contenir des informations sur l'accès aux API. Les jetons d'accès sont utiles dans les modèles d'autorisation qui utilisent le contrôle d'accès basé sur les rôles (RBAC). Les modèles d'autorisation qui reposent sur des revendications de jetons d'accès autres que l'appartenance à un groupe nécessitent des efforts supplémentaires pour configurer le schéma.

Cartographie des jetons d'accès Amazon Cognito

Les jetons d'accès Amazon Cognito contiennent des revendications qui peuvent être utilisées à des fins d'autorisation :

Réclamations utiles concernant les jetons d'accès Amazon Cognito

`client_id`

L'ID de l'application cliente d'une partie utilisatrice de l'OIDC. À l'aide de l'ID client, Verified Permissions peut vérifier que la demande d'autorisation provient d'un client autorisé pour le magasin de politiques. Dans le machine-to-machine cadre de l'autorisation (M2M), le système demandeur autorise une demande avec un secret client et fournit l'identifiant du client et les champs d'application comme preuve d'autorisation.

scope

Les étendues OAuth 2.0 qui représentent les autorisations d'accès du porteur du jeton.

cognito:groups

Appartenances à un groupe d'utilisateurs. Dans un modèle d'autorisation basé sur le contrôle d'accès basé sur les rôles (RBAC), cette affirmation présente les rôles que vous pouvez évaluer dans vos politiques.

Réclamations transitoires

Réclamations qui ne constituent pas une autorisation d'accès, mais qui sont ajoutées au moment de l'exécution par un groupe d'utilisateurs. Déclencheur Lambda avant la génération de jetons. Les réclamations transitoires ressemblent aux réclamations standard mais ne sont pas conformes à la norme, par exemple `tenant` ou`department`. La personnalisation des jetons d'accès augmente le coût de votre AWS facture.

Pour plus d'informations sur la structure des jetons d'accès issus des groupes d'utilisateurs Amazon Cognito, consultez la section Utilisation du jeton d'accès dans le manuel Amazon Cognito Developer Guide.

Un jeton d'accès Amazon Cognito est mappé à un objet de contexte lorsqu'il est transmis à Verified Permissions. Les attributs du jeton d'accès peuvent être référencés à l'aide de`context.token.`*`attribute_name`*. L'exemple de jeton d'accès suivant inclut à la fois les `scope` revendications `client_id` et.

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

```
    "iat": 1688092966,
    "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN2222222",
    "username": "alice"
}
```

L'exemple suivant montre comment refléter les attributs de l'exemple de jeton d'accès dans votre schéma d'autorisations vérifiées. Pour plus d'informations sur la modification de votre schéma, consultezModification de schémas en mode JSON.

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

```
                }
            }
        },
        "type": "Record"
    }
  }
 }
}
```

Après avoir mis à jour votre schéma pour refléter les attributs Amazon Cognito, vous pouvez créer des politiques qui font référence à ces attributs.

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

Cartographie des jetons d'accès OIDC

La plupart des jetons d'accès provenant de fournisseurs OIDC externes sont étroitement liés aux jetons d'accès Amazon Cognito. Un jeton d'accès OIDC est mappé à un objet de contexte lorsqu'il est transmis à Verified Permissions. Les attributs du jeton d'accès peuvent être référencés à l'aide de`context.token.`*`attribute_name`*. L'exemple de jeton d'accès OIDC suivant inclut des exemples de revendications de base.

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

```
}
```

L'exemple suivant montre comment refléter les attributs de l'exemple de jeton d'accès dans votre schéma d'autorisations vérifiées. Pour plus d'informations sur la modification de votre schéma, consultezModification de schémas en mode JSON.

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

```
            "type": "Record"
         }
      }
   }
}
```

Après avoir mis à jour votre schéma pour refléter les attributs de l'IdP, vous pouvez créer des politiques qui font référence aux attributs.

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

Notation alternative pour les demandes délimitées par des deux-points sur Amazon Cognito

Au moment du lancement de Verified Permissions, le schéma recommandé pour le jeton Amazon Cognito prétendait aimer `cognito:groups` et `custom:store` convertir ces chaînes séparées par des points pour utiliser le `.` caractère comme séparateur hiérarchique. Ce format est appelé notation par points. Par exemple, une référence à `cognito:groups` est devenue `principal.cognito.groups` dans vos politiques. Bien que vous puissiez continuer à utiliser ce format, nous vous recommandons de créer votre schéma et vos politiques avec la notation entre crochets. Dans ce format, une référence à `cognito:groups` apparaît `principal["cognito:groups"]` dans vos politiques. Les schémas générés automatiquement pour les jetons d'identification du groupe d'utilisateurs à partir de la console Verified Permissions utilisent la notation entre crochets.

Vous pouvez continuer à utiliser la notation par points dans le schéma et les politiques créés manuellement pour les sources d'identité Amazon Cognito. Vous ne pouvez pas utiliser la notation par points `:` ou tout autre caractère non alphanumérique dans le schéma ou les politiques pour tout autre type d'IdP OIDC.

Un schéma de notation par points imbrique chaque instance d'un `:` caractère en tant qu'enfant de la phrase `custom` initiale `cognito` ou de la phrase initiale, comme le montre l'exemple suivant :

```
"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
         }
      }
   }
}
```

Avec un schéma de ce format, vous pouvez créer une politique avec une notation par points, comme dans l'exemple suivant :

```
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"
};
```

Conception d'un modèle d'autorisation pour votre application

Lorsque vous vous préparez à utiliser le service Amazon Verified Permissions dans une application logicielle, il peut être difficile de se lancer immédiatement dans la rédaction de déclarations de politique dans un premier temps. Cela reviendrait à commencer le développement d'autres parties d'une application en écrivant des instructions SQL ou des spécifications d'API avant de décider complètement de ce que l'application doit faire. Vous devriez plutôt commencer par une expérience utilisateur, en comprenant clairement ce que les utilisateurs finaux devraient voir lors de la gestion des autorisations dans l'interface utilisateur de l'application. Ensuite, revenez en arrière à partir de cette expérience pour arriver à une approche de mise en œuvre.

Au cours de ce travail, vous vous poserez des questions telles que :

- Quelles sont mes ressources ? Ont-ils des relations les uns avec les autres ? Par exemple, les fichiers se trouvent-ils dans un dossier ?
- Quelles actions les directeurs peuvent-ils effectuer sur chaque ressource ?
- Comment les directeurs obtiennent-ils ces autorisations ?
- Voulez-vous que vos utilisateurs finaux puissent choisir parmi des autorisations prédéfinies telles que « Administrateur », « Opérateur » ou « ReadOnly », ou devraient-ils créer des déclarations de politique ad hoc ? Ou les deux ?
- Les autorisations doivent-elles être transmises à toutes les ressources, telles que les fichiers héritant des autorisations d'un dossier parent ?
- Quels types de requêtes sont nécessaires pour améliorer l'expérience utilisateur ? Par exemple, devez-vous répertorier toutes les ressources auxquelles un directeur peut accéder pour afficher la page d'accueil de cet utilisateur ?
- Les utilisateurs peuvent-ils accidentellement se priver de leurs propres ressources ? Cela doit-il être évité ?

Le résultat final de cet exercice est appelé modèle d'autorisation ; il définit les principes, les ressources, les actions et la façon dont ils sont liés les uns aux autres. La production de ce modèle ne nécessite aucune connaissance unique de Cedar ou du service Verified Permissions. Il s'agit avant tout d'un exercice de conception de l'expérience utilisateur, un peu comme les autres, qui peut se traduire par des artefacts tels que des maquettes d'interface, des diagrammes logiques et une description globale de la manière dont les autorisations influencent ce que les utilisateurs voient dans le produit. Cedar est conçu pour être suffisamment flexible pour répondre aux besoins des clients

selon un modèle, plutôt que de forcer le modèle à se plier de manière anormale pour se conformer à la mise en œuvre d'un modèle Cedar. Par conséquent, une compréhension précise de l'expérience utilisateur souhaitée est le meilleur moyen de parvenir à un modèle optimal.

Cette section fournit des conseils généraux sur la manière d'aborder l'exercice de conception, les points à surveiller et un ensemble de bonnes pratiques pour utiliser correctement les autorisations vérifiées.

Outre les directives présentées ici, n'oubliez pas de prendre en compte les meilleures pratiques du guide de référence sur le langage politique de Cedar.

Rubriques

Il n'existe pas de modèle canonique « correct »

Lorsque vous concevez un modèle d'autorisation, il n'existe pas de réponse unique et correcte. Différentes applications peuvent utiliser efficacement différents modèles d'autorisation pour des concepts similaires, et c'est OK. Par exemple, considérez la représentation du système de fichiers d'un ordinateur. Lorsque vous créez un fichier dans un système d'exploitation de type Unix, il n'hérite pas automatiquement des autorisations du dossier parent. En revanche, dans de nombreux autres systèmes d'exploitation et dans la plupart des services de partage de fichiers en ligne, les fichiers héritent des autorisations de leur dossier parent. Les deux choix sont valides en fonction des circonstances pour lesquelles l'application est optimisée.

L'exactitude d'une solution d'autorisation n'est pas absolue, mais elle doit être considérée en fonction de la manière dont elle fournit l'expérience souhaitée par vos clients et de la manière dont elle protège leurs ressources comme ils l'attendent. Si votre modèle d'autorisation tient ses promesses, cela signifie qu'il est efficace.

C'est pourquoi commencer votre conception avec l'expérience utilisateur souhaitée est la condition préalable la plus utile à la création d'un modèle d'autorisation efficace.

Concentrez-vous sur vos ressources au-delà des opérations d'API

Dans la plupart des applications destinées aux consommateurs, les autorisations sont calquées sur les ressources prises en charge par l'application. Par exemple, une application de partage de fichiers peut représenter les autorisations comme des actions pouvant être effectuées sur un fichier ou un dossier. Il s'agit d'un bon modèle simple qui fait abstraction de l'implémentation sous-jacente et des opérations de l'API principale.

En revanche, d'autres types d'applications, en particulier les services Web, conçoivent fréquemment des autorisations en fonction des opérations d'API elles-mêmes. Par exemple, si un service Web fournit une API nommée`createThing()`, le modèle d'autorisation peut définir une autorisation correspondante, ou une autorisation nommée `action` dans Cedar`createThing`. Cela fonctionne dans de nombreuses situations et facilite la compréhension des autorisations. Pour appeler l'`createThing`opération, vous devez disposer de l'autorisation `createThing` d'action. Cela semble simple, non ?

Vous constaterez que le processus de démarrage dans la console des autorisations vérifiées inclut la possibilité de créer vos ressources et vos actions directement à partir d'une API. Il s'agit d'une base de référence utile : un mappage direct entre votre magasin de politiques et l'API qu'il autorise.

Cependant, cette approche axée sur les API peut être loin d'être optimale, car les API ne sont qu'un proxy pour ce que vos clients essaient réellement de protéger : les données et les ressources sous-jacentes. Si plusieurs API contrôlent l'accès aux mêmes ressources, il peut être difficile pour les administrateurs de raisonner sur les chemins d'accès à ces ressources et de gérer l'accès en conséquence.

Prenons l'exemple d'un annuaire d'utilisateurs qui contient les membres d'une organisation. Les utilisateurs peuvent être organisés en groupes, et l'un des objectifs de sécurité est d'empêcher les parties non autorisées de découvrir leur appartenance à un groupe. Le service qui gère ce répertoire d'utilisateurs fournit deux opérations d'API :

- `listMembersOfGroup`
- `listGroupMembershipsForUser`

Les clients peuvent utiliser l'une ou l'autre de ces opérations pour découvrir l'appartenance à un groupe. Par conséquent, l'administrateur des autorisations doit se rappeler de coordonner l'accès aux deux opérations. Cela est encore plus compliqué si vous choisissez ultérieurement d'ajouter une nouvelle opération d'API pour traiter des cas d'utilisation supplémentaires, tels que les suivants.

- `isUserInGroups`(une nouvelle API pour tester rapidement si un utilisateur appartient à un ou plusieurs groupes)

Du point de vue de la sécurité, cette API ouvre une troisième voie pour découvrir les appartenances à des groupes, perturbant ainsi les autorisations soigneusement conçues de l'administrateur.

Nous vous recommandons d'ignorer la sémantique de l'API et de vous concentrer plutôt sur les données et les ressources sous-jacentes ainsi que sur leurs opérations d'association. L'application de cette approche à l'exemple d'appartenance à un groupe conduirait à une autorisation abstraite`viewGroupMembership`, par exemple, que chacune des trois opérations d'API doit consulter.

Nom d'API	Autorisations
`listMembersOfGroup`	nécessite une `viewGroupMembership` autorisation sur le groupe
`listGroupMembershipsForUser`	nécessite `viewGroupMembership` l'autorisation de l'utilisateur
`isUserInGroups`	nécessite `viewGroupMembership` l'autorisation de l'utilisateur

En définissant cette autorisation unique, l'administrateur contrôle avec succès l'accès à la découverte des appartenances à des groupes, maintenant et pour toujours. En contrepartie, chaque opération d'API doit désormais documenter les différentes autorisations éventuellement requises, et l'administrateur doit consulter cette documentation lors de l'élaboration des autorisations. Cela peut être un compromis valable lorsque cela est nécessaire pour répondre à vos exigences de sécurité.

L'autorisation composée est normale

L'autorisation composée se produit lorsqu'une seule activité utilisateur, telle que le fait de cliquer sur un bouton dans l'interface de votre application, nécessite plusieurs requêtes d'autorisation individuelles pour déterminer si cette activité est autorisée. Par exemple, le déplacement d'un fichier vers un nouveau répertoire d'un système de fichiers peut nécessiter trois autorisations différentes : la possibilité de supprimer un fichier du répertoire source, la possibilité d'ajouter un fichier dans le répertoire de destination et éventuellement la possibilité de toucher le fichier lui-même (selon l'application).

Si vous débutez dans la conception d'un modèle d'autorisation, vous pensez peut-être que chaque décision d'autorisation doit pouvoir être résolue en une seule requête d'autorisation. Mais cela peut mener à des modèles trop complexes et à des déclarations politiques alambiquées. Dans la pratique, l'utilisation d'autorisations composées peut être utile pour vous aider à produire un modèle d'autorisation plus simple. L'une des caractéristiques d'un modèle d'autorisation bien conçu est que, lorsque les actions individuelles sont suffisamment décomposées, vos opérations combinées, telles que le déplacement d'un fichier, peuvent être représentées par une agrégation intuitive de primitives.

Une autre situation dans laquelle une autorisation composée se produit est lorsque plusieurs parties sont impliquées dans le processus d'octroi d'une autorisation. Imaginons un annuaire organisationnel dans lequel les utilisateurs peuvent être membres de groupes. Une approche simple consiste à autoriser le propriétaire du groupe à ajouter n'importe qui. Cependant, que se passe-t-il si vous souhaitez que vos utilisateurs consentent d'abord à être ajoutés ? Cela introduit un accord de poignée de main dans lequel l'utilisateur et le groupe doivent consentir à l'adhésion. Pour ce faire, vous pouvez introduire une autre autorisation liée à l'utilisateur et indiquant si l'utilisateur peut être ajouté à un groupe ou à un groupe en particulier. Lorsqu'un appelant tente ensuite d'ajouter des membres à un groupe, l'application doit appliquer les deux côtés des autorisations : que l'appelant soit autorisé à ajouter des membres au groupe spécifié et que l'utilisateur individuel ajouté dispose des autorisations nécessaires pour être ajouté. QuandN-la façon dont les poignées de main existent, il est courant de l'observerNrequêtes d'autorisation composées pour appliquer chaque partie de l'accord.

Si vous êtes confronté à un défi de conception impliquant plusieurs ressources et que vous ne savez pas comment modéliser les autorisations, cela peut indiquer que vous avez un scénario d'autorisation composé. Dans ce cas, une solution peut être trouvée en décomposant l'opération en plusieurs contrôles d'autorisation individuels.

Considérations relatives à la location multiple

Vous souhaiterez peut-être développer des applications destinées à plusieurs clients (entreprises qui utilisent votre application ou locataires) et les intégrer aux autorisations vérifiées d'Amazon. Avant de développer votre modèle d'autorisation, élaborez une stratégie multi-locataires. Vous pouvez gérer les politiques de vos clients dans un magasin de politiques partagé ou attribuer à chacun un magasin de politiques par locataire.

1. Un magasin de politiques partagé

   Tous les locataires partagent un seul magasin de politiques. L'application envoie toutes les demandes d'autorisation au magasin de politiques partagé.
2. Boutique de politiques par locataire

   Chaque locataire dispose d'un magasin de polices dédié. L'application interrogera différents magasins de politiques pour obtenir une décision d'autorisation, en fonction du locataire qui fait la demande.

Aucune de ces stratégies ne crée un volume relativement élevé de demandes d'autorisation susceptibles d'avoir un impact sur votre facture. AWS Alors, comment devriez-vous concevoir votre approche ? Les conditions suivantes sont courantes susceptibles de contribuer à votre stratégie d'autorisation multi-tenant avec Verified Permissions.

Isolement des politiques relatives aux locataires

Il est important d'isoler les politiques de chaque locataire des autres pour protéger les données des locataires. Lorsque chaque locataire possède son propre magasin de politiques, il dispose de son propre ensemble de politiques isolé.

Flux d'autorisation

Vous pouvez identifier un locataire qui fait une demande d'autorisation à l'aide d'un identifiant de magasin de politiques figurant dans la demande, avec des magasins de politiques par locataire. Dans le cas d'un magasin de règles partagé, toutes les demandes utilisent le même identifiant de magasin de politiques.

Gestion des modèles et des schémas

Vos modèles de politiques et un schéma de magasin de politiques ajoutent un certain niveau de charge de conception et de maintenance à chaque magasin de politiques.

Gestion des politiques globales

Vous souhaiterez peut-être appliquer certaines politiques globales à chaque locataire. Le niveau de surcharge lié à la gestion des politiques globales varie entre les modèles de magasins de politiques partagés et par locataire.

Débarquement du locataire

Certains locataires apporteront à votre schéma et à vos politiques des éléments spécifiques à leur cas. Lorsqu'un locataire n'est plus actif au sein de votre organisation et que vous souhaitez supprimer ses données, le niveau d'effort varie en fonction de son niveau d'isolation par rapport aux autres locataires.

Quotas de ressources de service

Verified Permissions dispose de quotas de ressources et de taux de demandes susceptibles d'influencer votre décision de location multiple. Pour de plus amples informations sur les quotas, veuillez consulter Quotas de ressources.

Comparaison des magasins de politiques partagés et des magasins de politiques par locataire

Chaque considération nécessite son propre niveau d'engagement en termes de temps et de ressources dans les modèles de magasins de polices partagés et par locataire.

Considération	Niveau d'effort dans un magasin de politiques partagé	Niveau d'effort dans les magasins de polices par locataire
Isolement des politiques relatives aux locataires	Moyen. Must include tenant identifiers in policies and authorization requests.	Faible. Isolation is default behavior. Tenant-specific policies are inaccessible to other tenants.
Flux d'autorisation	Faible. All queries target one policy store.	Moyen. Must maintain mappings between each tenant and their policy store ID.

Gestion des modèles et des schémas	Faible. Must make one schema work for all tenants.	Élevée. Schemas and templates might be less complex individually, but changes require more coordination and complexity.
Gestion des politiques globales	Faible. All policies are global and can be centrally updated.	Élevée. You must add global policies to each policy store in onboarding. Replicate global policy updates between many policy stores.
Débarquement du locataire	Moyen. Must identify and delete only tenant-specific policies.	Faible. Delete the policy store.
Quotas de ressources de service	Élevée. Tenants share resource quotas that affect policy stores like schema size, policy size per resource, and identity sources per policy store.	Faible. Each tenant has dedicated resource quotas.

Comment choisir

Chaque application mutualisée est différente. Comparez soigneusement les deux approches et leurs considérations avant de prendre une décision architecturale.

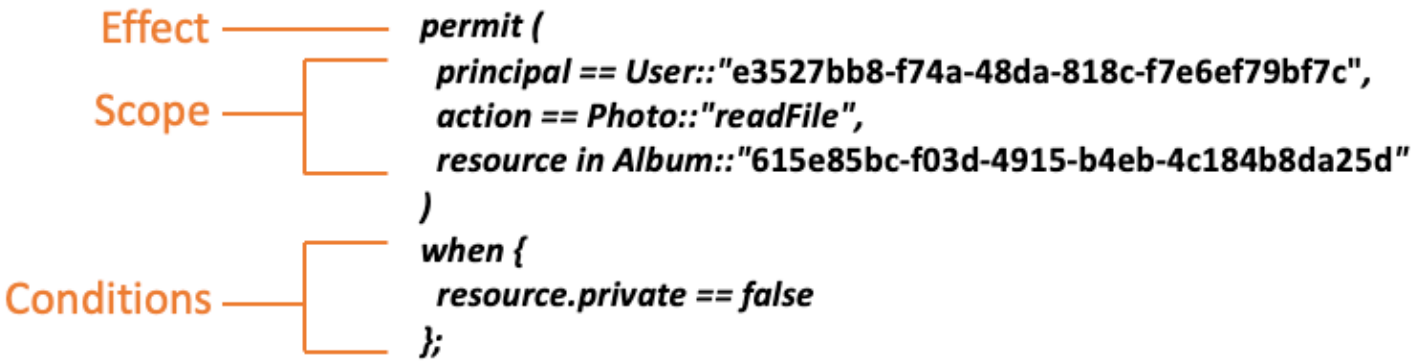
Si votre application ne nécessite pas de politiques spécifiques aux locataires et utilise une source d'identité unique, un magasin de politiques partagé pour tous les locataires est probablement la solution la plus efficace. Cela se traduit par une simplification du flux d'autorisation et de la gestion globale des politiques. Le désengagement d'un locataire à l'aide d'un magasin de politiques partagé nécessite moins d'efforts, car l'application n'a pas besoin de supprimer les politiques spécifiques au locataire.

Toutefois, si votre application nécessite de nombreuses politiques spécifiques au locataire ou utilise plusieurs sources d'identité, les magasins de politiques par locataire sont probablement les plus

efficaces. Vous pouvez contrôler l'accès aux politiques des locataires à l'aide de IAM politiques qui accordent des autorisations par locataire à chaque magasin de politiques. Le désembarquement d'un locataire implique la suppression de son magasin de politiques ; dans un shared-policy-store environnement, vous devez rechercher et supprimer les politiques spécifiques au locataire.

Dans la mesure du possible, renseignez le champ d'application de la politique

Le champ d'application de la politique est la partie d'une déclaration de politique de Cedar après le`permit`ou`forbid`mots clés et entre parenthèses ouvrantes.



Nous vous recommandons d'insérer les valeurs pour`principal`et`resource`dans la mesure du possible. Cela permet à Verified Permissions d'indexer les politiques pour une extraction plus efficace et donc d'améliorer les performances. Si vous devez accorder les mêmes autorisations à de nombreux principaux ou ressources différents, nous vous recommandons d'utiliser un modèle de politique et de l'associer à chaque paire principal/ressource.

Évitez de créer une seule grande politique contenant des listes de principes et de ressources dans un`when`clause. Cela vous exposera probablement à des limites d'évolutivité ou à des défis opérationnels. Par exemple, pour ajouter ou supprimer un seul utilisateur d'une longue liste au sein d'une politique, il est nécessaire de lire la politique dans son intégralité, de modifier la liste, d'écrire la nouvelle politique dans son intégralité et de gérer les erreurs de simultanéité si un administrateur remplace les modifications d'un autre. En revanche, en utilisant de nombreuses autorisations détaillées, l'ajout ou la suppression d'un utilisateur est aussi simple que d'ajouter ou de supprimer la seule politique qui s'applique à lui.

Chaque ressource se trouve dans un conteneur

Lorsque vous concevez un modèle d'autorisation, chaque action doit être associée à une ressource particulière. Avec une action telle que`viewFile`, la ressource à laquelle vous pouvez l'appliquer est intuitive : un fichier individuel, ou peut-être un ensemble de fichiers dans un dossier. Toutefois, une opération telle que`createFile`est moins intuitif. Lorsque vous modélisez la capacité de créer un fichier, à quelle ressource cela s'applique-t-il ? Il ne peut pas s'agir du fichier lui-même, car il n'existe pas encore.

C'est un exemple du problème généralisé de la création de ressources. La création de ressources est un problème d'amorçage. Il doit y avoir un moyen pour que quelque chose soit autorisé à créer des ressources même si aucune ressource n'existe encore. La solution consiste à reconnaître que chaque ressource doit exister dans un conteneur, et que c'est le conteneur lui-même qui sert de point d'ancrage pour les autorisations. Par exemple, si un dossier existe déjà dans le système, la possibilité de créer un fichier peut être modélisée comme une autorisation sur ce dossier, car c'est à cet endroit que les autorisations sont nécessaires pour instancier la nouvelle ressource.

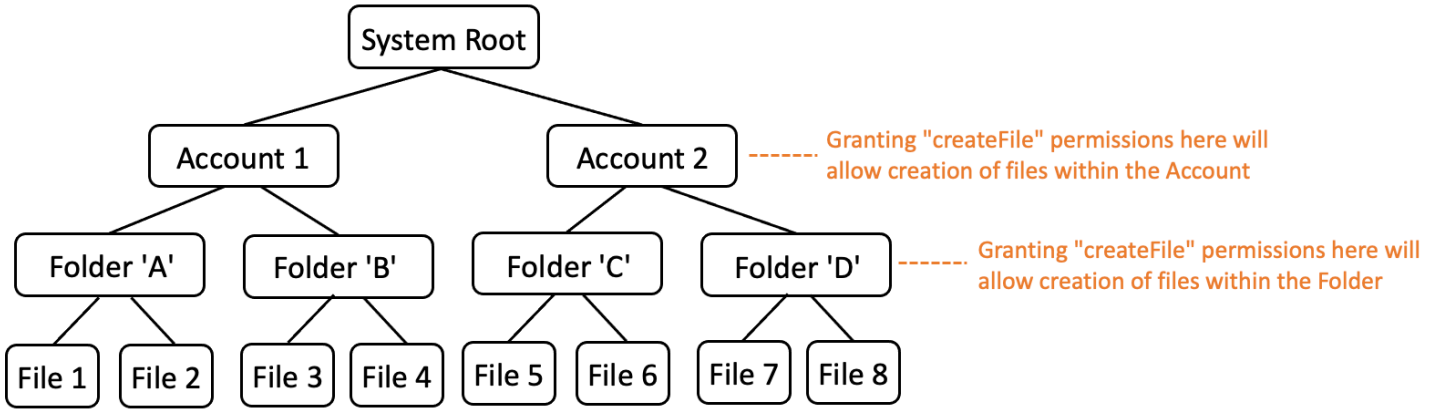
```
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    action == Action::"createFile",
    resource == Folder::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

Mais que faire si aucun dossier n'existe ? Il s'agit peut-être d'un tout nouveau compte client dans une application où aucune ressource n'existe encore. Dans cette situation, il existe toujours un contexte qui peut être compris intuitivement en se demandant : où le client peut-il créer de nouveaux fichiers ? Vous ne voulez pas qu'ils puissent créer des fichiers dans un compte client aléatoire. Il existe plutôt un contexte implicite : la limite du compte du client. Par conséquent, le compte lui-même représente le conteneur pour la création des ressources, ce qui peut être modélisé de manière explicite dans une politique similaire à celle de l'exemple suivant.

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

Et si aucun compte n'existe non plus ? Vous pouvez choisir de concevoir le flux de travail d'inscription des clients de manière à ce qu'il crée de nouveaux comptes dans le système. Si tel est le cas, vous aurez besoin d'un conteneur pour contenir la limite la plus éloignée dans laquelle le processus peut créer les comptes. Ce conteneur de niveau racine représente le système dans son ensemble et peut être nommé quelque chose comme « racine du système ». Cependant, c'est à vous, le propriétaire de l'application, de décider si cela est nécessaire et quel nom lui donner.

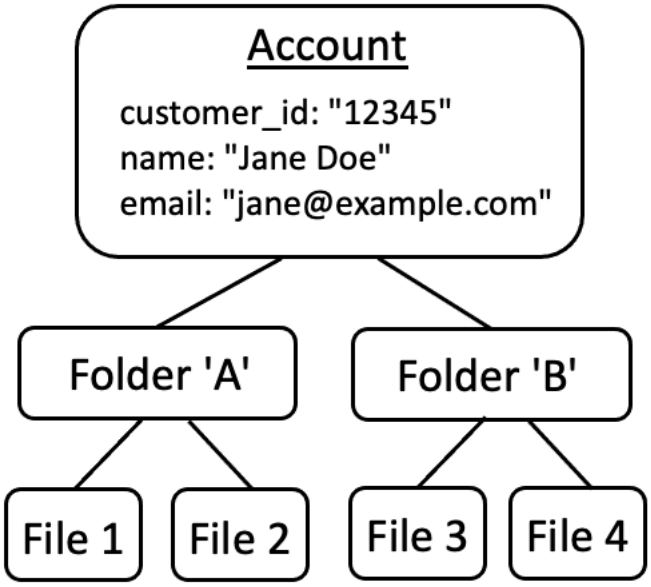
Pour cet exemple d'application, la hiérarchie des conteneurs qui en résulte apparaît donc comme suit :



Il s'agit d'un exemple de hiérarchie. D'autres sont également valides. Il ne faut pas oublier que la création de ressources se fait toujours dans le contexte d'un conteneur de ressources. Ces conteneurs peuvent être implicites, comme une limite de compte, et il peut être facile de les ignorer. Lorsque vous concevez votre modèle d'autorisation, veillez à prendre en compte ces hypothèses implicites afin qu'elles puissent être officiellement documentées et représentées dans le modèle d'autorisation.

Séparez les principaux des conteneurs de ressources

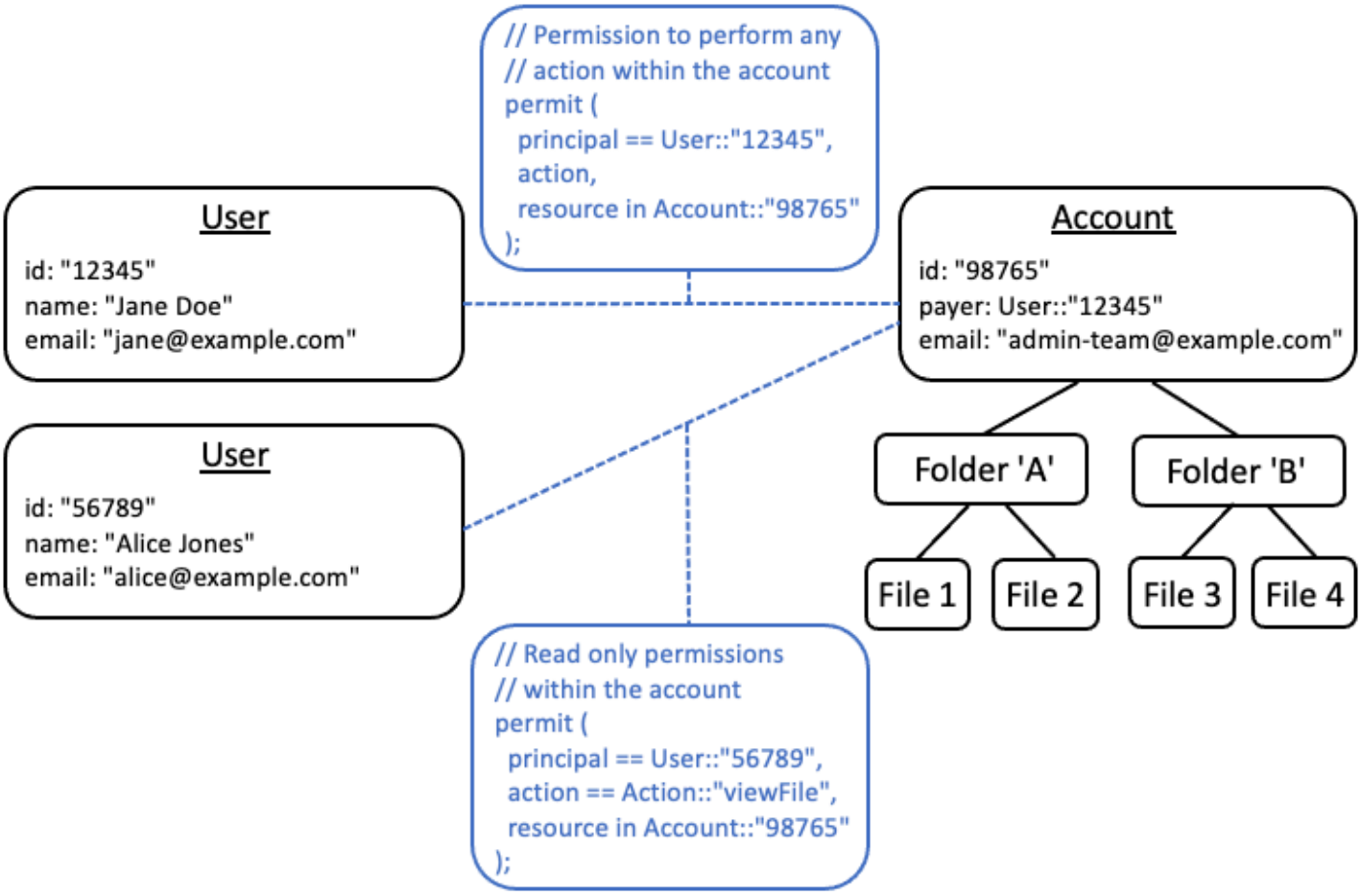
Lorsque vous concevez une hiérarchie des ressources, l'une des tendances les plus courantes, en particulier pour les applications destinées aux consommateurs, est d'utiliser l'identité utilisateur du client comme conteneur pour les ressources d'un compte client.



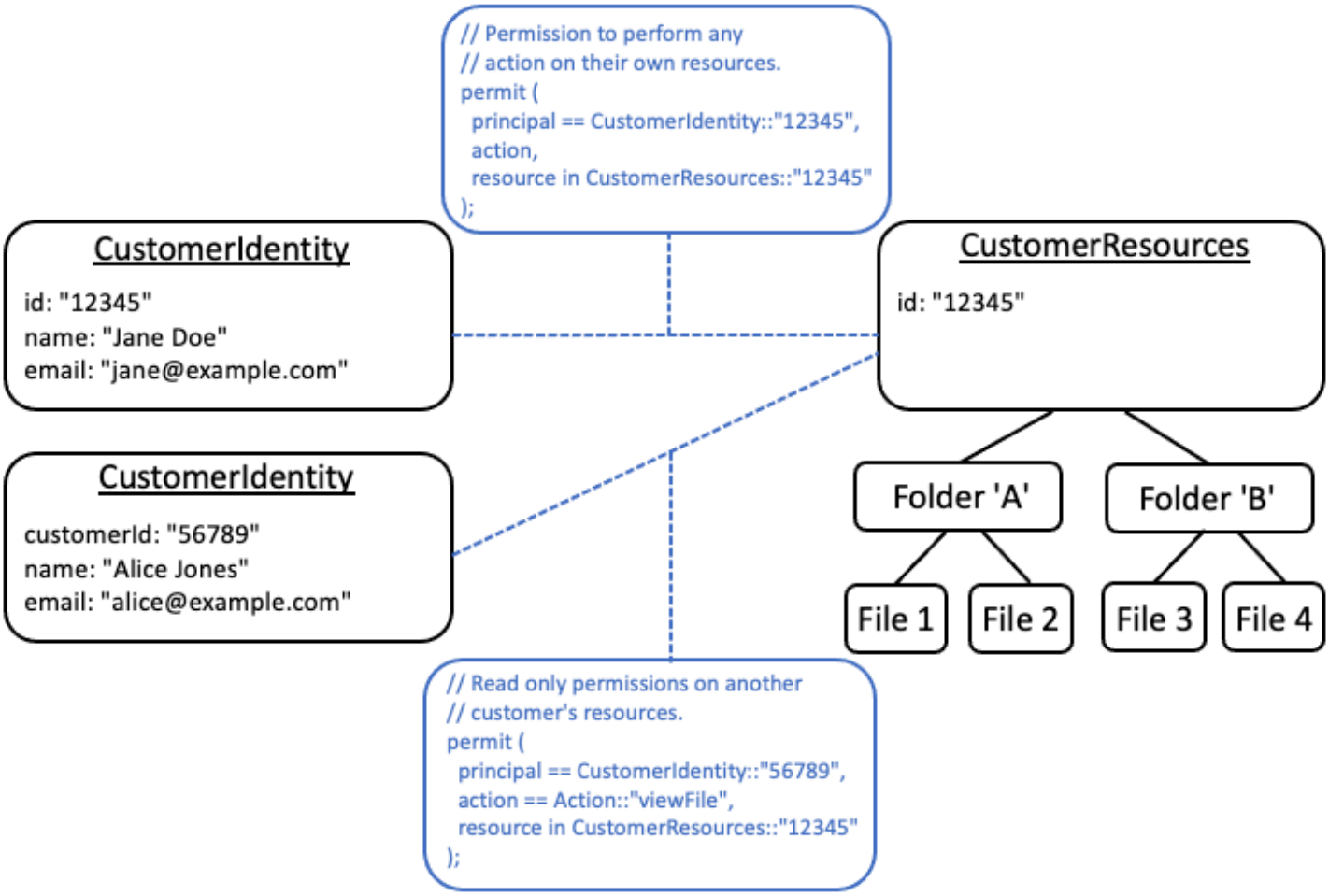
Nous vous recommandons de traiter cette stratégie comme un anti-modèle. Cela s'explique par le fait que les applications les plus riches ont naturellement tendance à déléguer l'accès à des utilisateurs supplémentaires. Par exemple, vous pouvez choisir d'introduire des comptes « familiaux », dans lesquels d'autres utilisateurs peuvent partager les ressources du compte. De même, les entreprises clientes souhaitent parfois désigner plusieurs membres du personnel comme opérateurs pour certaines parties du compte. Vous devrez peut-être également transférer la propriété d'un compte à un autre utilisateur ou fusionner les ressources de plusieurs comptes.

Lorsqu'une identité d'utilisateur est utilisée comme conteneur de ressources pour un compte, les scénarios précédents deviennent plus difficiles à réaliser. Plus alarmant encore, si d'autres personnes obtiennent l'accès au conteneur de comptes dans le cadre de cette approche, elles pourraient par inadvertance être autorisées à modifier l'identité de l'utilisateur elle-même, par exemple en modifiant l'adresse e-mail ou les informations de connexion de Jane.

Par conséquent, lorsque cela est possible, une approche plus résiliente consiste à séparer les principaux des conteneurs de ressources et à modéliser la connexion entre eux en utilisant des concepts tels que « autorisations d'administrateur » ou « propriété ».



Si une application existante n'est pas en mesure de suivre ce modèle découplé, nous vous recommandons d'envisager de l'imiter autant que possible lors de la conception d'un modèle d'autorisation. Par exemple, une application qui ne possède qu'un seul concept nomméCustomerqui encapsule l'identité de l'utilisateur, les informations de connexion et les ressources qu'il possède, pourrait associer cela à un modèle d'autorisation contenant une entité logique pourCustomer Identity(contenant le nom, l'e-mail, etc.) et une entité logique distincte pourCustomer ResourcesouCustomer Account, agissant en tant que nœud parent pour toutes les ressources qu'ils possèdent. Les deux entités peuvent partager la même choseId, mais avec un autreType.



N'intégrez pas d'autorisations dans les attributs

Il est préférable d'utiliser les attributs en tant queentréeà la décision d'autorisation. N'utilisez pas d'attributs pour représenter les autorisations elles-mêmes, par exemple en déclarant un attribut nommé « PermittedFolders » sur un utilisateur :

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

```
}
```

Ensuite, en utilisant l'attribut dans une politique :

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

Cette approche transforme ce qui serait autrement un simple modèle d'autorisation, dans lequel un principal spécifique a accès à un dossier spécifique, en un modèle de contrôle d'accès basé sur les attributs (ABAC) avec les compromis qui l'accompagnent. L'un de ces compromis est qu'il devient de plus en plus difficile de déterminer rapidement qui est autorisé à accéder à une ressource. Dans l'exemple précédent, pour déterminer qui a accès à un dossier en particulier, il est nécessaire d'effectuer une itération sur chaque utilisateur pour vérifier si ce dossier est répertorié dans ses attributs, en gardant particulièrement à l'esprit qu'il existe une politique qui accorde l'accès lorsqu'il y a accès.

Un autre risque lié à cette approche réside dans les facteurs d'échelle lorsque les autorisations sont regroupées au sein d'une seule`User`enregistré Si l'utilisateur a accès à de nombreux éléments, la taille cumulée de ses`User`le nombre de données augmentera et atteindra peut-être la limite maximale du système stockant les données.

Nous vous recommandons plutôt de représenter ce scénario à l'aide de plusieurs politiques individuelles, par exemple en utilisant des modèles de politiques pour minimiser les répétitions.

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

permit (
    principal == User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
```

```
    action == Action::"readFile",
    resource in Folder::"661817a9-d478-4096-943d-4ef1e082d19a"
);

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

Les autorisations vérifiées peuvent gérer efficacement de nombreuses politiques individuelles et précises lors de l'évaluation des autorisations. La modélisation des choses de cette manière est plus facile à gérer et à auditer au fil du temps.

Préférez les autorisations détaillées dans le modèle et les autorisations agrégées dans l'interface utilisateur

Une stratégie que les concepteurs regrettent souvent par la suite consiste à concevoir un modèle d'autorisation comportant des actions très larges, telles que`Read`et`Write`, et en se rendant compte plus tard que des mesures plus précises sont nécessaires. Le besoin d'une granularité plus fine peut être motivé par les commentaires des clients en faveur de contrôles d'accès plus précis, ou par les auditeurs de conformité et de sécurité qui encouragent les autorisations du moindre privilège.

Si les autorisations détaillées ne sont pas définies dès le départ, il peut être nécessaire de procéder à une conversion complexe pour modifier le code de l'application et les déclarations de politique en autorisations plus précises pour l'utilisateur. Par exemple, le code d'application précédemment autorisé par rapport à une action précise devra être modifié pour utiliser les actions détaillées. En outre, les politiques devront être mises à jour pour tenir compte de la migration :

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

Pour éviter cette migration coûteuse, il est préférable de définir des autorisations précises dès le départ. Cela peut toutefois entraîner un compromis si vos utilisateurs finaux sont ensuite contraints de comprendre un plus grand nombre d'autorisations précises, en particulier si la plupart des clients sont satisfaits de contrôles précis tels que`Read`et`Write`. Pour tirer le meilleur parti des deux mondes, vous pouvez regrouper des autorisations précises dans des collections prédéfinies telles que`Read`et`Write`en utilisant des mécanismes tels que des modèles de politiques ou des groupes d'action. En utilisant cette approche, les clients ne voient que les autorisations spécifiques au cours. Mais dans les coulisses, vous avez assuré la pérennité de votre application en modélisant les autorisations spécifiques sous la forme d'un ensemble d'actions précises. Lorsque les clients ou les auditeurs le demandent, les autorisations détaillées peuvent être révélées.

Envisagez d'autres raisons de demander une autorisation

Nous associons généralement les contrôles d'autorisation aux demandes des utilisateurs. La vérification permet de déterminer si l'utilisateur est autorisé à exécuter cette demande. Cependant, vous pouvez également utiliser les données d'autorisation pour influencer la conception de l'interface de l'application. Par exemple, vous souhaiterez peut-être afficher un écran d'accueil qui affiche une liste des seules ressources auxquelles l'utilisateur final peut accéder. Lorsque vous consultez les détails d'une ressource, vous souhaiterez peut-être que l'interface affiche uniquement les opérations que l'utilisateur peut effectuer sur cette ressource.

Ces situations peuvent introduire des compromis dans le modèle d'autorisation. Par exemple, une forte dépendance à l'égard des politiques de contrôle d'accès basé sur les attributs (ABAC) peut rendre plus difficile la réponse rapide à la question « qui a accès à quoi ? » En effet, pour répondre à cette question, il faut examiner chaque règle par rapport à chaque principal et à chaque ressource afin de déterminer s'il existe une correspondance. Par conséquent, un produit qui doit être optimisé pour répertorier uniquement les ressources accessibles par l'utilisateur peut choisir d'utiliser un modèle de contrôle d'accès basé sur les rôles (RBAC). En utilisant le RBAC, il peut être plus facile d'itérer toutes les politiques associées à un utilisateur afin de déterminer l'accès aux ressources.

Banc d'essai

Le banc de test des autorisations vérifiées vous permet de tester et de résoudre les problèmes liés aux politiques d'autorisations vérifiées en exécutant des demandes d'autorisation par rapport à celles-ci. Le banc de test utilise les paramètres que vous spécifiez pour déterminer si les politiques Cedar de votre magasin de politiques autoriseraient la demande. Vous pouvez basculer entre le mode visuel et le mode JSON lorsque vous testez les demandes d'autorisation. Pour plus d'informations sur la manière dont les politiques de Cedar sont structurées et évaluées, voir Construction de base des politiques dans Cedar dans le Guide de référence sur le langage des politiques de Cedar.

> **Note**
>
> Lorsque vous faites une demande d'autorisation à l'aide d'autorisations vérifiées, vous pouvez fournir la liste des principaux et des ressources dans le cadre de la demande dans la section Entités supplémentaires. Toutefois, vous ne pouvez pas inclure les détails relatifs aux actions. Ils doivent être spécifiés dans le schéma ou déduits de la demande. Vous ne pouvez pas placer d'action dans la section Entités supplémentaires.

Pour un aperçu visuel et une démonstration du banc d'essai, regardez cette vidéo.

Visual mode

> **Note**
>
> Vous devez avoir défini un schéma dans votre magasin de politiques pour utiliser le mode visuel du banc de test.

Pour tester les politiques en mode visuel

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Test bench.
3. Choisissez le mode visuel.
4. Dans la section Principal, choisissez le Principal exécutant l'action parmi les principaux types de votre schéma. Entrez un identifiant pour le principal dans la zone de texte.

5. (Facultatif) Choisissez Ajouter un parent pour ajouter des entités parents pour le principal spécifié. Pour supprimer un parent qui a été ajouté au principal, choisissez Supprimer à côté du nom du parent.
6. Spécifiez la valeur d'attribut pour chaque attribut du principal spécifié. Le banc de test utilise les valeurs d'attribut spécifiées dans la demande d'autorisation simulée.
7. Dans la section Ressource, choisissez la ressource sur laquelle le principal agit. Entrez un identifiant pour la ressource dans la zone de texte.
8. (Facultatif) Choisissez Ajouter un parent pour ajouter des entités parents pour la ressource spécifiée. Pour supprimer un parent qui a été ajouté à la ressource, choisissez Supprimer à côté du nom du parent.
9. Spécifiez la valeur d'attribut pour chaque attribut de la ressource spécifiée. Le banc de test utilise les valeurs d'attribut spécifiées dans la demande d'autorisation simulée.
10. Dans la section Action, choisissez l'action entreprise par le principal dans la liste des actions valides pour le principal et la ressource spécifiés.
11. Spécifiez la valeur d'attribut pour chaque attribut de l'action spécifiée. Le banc de test utilise les valeurs d'attribut spécifiées dans la demande d'autorisation simulée.
12. (Facultatif) Dans la section Entités supplémentaires, choisissez Ajouter une entité pour ajouter des entités à évaluer en vue de la décision d'autorisation.
13. Choisissez l'identifiant de l'entité dans la liste déroulante et saisissez l'identifiant de l'entité.
14. (Facultatif) Choisissez Ajouter un parent pour ajouter des entités parents pour l'entité spécifiée. Pour supprimer un parent qui a été ajouté à l'entité, choisissez Supprimer à côté du nom du parent.
15. Spécifiez la valeur d'attribut pour chaque attribut de l'entité spécifiée. Le banc de test utilise les valeurs d'attribut spécifiées dans la demande d'autorisation simulée.
16. Choisissez Confirmer pour ajouter l'entité au banc de test.
17. Choisissez Exécuter une demande d'autorisation pour simuler la demande d'autorisation pour les politiques Cedar dans votre magasin de polices. Le banc de test affiche la décision d'accepter ou de refuser la demande ainsi que des informations sur les politiques satisfaites ou les erreurs rencontrées lors de l'évaluation.

JSON mode

Pour tester les politiques en mode JSON

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Test bench.
3. Choisissez le mode JSON.
4. Dans la section Détails de la demande, si vous avez défini un schéma, choisissez le principal agissant parmi les principaux types de votre schéma. Entrez un identifiant pour le principal dans la zone de texte.

   Si aucun schéma n'est défini, tapez le principal dans la zone de texte Principal prenant une action.
5. Si un schéma est défini, choisissez la ressource parmi les types de ressources de votre schéma. Entrez un identifiant pour la ressource dans la zone de texte.

   Si aucun schéma n'est défini, tapez la ressource dans la zone de texte Ressource.
6. Si un schéma est défini, choisissez l'action dans la liste des actions valides pour le principal et la ressource spécifiés.

   Si aucun schéma n'est défini, tapez l'action dans la zone de texte Action.
7. Entrez le contexte de la demande à simuler dans le champ Contexte. Le contexte de la demande est une information supplémentaire qui peut être utilisée pour les décisions d'autorisation.
8. Dans le champ Entités, entrez la hiérarchie des entités et leurs attributs à évaluer pour la décision d'autorisation.
9. Choisissez Exécuter une demande d'autorisation pour simuler la demande d'autorisation pour les politiques Cedar dans votre magasin de polices. Le banc de test affiche la décision d'accepter ou de refuser la demande ainsi que des informations sur les politiques satisfaites ou les erreurs rencontrées lors de l'évaluation.

Implémentation de l'autorisation dans Amazon Verified Permissions

Une fois que vous avez créé votre magasin de politiques, vos politiques, vos modèles, votre schéma et votre modèle d'autorisation, vous êtes prêt à autoriser les demandes à l'aide des autorisations vérifiées par Amazon. Pour implémenter l'autorisation des autorisations vérifiées, vous devez combiner la configuration des politiques AWS avec l'intégration dans une application. Pour intégrer les autorisations vérifiées à votre application, ajoutez un AWS SDK et implémentez les méthodes qui invoquent l'API Verified Permissions et génèrent des décisions d'autorisation par rapport à votre magasin de politiques.

L'autorisation avec autorisations vérifiées est utile pour les autorisations UX et les autorisations API dans vos applications.

Autorisations d'expérience utilisateur

> Contrôlez l'accès des utilisateurs à l'expérience utilisateur de votre application. Vous pouvez autoriser un utilisateur à afficher uniquement les formulaires, boutons, graphiques et autres ressources exacts auxquels il a besoin pour accéder. Par exemple, lorsqu'un utilisateur se connecte, vous souhaiterez peut-être déterminer si le bouton « Transférer des fonds » est visible sur son compte. Vous pouvez également contrôler les actions qu'un utilisateur peut effectuer. Par exemple, dans la même application bancaire, vous souhaiterez peut-être déterminer si votre utilisateur est autorisé à modifier la catégorie d'une transaction.

Autorisations d'API

> Contrôlez l'accès des utilisateurs aux données. Les applications font souvent partie d'un système distribué et fournissent des informations provenant d'API externes. Dans l'exemple de l'application bancaire où les autorisations vérifiées ont autorisé l'affichage d'un bouton « Transférer des fonds », une décision d'autorisation plus complexe doit être prise lorsque votre utilisateur initie un transfert. Les autorisations vérifiées peuvent autoriser la demande d'API qui répertorie les comptes de destination qui sont des cibles de transfert éligibles, puis la demande de transfert vers l'autre compte.

Les exemples illustrant ce contenu proviennent d'un exemple de magasin de politiques. Pour suivre, créez le DigitalPetmagasin d'exemples de politiques Store dans votre environnement de test.

Pour un exemple d'application de bout en bout qui implémente les autorisations UX à l'aide d'une autorisation par lots, consultez Utiliser les autorisations vérifiées par Amazon pour une autorisation précise à grande échelle sur le blog AWS de sécurité.

Opérations d'API pour l'autorisation

L'API Verified Permissions effectue les opérations d'autorisation suivantes.

IsAuthorized

Le fonctionnement de l'`IsAuthorized`API est le point d'entrée des demandes d'autorisation avec des autorisations vérifiées. Vous devez soumettre des éléments principaux, d'action, de ressources, de contexte et d'entités. Verified Permissions valide les entités de votre demande par rapport au schéma de votre magasin de politiques. Verified Permissions évalue ensuite votre demande par rapport à toutes les politiques du magasin de politiques demandé qui s'appliquent aux entités de la demande.

IsAuthorizedWithToken

L'`IsAuthorizedWithToken`opération génère une demande d'autorisation à partir des données utilisateur contenues dans les jetons Web (JWT) Amazon Cognito JSON. Les autorisations vérifiées fonctionnent directement avec Amazon Cognito en tant que source d'identité dans votre magasin de politiques. Verified Permissions renseigne tous les attributs du principal de votre demande à partir des demandes figurant dans les identifiants des utilisateurs ou les jetons d'accès. Vous pouvez autoriser des actions et des ressources à partir des attributs utilisateur ou de l'appartenance à un groupe dans un groupe d'utilisateurs Amazon Cognito.

Vous ne pouvez pas inclure d'informations sur les types principaux de groupes ou d'utilisateurs dans une `IsAuthorizedWithToken` demande. Vous devez renseigner toutes les données principales du JWT que vous fournissez.

BatchIsAutorisé

L'`BatchIsAuthorized`opération traite plusieurs décisions d'autorisation pour un seul principal ou une seule ressource dans le cadre d'une seule demande d'API. Cette opération regroupe les demandes en une seule opération par lots qui minimise l'utilisation des quotas et renvoie les décisions d'autorisation pour chacune des 30 actions imbriquées complexes (maximum). Avec l'autorisation par lots pour une seule ressource, vous pouvez filtrer les actions qu'un utilisateur peut effectuer sur une ressource. Avec l'autorisation par lots pour un seul principal, vous pouvez filtrer les ressources sur lesquelles un utilisateur peut agir.

BatchIsAuthorizedWithJeton

L'`BatchIsAuthorizedWithToken`opération traite plusieurs décisions d'autorisation pour un seul principal dans une seule demande d'API. Le principal est fourni par la source d'identité de votre magasin de politiques sous forme d'identifiant ou de jeton d'accès. Cette opération regroupe les demandes en une seule opération par lots qui minimise l'utilisation des quotas et renvoie les décisions d'autorisation pour chacune des 30 demandes d'actions et de ressources (maximum). Dans vos politiques, vous pouvez autoriser leur accès à partir de leurs attributs ou de leur appartenance à un groupe dans un groupe d'utilisateurs Amazon Cognito.

De même`IsAuthorizedWithToken`, vous ne pouvez pas inclure d'informations sur les types principaux de groupes ou d'utilisateurs dans une `BatchIsAuthorizedWithToken` demande. Vous devez renseigner toutes les données principales du JWT que vous fournissez.

Tester votre modèle d'autorisation

Pour comprendre l'effet de la décision d'autorisation des autorisations vérifiées lorsque vous déployez votre application, vous pouvez évaluer vos politiques au fur Banc d'essai et à mesure que vous les développez à l'aide des demandes d'API REST HTTPS adressées aux autorisations vérifiées. Le banc de test est un outil permettant d' AWS Management Console évaluer les demandes d'autorisation et les réponses dans votre magasin de politiques.

L'API REST Verified Permissions est la prochaine étape de votre développement alors que vous passez de la compréhension conceptuelle à la conception d'applications. L'API Verified Permissions accepte les demandes d'autorisation avec IsAuthorizedIsAuthorizedWithToken, et BatchIsAuthorized en tant que demandes d' AWS API signées adressées aux points de terminaison de service régionaux. Pour tester votre modèle d'autorisation, vous pouvez générer des demandes auprès de n'importe quel client d'API et vérifier que vos politiques renvoient les décisions d'autorisation comme prévu.

Par exemple, vous pouvez effectuer un test `IsAuthorized` dans un exemple de magasin de politiques en suivant la procédure suivante.

Test bench

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Créez un magasin de politiques à partir du magasin de politiques Sample avec le nom DigitalPetStore.

2. Sélectionnez Test bench dans votre nouveau magasin de polices.
3. Remplissez votre demande de banc de test IsAuthorizeddans la référence de l'API Verified Permissions. Les détails suivants reproduisent les conditions de l'exemple 4 qui fait référence à l'échantillon DigitalPetStore.

    a. Définissez Alice comme directrice. Pour que le principal passe à l'action, choisissez `DigitalPetStore::User` et entrez`Alice`.
    b. Définissez le rôle d'Alice en tant que cliente. Choisissez Ajouter un parent`DigitalPetStore::Role`, puis saisissez Client.
    c. Définissez la ressource dans l'ordre « 1234 ». Pour la ressource sur laquelle le principal agit, choisissez `DigitalPetStore::Order` et entrez`1234`.
    d. La `DigitalPetStore::Order` ressource nécessite un `owner` attribut. Définissez Alice comme propriétaire de la commande. Choisissez `DigitalPetStore::User` et entrez `Alice`
    e. Alice a demandé à voir la commande. Dans la zone Action entreprise par le principal, sélectionnez`DigitalPetStore::Action::"GetOrder"`.
4. Choisissez Exécuter la demande d'autorisation. Dans un magasin de politiques non modifié, cette demande aboutit à une `ALLOW` décision. Notez la politique Satisfied qui a renvoyé la décision.
5. Choisissez Politiques dans la barre de navigation de gauche. Passez en revue la politique statique avec la description Rôle du client - Obtenir une commande.
6. Notez que Verified Permissions a autorisé la demande parce que le principal occupait un rôle de client et était le propriétaire de la ressource.

REST API

1. Ouvrez la console des autorisations vérifiées à l'adresse https://console.aws.amazon.com/verifiedpermissions/. Créez un magasin de politiques à partir du magasin de politiques Sample avec le nom DigitalPetStore.
2. Notez l'ID du magasin Policy de votre nouveau Policy Store.
3. IsAuthorizedDans la référence de l'API Verified Permissions, copiez le corps de la demande de l'exemple 4 qui fait référence à l'exemple DigitalPetStore.
4. Ouvrez votre client API et créez une demande auprès du point de terminaison du service régional pour votre magasin de politiques. Remplissez les en-têtes comme indiqué dans l'exemple.

5. Collez le corps de la demande d'exemple et remplacez la valeur de par `policyStoreId` l'ID du magasin de politiques que vous avez indiqué précédemment.
6. Soumettez la demande et examinez les résultats. Dans un magasin de politiques DigitalPetStore par défaut, cette demande renvoie une `ALLOW` décision.

Vous pouvez modifier les politiques, le schéma et les demandes dans votre environnement de test afin de modifier les résultats et de prendre des décisions plus complexes.

1. Modifiez la demande de manière à modifier la décision prise à partir des autorisations vérifiées. Par exemple, remplacez le rôle d'Alice par `Employee` ou modifiez l'`owner`attribut de l'ordre 1234 par`Bob`.
2. Modifiez les politiques de manière à affecter les décisions d'autorisation. Par exemple, modifiez la politique avec la description Rôle du client - Obtenir la commande pour supprimer la condition selon laquelle `User` il doit être le propriétaire du `Resource` et modifiez la demande afin que `Bob` celui-ci souhaite consulter la commande.
3. Modifiez le schéma pour permettre aux politiques de prendre des décisions plus complexes. Mettez à jour les entités de demande afin qu'Alice puisse satisfaire aux nouvelles exigences. Par exemple, modifiez le schéma pour autoriser `User` à être membre de `ActiveUsers` ou`InactiveUsers`. Mettez à jour la politique afin que seuls les utilisateurs actifs puissent consulter leurs propres commandes. Mettez à jour les entités de demande afin qu'Alice soit une utilisateur active ou inactive.

Intégration aux applications et aux AWS SDK

Pour implémenter les autorisations vérifiées par Amazon dans votre application, vous devez définir les politiques et le schéma que vous souhaitez que votre application applique. Une fois votre modèle d'autorisation en place et testé, l'étape suivante consiste à commencer à générer des demandes d'API dès leur mise en application. Pour ce faire, vous devez configurer la logique de l'application afin de collecter les données utilisateur et de les renseigner en fonction des demandes d'autorisation.

Comment une application autorise les demandes avec des autorisations vérifiées

1. Recueillez des informations sur l'utilisateur actuel. Généralement, les informations d'un utilisateur sont fournies dans les détails d'une session authentifiée, comme un JWT ou un cookie de session Web. Ces données utilisateur peuvent provenir d'une source d'identité Amazon Cognito liée à votre magasin de politiques ou d'un autre fournisseur OpenID Connect (OIDC).

2. Rassemblez des informations sur la ressource à laquelle un utilisateur souhaite accéder. En règle générale, votre application reçoit des informations sur la ressource lorsqu'un utilisateur effectue une sélection nécessitant le chargement d'une nouvelle ressource par votre application.
3. Déterminez l'action que votre utilisateur souhaite effectuer.
4. Générez une demande d'autorisation auprès de Verified Permissions avec le principal, l'action, la ressource et les entités correspondant à la tentative d'opération de votre utilisateur. Verified Permissions évalue la demande par rapport aux politiques de votre magasin de politiques et renvoie une décision d'autorisation.
5. Votre application lit la réponse d'autorisation ou de refus de Verified Permissions et applique la décision concernant la demande de l'utilisateur.

Les opérations de l'API Verified Permissions sont intégrées AWS aux SDK. Pour inclure des autorisations vérifiées dans une application, intégrez le AWS SDK correspondant à la langue de votre choix dans le package de l'application.

Pour en savoir plus et télécharger AWS les SDK, consultez la section Outils pour Amazon Web Services.

Vous trouverez ci-dessous des liens vers la documentation relative aux ressources d'autorisations vérifiées dans différents AWS SDK.

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

L' AWS SDK for JavaScript exemple suivant `IsAuthorized` provient de Simplify fined authorization with Amazon Verified Permissions et 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"
                }
            }
         }
       ]
    }
});
```

Plus de ressources pour les développeurs

- Atelier sur les autorisations vérifiées par Amazon
- Autorisations vérifiées par Amazon - Ressources
- Implémenter un fournisseur de politique d'autorisation personnalisé pour les applications ASP.NET Core à l'aide d'Amazon Verified Permissions
- Créez un service d'autorisation pour les applications professionnelles à l'aide d'Amazon Verified Permissions
- Simplifiez les autorisations détaillées avec Amazon Verified Permissions et Amazon Cognito

Ajouter du contexte

Le contexte est l'information pertinente pour les décisions politiques, mais qui ne fait pas partie de l'identité de votre principal, de votre action ou de votre ressource. Vous souhaiterez peut-être autoriser une action uniquement à partir d'un ensemble d'adresses IP sources, ou uniquement si votre utilisateur s'est connecté à l'aide de l'authentification MFA. Votre application a accès à ces données de session contextuelles et doit les renseigner pour répondre aux demandes d'autorisation. Les données contextuelles d'une demande d'autorisation Verified Permissions doivent être au format JSON dans un élément. `contextMap`

Les exemples illustrant ce contenu proviennent d'un exemple de magasin de politiques. Pour suivre, créez le magasin DigitalPetStored'exemples de règles dans votre environnement de test.

L'objet de contexte suivant déclare un type de données Cedar pour chaque type de données pour une application en fonction de l'exemple de magasin DigitalPetStorede politiques.

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

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

Types de données dans le contexte des autorisations

Booléen

Un binaire `true` ou une `false` valeur. Dans l'exemple, la valeur booléenne de `true` for `MfaAuthenticated` indique que le client a effectué une authentification multifactorielle avant de demander à consulter sa commande.

Définir

Un ensemble d'éléments contextuels. Les membres de l'ensemble peuvent être tous du même type, comme dans cet exemple, ou de types différents, y compris un ensemble imbriqué. Dans l'exemple, le client est associé à 3 comptes différents.

Chaîne

Séquence de lettres, de chiffres ou de symboles entourée de '' caractères. Dans l'exemple, la `UserAgent` chaîne représente le navigateur que le client a utilisé pour demander à consulter sa commande.

Long

Un entier. Dans l'exemple, cela `RequestedOrderCount` indique que cette demande fait partie d'un lot résultant du fait que le client a demandé à consulter quatre de ses commandes passées.

Enregistrer

Une collection d'attributs. Vous devez déclarer ces attributs dans le contexte de la demande. Un magasin de politiques doté d'un schéma doit inclure cette entité et ses attributs dans le schéma. Dans l'exemple, l'`NetworkInfo`enregistrement contient des informations sur l'adresse IP d'origine de l'utilisateur, la géolocalisation de cette adresse IP telle que déterminée par le client et le chiffrement en cours de transit.

EntityIdentifier

Référence à une entité et à des attributs déclarés dans l'`entities`élément de la demande. Dans l'exemple, la commande de l'utilisateur a été approuvée par l'employé`Bob`.

Pour tester cet exemple de contexte dans l'exemple d'DigitalPetStoreapplication, vous devez mettre à jour votre demande`entities`, le schéma de votre magasin de politiques et la politique statique avec la description Customer Role - Get Order.

Modifier DigitalPetStore pour accepter le contexte d'autorisation

Au départ, DigitalPetStorece n'est pas un magasin de politiques très complexe. Il n'inclut aucune politique ou attribut de contexte préconfiguré pour prendre en charge le contexte que nous avons présenté. Pour évaluer un exemple de demande d'autorisation avec ces informations contextuelles, apportez les modifications suivantes à votre magasin de politiques et à votre demande d'autorisation.

Schema

Appliquez les mises à jour suivantes au schéma de votre magasin de politiques pour prendre en charge les nouveaux attributs de contexte. Effectuez `GetOrder` la mise à jour `actions` comme suit.

```
"GetOrder": {
   "memberOf": [],
   "appliesTo": {
      "resourceTypes": [
         "Order"
```

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

Pour référencer le type de `record` données nommé `NetworkInfo` dans le contexte de votre demande, créez une construction CommonType dans votre schéma comme suit. Une `commonType` construction est un ensemble partagé d'attributs que vous pouvez appliquer à différentes entités.

> **Note**
>
> L'éditeur de schéma visuel Verified Permissions ne prend actuellement pas en charge `commonType` les constructions. Lorsque vous les ajoutez à votre schéma, vous ne pouvez plus le visualiser en mode visuel.

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

Policy

La politique suivante définit les conditions qui doivent être remplies par chacun des éléments de contexte fournis. Il s'appuie sur la politique statique existante avec la description Customer Role - Get Order. Au départ, cette politique exige uniquement que le principal qui fait une demande soit le propriétaire de la ressource.

```
permit (
    principal in DigitalPetStore::Role::"Customer",
    action in [DigitalPetStore::Action::"GetOrder"],
    resource
) when {
    principal == resource.owner &&
    context.MfaAuthorized == true &&
```

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

Nous avons maintenant exigé que la demande de récupération d'une commande réponde aux conditions contextuelles supplémentaires que nous avons ajoutées à la demande.

1. L'utilisateur doit s'être connecté à l'aide de la MFA.
2. Le navigateur Web de l'utilisateur `User-Agent` doit contenir la chaîne`My UserAgent`.
3. L'utilisateur doit avoir demandé à consulter 4 commandes ou moins.
4. L'un des codes de compte de l'utilisateur doit être`111122223333`.
5. L'adresse IP de l'utilisateur doit provenir des États-Unis, il doit être sur une session cryptée et son adresse IP doit commencer par`192.0.2.`.
6. Un employé doit avoir approuvé sa commande. Dans l'`entities`élément de la demande d'autorisation, nous déclarerons un utilisateur Bob ayant le rôle de`Employee`.

Request body

Après avoir configuré votre magasin de politiques avec le schéma et la politique appropriés, vous pouvez présenter cette demande d'autorisation à l'opération de l'API Verified Permissions IsAuthorized. Notez que le `entities` segment contient une définition de`Bob`, un utilisateur avec un rôle de`Employee`.

```
{
   "principal": {
      "entityType": "DigitalPetStore::User",
      "entityId": "Alice"
   },
   "action": {
      "actionType": "DigitalPetStore::Action",
      "actionId": "GetOrder"
   },
   "resource": {
      "entityType": "DigitalPetStore::Order",
```

```
      "entityId": "1234"
    },
    "context": {
      "contextMap": {
       "MfaAuthorized": {
        "boolean": true
       },
       "UserAgent": {
        "string": "My UserAgent 1.12"
       },
       "RequestedOrderCount":{
        "long": 4
       },
       "AccountCodes": {
        "set": [
         {"long": 111122223333},
         {"long": 444455556666},
         {"long": 123456789012}
         ]
       },
       "NetworkInfo": {
         "record": {
           "IPAddress": {"string": "192.0.2.178"},
           "Country": {"string": "United States of America"},
           "SSL": {"boolean": true}
         }
       },
       "approvedBy": {
         "entityIdentifier": {
          "entityId": "Bob",
          "entityType": "DigitalPetStore::User"
         }
       }
      }
    },
    "entities": {
      "entityList": [
        {
          "identifier": {
             "entityType": "DigitalPetStore::User",
             "entityId": "Alice"
          },
          "attributes": {
             "memberId": {
```

```
            "string": "801b87f2-1a5c-40b3-b580-eacad506d4e6"
            }
        },
        "parents": [
           {
           "entityType": "DigitalPetStore::Role",
           "entityId": "Customer"
           }
        ]
      },
      {
        "identifier": {
           "entityType": "DigitalPetStore::User",
           "entityId": "Bob"
        },
        "attributes": {
           "memberId": {
             "string": "49d9b81e-735d-429c-989d-93bec0bcfd8b"
             }
        },
        "parents": [
           {
           "entityType": "DigitalPetStore::Role",
           "entityId": "Employee"
           }
        ]
      },
      {
        "identifier": {
           "entityType": "DigitalPetStore::Order",
           "entityId": "1234"
        },
        "attributes": {
           "owner": {
             "entityIdentifier": {
               "entityType": "DigitalPetStore::User",
               "entityId": "Alice"
             }
          }
        },
        "parents": []
      }
    ]
  },
```

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

Sécurité dans Amazon Verified Permissions

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est responsable de la protection de l'infrastructure qui exécute des services AWS dans le cloud AWS Cloud. AWS vous fournit également les services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des AWS programmes de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent aux autorisations vérifiées par Amazon, consultezAWSServices couverts par programme de conformité.
- Sécurité dans le cloud : votre responsabilité est déterminée par le service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'autorisations vérifiées. Les rubriques suivantes expliquent comment configurer les autorisations vérifiées pour atteindre vos objectifs de sécurité et de conformité. Vous apprenez également à utiliser d'autresAWSdes services qui vous aident à surveiller et à sécuriser vos ressources d'autorisations vérifiées.

Rubriques

Protection des données dans Amazon Verified Permissions

LeAWS modèle de responsabilité partagées'applique à la protection des données dans Amazon Verified Permissions. Comme décrit dans ce modèle, AWS est responsable de la protection de

l'infrastructure globale sur laquelle l'ensemble du AWS Cloud s'exécute. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Ce contenu comprend les tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour en savoir plus sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWSet RGPD (Règlement général sur la protection des données) sur le AWSBlog de sécurité.

- À des fins de protection des données, nous vous recommandons de protégerCompte AWSinformations d'identification et configuration d'utilisateurs individuels avecAWS IAM Identity CenterouAWS Identity and Access Management(IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches.
- Nous vous recommandons de sécuriser vos données de la manière suivante :
  - Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
  - Utilisez les certificats SSL/TLS pour communiquer avec les ressources AWS. Nous avons besoin du protocole TLS 1.2.
  - Configurez une API (Interface de programmation) et le journal de l'activité des utilisateurs avec AWS CloudTrail.
  - Utilisez des solutions de chiffrement AWS, ainsi que tous les contrôles de sécurité par défaut au sein des Services AWS.
  - Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
  - Si vous avez besoin de modules cryptographiques validés FIPS (Federal Information Processing Standard) 140-2 lorsque vous accédez à AWS via une CLI (Interface de ligne de commande) ou une API (Interface de programmation), utilisez un point de terminaison FIPS (Federal Information Processing Standard). Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).
- Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec des autorisations vérifiées ou d'autresServices AWSà l'aide de la console, de l'API,AWS CLI, ouAWSSDK. Toutes les données que vous saisissez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne

pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

- Les noms de vos actions ne doivent contenir aucune information sensible.
- Nous vous recommandons également vivement de toujours utiliser des identifiants uniques, non modifiables et non réutilisables pour vos entités (ressources et principaux). Dans un environnement de test, vous pouvez choisir d'utiliser des identifiants d'entités simples, tels que`jane`ou`bob`pour le nom d'une entité de type`User`. Toutefois, dans un système de production, il est essentiel pour des raisons de sécurité d'utiliser des valeurs uniques qui ne peuvent pas être réutilisées. Nous vous recommandons d'utiliser des valeurs telles que des identifiants uniques universels (UUID). Par exemple, considérez l'utilisateur`jane`qui quitte l'entreprise. Plus tard, vous laisserez quelqu'un d'autre utiliser le nom`jane`. Ce nouvel utilisateur accède automatiquement à tout ce qui est accordé par les politiques qui font toujours référence`User::"jane"`. Autorisations vérifiées et Cedar ne peuvent pas faire la distinction entre le nouvel utilisateur et l'utilisateur précédent.

  Ces directives s'appliquent à la fois aux identificateurs principaux et aux identificateurs de ressources. Utilisez toujours des identifiants dont l'unicité est garantie et qui ne seront jamais réutilisés afin de ne pas accorder l'accès par inadvertance en raison de la présence d'un ancien identifiant dans une politique.

- Assurez-vous que les chaînes que vous fournissez pour définir`Long`et`Decimal`les valeurs se situent dans la plage valide de chaque type. Assurez-vous également que votre utilisation d'opérateurs arithmétiques ne se traduit pas par une valeur en dehors de la plage valide. Si la plage est dépassée, l'opération entraîne une exception de débordement. Une politique qui entraîne une erreur est ignorée, ce qui signifie qu'une politique d'autorisation peut échouer de manière inattendue à autoriser l'accès, ou qu'une politique d'interdiction peut échouer de manière inattendue à bloquer l'accès.

Chiffrement des données

Amazon Verified Permissions chiffre automatiquement toutes les données des clients, telles que les politiques, à l'aide d'unClé gérée par AWS, de sorte que l'utilisation d'une clé gérée par le client n'est ni nécessaire ni prise en charge.

Gestion des identités et des accès pour Amazon Verified Permissions

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. IAM les administrateurs contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources d'autorisations vérifiées. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans Permissions vérifiées.

Utilisateur du service : si vous utilisez le service d'autorisations vérifiées pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités d'autorisations vérifiées pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne parvenez pas à accéder à une fonctionnalité dans Autorisations vérifiées, consultezRésolution des problèmes liés à l'identité et à l'accès aux autorisations vérifiées par Amazon.

Administrateur du service — Si vous êtes responsable des ressources des autorisations vérifiées au sein de votre entreprise, vous avez probablement un accès complet aux autorisations vérifiées. C'est à vous de déterminer les fonctionnalités et les ressources des autorisations vérifiées auxquelles les utilisateurs du service doivent accéder. Vous devez ensuite envoyer des demandes à votre

IAM administrateur pour modifier les autorisations des utilisateurs de votre service. Consultez les informations de cette page pour comprendre les concepts de base de IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM les autorisations vérifiées, consultezComment fonctionne Amazon Verified Permissions avec IAM.

IAM administrateur : si vous êtes IAM administrateur, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès aux autorisations vérifiées. Pour consulter des exemples de politiques basées sur l'identité relatives aux autorisations vérifiées que vous pouvez utiliser IAM, consultez. Exemples de politiques basées sur l'identité pour Amazon Verified Permissions

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un IAM rôle. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez en tant qu'identité fédérée, votre administrateur a préalablement configuré la fédération d'identité à l'aide de IAM rôles. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section Signature des demandes d' AWS API dans le guide de IAM l'utilisateur.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir

plus, voir Authentification multifactorielle dans le guide de AWS IAM Identity Center l'utilisateur et Utilisation de l'authentification multifactorielle (MFA) AWS dans le guide de l'utilisateur.IAM

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui nécessitent que vous vous connectiez en tant qu'utilisateur root, consultez la section Tâches nécessitant des informations d'identification utilisateur root dans le Guide de IAM l'utilisateur.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés

d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, voir Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification à long terme dans le Guide de IAM l'utilisateur.

Un IAM groupe est une identité qui spécifie un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé Admins IAM et accorder à ce groupe les autorisations leur permettant d'administrer des ressources IAM .

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, voir Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'IAM utilisateur.

IAM rôles

Un IAM rôle est une identité au sein de Compte AWS vous dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un IAM rôle dans le en AWS Management Console changeant de rôle. Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section Utilisation IAM des rôles dans le Guide de IAM l'utilisateur.

IAM les rôles dotés d'informations d'identification temporaires sont utiles dans les situations suivantes :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour plus d'informations sur les rôles pour la fédération, voir Création d'un rôle pour un fournisseur d'identité tiers dans le guide de IAM l'utilisateur. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

- Autorisations utilisateur IAM temporaires : un utilisateur ou un rôle IAM peut assumer un IAM rôle afin d'obtenir temporairement différentes autorisations pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (un principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, consultez la section En quoi les IAM rôles diffèrent des politiques basées sur les ressources dans le Guide de l'utilisateur.IAM
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un IAM rôle pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes AWS CLI d' AWS API. Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez la section Utilisation d'un IAM rôle pour accorder des autorisations aux applications exécutées sur Amazon EC2 des instances dans le Guide de IAM l'utilisateur.

Pour savoir s'il faut utiliser IAM des rôles ou des utilisateurs IAM, voir Quand créer un IAM rôle (au lieu d'un utilisateur) dans le guide de l'IAM utilisateur.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez la section Présentation des politiques JSON dans le guide de IAM l'utilisateur.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

IAM les politiques définissent les autorisations pour une action, quelle que soit la méthode que vous utilisez pour effectuer l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, consultez la section Création de IAM politiques dans le Guide de l'IAM utilisateur.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour savoir comment choisir entre une politique gérée ou une politique intégrée, voir Choisir entre des politiques gérées et des politiques intégrées dans le Guide de l'IAM utilisateur.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser de politiques AWS gérées depuis une IAM stratégie basée sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principals (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Vue d'ensemble des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limites d'autorisations — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez la section Limites d'autorisations pour les IAM entités dans le Guide de IAM l'utilisateur.
- Politiques de contrôle des services (SCP) — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans. AWS Organizations AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations .

- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour en savoir plus, consultez Politiques de session dans le Guide de l'utilisateur IAM .

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de IAM l'utilisateur.

Comment fonctionne Amazon Verified Permissions avec IAM

Avant de gérer l' IAM accès aux autorisations vérifiées, découvrez quelles IAM fonctionnalités peuvent être utilisées avec les autorisations vérifiées.

IAM fonctionnalités que vous pouvez utiliser avec Amazon Verified Permissions

IAM fonctionnalité	Support des autorisations vérifiées
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition d'une politique	Non
ACL	Non
ABAC (étiquettes dans les politiques)	Non
Informations d'identification temporaires	Oui

IAM fonctionnalité	Support des autorisations vérifiées
Autorisations de principal	Oui
Fonctions du service	Non
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble du fonctionnement des autorisations vérifiées et AWS des autres services avec la plupart des IAM fonctionnalités, consultez la section AWS Services compatibles IAM dans le Guide de IAM l'utilisateur.

Politiques basées sur l'identité pour les autorisations vérifiées

Prend en charge les politiques basées sur l'identité	Oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, consultez la section Création de IAM politiques dans le Guide de l'IAM utilisateur.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier les actions et les ressources autorisées ou refusées ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour en savoir plus sur tous les éléments que vous pouvez utiliser dans une politique JSON, consultez la référence des éléments de stratégie IAM JSON dans le guide de IAM l'utilisateur.

Exemples de politiques basées sur l'identité pour les autorisations vérifiées

Pour consulter des exemples de politiques basées sur l'identité relatives aux autorisations vérifiées, consultez. Exemples de politiques basées sur l'identité pour Amazon Verified Permissions

Politiques basées sur les ressources dans le cadre des autorisations vérifiées

Prend en charge les politiques basées sur les ressources	Non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour activer l'accès entre comptes, vous pouvez spécifier un compte entier ou IAM des entités d'un autre compte comme principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un IAM administrateur du compte de confiance doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez la section Accès aux ressources entre comptes IAM dans le Guide de IAM l'utilisateur.

Actions politiques pour les autorisations vérifiées

Prend en charge les actions de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom

que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions d'autorisations vérifiées, consultez la section Actions définies par Amazon Verified Permissions dans la référence d'autorisation de service.

Les actions de politique dans les autorisations vérifiées utilisent le préfixe suivant avant l'action :

```
verifiedpermissions
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

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

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Get`, incluez l'action suivante :

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

Pour consulter des exemples de politiques basées sur l'identité relatives aux autorisations vérifiées, consultez. Exemples de politiques basées sur l'identité pour Amazon Verified Permissions

Ressources relatives aux politiques relatives aux autorisations vérifiées

Prend en charge les ressources de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de ressources avec autorisations vérifiées et leurs ARN, consultez la section Types de ressources définis par Amazon Verified Permissions dans la référence d'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon Verified Permissions.

Clés de conditions de politique pour les autorisations vérifiées

Prend en charge les clés de condition de politique spécifiques au service	Non

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez IAM la section Éléments de politique : variables et balises dans le Guide de IAM l'utilisateur.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés contextuelles de condition AWS globales dans le guide de IAM l'utilisateur.

ACL dans les autorisations vérifiées

Prend en charge les listes ACL	Non

Les listes de contrôle d'accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec autorisations vérifiées

Prise en charge d'ABAC (identifications dans les politiques)	Non

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises à IAM des entités (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, voir Qu'est-ce que l'ABAC ? dans le guide de IAM l'utilisateur. Pour consulter un didacticiel présentant les étapes de configuration d'ABAC, voir Utiliser le contrôle d'accès basé sur les attributs (ABAC) dans le guide de l'utilisateur.IAM

Utilisation d'informations d'identification temporaires avec autorisations vérifiées

Prend en charge les informations d'identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation IAM dans le Guide de IAM l'utilisateur.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez la section Passage à un rôle (console) dans le guide de IAM l'utilisateur.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Autorisations principales interservices pour les autorisations vérifiées

Prend en charge les autorisations de principal	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Rôles de service pour les autorisations vérifiées

Prend en charge les fonctions de service	Non

Un rôle de service est un IAM rôle qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieur IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM .

Rôles liés à un service pour les autorisations vérifiées

Prend en charge les rôles liés à un service	Non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un IAM administrateur peut consulter, mais pas modifier les autorisations pour les rôles liés à un service.

Pour plus de détails sur la création ou la gestion des rôles liés à un service, consultez la section AWS Services compatibles avec. IAM Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour Amazon Verified Permissions

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources d'autorisations vérifiées. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Un IAM administrateur doit créer des IAM politiques qui accordent aux utilisateurs et aux rôles l'autorisation d'effectuer des actions sur les ressources dont ils ont besoin. L'administrateur doit ensuite attacher ces stratégies aux utilisateurs qui en ont besoin.

Pour savoir comment créer une politique IAM basée sur l'identité à l'aide de ces exemples de documents de stratégie JSON, consultez la section Création de IAM politiques dans le guide de l'IAM utilisateur.

Pour plus de détails sur les actions et les types de ressources définis par Verified Permissions, y compris le format des ARN pour chacun des types de ressources, consultez la section Actions, ressources et clés de condition pour les autorisations vérifiées Amazon dans le Service Authorization Reference.

Rubriques

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources d'autorisations vérifiées dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à

vos cas d'utilisation. Pour plus d'informations, consultez les politiques AWS gérées ou les politiques AWS gérées pour les fonctions professionnelles dans le Guide de IAM l'utilisateur.

- Appliquer les autorisations du moindre privilège : lorsque vous définissez des autorisations à IAM l'aide de politiques, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation IAM pour appliquer des autorisations, consultez la section Politiques et autorisations IAM dans le guide de IAM l'utilisateur.
- Utilisez des conditions dans IAM les politiques pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques pour limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez la section Éléments de politique JSON IAM : Condition dans le Guide de l'utilisateur IAM .
- Utilisez IAM Access Analyzer pour valider vos IAM politiques afin de garantir des autorisations sécurisées et fonctionnelles. IAM Access Analyzer valide les politiques nouvelles et existantes afin qu'elles soient conformes au langage des politiques (JSON) et aux IAM meilleures pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez la section Validation des politiques d'IAM Access Analyzer dans le guide de l'IAM utilisateur.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez la section Configuration de l'accès aux API protégées par MFA dans le guide de l'IAM utilisateur.

Pour plus d'informations sur les meilleures pratiques en matière de sécurité IAM, consultez la section Bonnes pratiques en matière de sécurité IAM dans le Guide de IAM l'utilisateur.

Utilisation de la console d'autorisations vérifiées

Pour accéder à la console Amazon Verified Permissions, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails des ressources d'autorisations vérifiées de votre Compte AWS. Si vous créez une stratégie

basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la console des autorisations vérifiées, associez également les autorisations vérifiées *ConsoleAccess* ou la politique *ReadOnly* AWS gérée aux entités. Pour plus d'informations, consultez la section Ajouter des autorisations à un utilisateur dans le Guide de IAM l'utilisateur.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or 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": "*"
        }
    ]
}
```

Résolution des problèmes liés à l'identité et à l'accès aux autorisations vérifiées par Amazon

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec des autorisations vérifiées et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans Permissions vérifiées
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources d'autorisations vérifiées

Je ne suis pas autorisé à effectuer une action dans Permissions vérifiées

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `verifiedpermissions:`*`GetWidget`* fictives.

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

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `verifiedpermissions:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle aux autorisations vérifiées.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Autorisations vérifiées. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources d'autorisations vérifiées

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si les autorisations vérifiées prennent en charge ces fonctionnalités, consultezComment fonctionne Amazon Verified Permissions avec IAM.

- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'IAM utilisateur.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le Guide de IAM l'utilisateur.
- Pour savoir comment fournir un accès via la fédération d'identité, consultez la section Fournir un accès aux utilisateurs authentifiés de manière externe (fédération d'identité) dans le guide de l'IAM utilisateur.
- Pour connaître la différence entre l'utilisation de rôles et l'utilisation de politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux ressources entre comptes IAM dans le guide de l'IAM utilisateur.

Validation de conformité pour les autorisations vérifiées par Amazon

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides de démarrage rapide sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur AWS la sécurité et la conformité.
- Architecture axée sur la sécurité et la conformité à la loi HIPAA Amazon Web Services : ce livre blanc décrit comment les entreprises peuvent créer des applications AWS conformes à la loi HIPAA.

> **Note**
>
> Tous ne Services AWS sont pas éligibles à la loi HIPAA. Pour plus d'informations, consultez le HIPAA Eligible Services Reference.

- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).
- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.
- Amazon GuardDuty — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience dans Amazon Verified Permissions

L'infrastructure mondiale d'AWS est construite autour de zones de disponibilité et de Régions AWS. Les Régions AWSfournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones

de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Lorsque vous créez un magasin de politiques d'autorisations vérifiées, il est créé au sein d'un individuRégion AWS, et est automatiquement répliqué dans les centres de données qui constituent les zones de disponibilité de cette région. À l'heure actuelle, les autorisations vérifiées ne prennent pas en charge la réplication entre régions.

Pour plus d'informations sur les Régions AWS et les zones de disponibilité, consultez Infrastructure mondiale d'AWS.

Surveillance des autorisations vérifiées par Amazon

La surveillance est un aspect important du maintien de la fiabilité, de la disponibilité et des performances d'Amazon Verified Permissions et de vos autres AWS solutions. AWSfournit les outils de surveillance suivants pour surveiller les autorisations vérifiées, signaler les incidents et prendre des mesures automatiques le cas échéant :

- AWS CloudTrail capture les appels d'API et les événements associés créés par ou au nom de votre compte AWS et envoie les fichiers journaux à un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes qui ont appelé AWS, l'adresse IP source à partir de laquelle les appels ont été émis, ainsi que le moment où les appels ont eu lieu. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS CloudTrail.

Journalisation des appels d'API Amazon Verified Permissions à l'aide de AWS CloudTrail

Amazon Verified Permissions est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou un AWS service dans Verified Permissions. CloudTrail capture tous les appels d'API pour les autorisations vérifiées sous forme d'événements. Les appels capturés incluent des appels provenant de la console Verified Permissions et des appels de code vers les opérations de l'API Verified Permissions. Si vous créez un suivi, vous pouvez activer la diffusion continue des CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour les autorisations vérifiées. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite à Verified Permissions, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

Informations sur les autorisations vérifiées dans CloudTrail

CloudTrail est activé sur votre compte Compte AWS lorsque vous créez le compte. Lorsqu'une activité se produit dans les autorisations vérifiées, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre Compte AWS. Pour

plus d'informations, consultez la section Affichage des événements à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre siteCompte AWS, y compris des événements pour les autorisations vérifiées, créez une trace. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal de suivi consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour en savoir plus, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les actions relatives aux autorisations vérifiées sont enregistrées CloudTrail et documentées dans le guide de référence de l'API Amazon Verified Permissions. Par exemple, les appels aux `CreateIdentitySourceDeletePolicy`, et `ListPolicyStores` les actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification racine ou utilisateur AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été effectuée par un autre service AWS.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail .

Les événements liés aux données IsAuthorizedWithTokensont similaires IsAuthorizedet ne sont pas enregistrés par défaut lorsque vous créez un magasin de données de suivi ou d'événement.

Pour enregistrer CloudTrail les événements liés aux données, vous devez ajouter explicitement les ressources prises en charge ou les types de ressources pour lesquels vous souhaitez collecter des activités. Pour plus d'informations, veuillez consulter la rubrique Événements de données dans le Guide de l'utilisateur AWS CloudTrail.

Comprendre les entrées du fichier journal des autorisations vérifiées

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

Rubriques

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

> **Note**
>
> Certains champs qui ont été supprimés des exemples relatifs à la confidentialité des données.

IsAuthorized


```
{
    "eventVersion": "1.08",
    "userIdentity": {
  "type": "AssumedRole",
  "principalId": "EXAMPLE_PRINCIPAL_ID",
  "arn": "arn:aws:iam::123456789012:role/ExampleRole",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    "eventTime": "2023-11-20T22:55:03Z",
    "eventSource": "verifiedpermissions.amazonaws.com",
    "eventName": "IsAuthorized",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
    "requestParameters": {
        "principal": {
            "entityType": "PhotoFlash::User",
            "entityId": "alice"
        },
        "action": {
            "actionType": "PhotoFlash::Action",
            "actionId": "ViewPhoto"
        },
        "resource": {
            "entityType": "PhotoFlash::Photo",
            "entityId": "VacationPhoto94.jpg"
        },
        "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    "responseElements": null,
    "additionalEventData": {
        "decision": "ALLOW"
```

```
    },
    "requestID": "346c4b6a-d12f-46b6-bc06-6c857bd3b28e",
    "eventID": "8a4fed32-9605-45dd-a09a-5ebbf0715bbc",
    "readOnly": true,
    "resources": [
        {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
}
```

BatchIsAuthorized

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

```
                "action": {
                    "actionType": "PhotoFlash::Action",
                    "actionId": "ViewPhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            },
            {
                "principal": {
                    "entityType": "PhotoFlash::User",
                    "entityId": "annalisa"
                },
                "action": {
                    "actionType": "PhotoFlash::Action",
                    "actionId": "DeletePhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            }
        ],
        "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    "responseElements": null,
    "additionalEventData": {
        "results": [
            {
                "request": {
                    "principal": {
                        "entityType": "PhotoFlash::User",
                        "entityId": "alice"
                    },
                    "action": {
                        "actionType": "PhotoFlash::Action",
                        "actionId": "ViewPhoto"
                    },
                    "resource": {
                        "entityType": "PhotoFlash::Photo",
                        "entityId": "VacationPhoto94.jpg"
                    }
                },
```

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

CreatePolicyStore

```
{
  "eventVersion": "1.08",
  "userIdentity": {
```

```
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "validationSettings": {
      "mode": "OFF"
    }
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-22T07:43:33.962794Z",
    "lastUpdatedDate": "2023-05-22T07:43:33.962794Z"
  },
  "requestID": "1dd9360e-e2dc-4554-ab65-b46d2cf45c29",
  "eventID": "b6edaeee-3584-4b4e-a48e-311de46d7532",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```


ListPolicyStores


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

```
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListPolicyStores",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "maxResults": 10
  },
  "responseElements": null,
  "requestID": "5ef238db-9f87-4f37-ab7b-6cf0ba5df891",
  "eventID": "b0430fb0-12c3-4cca-8d05-84c37f99c51f",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

DeletePolicyStore

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

```
  "eventID": "ac482022-b2f6-4069-879a-dd509123d8d7",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

PutSchema

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

```
  "eventID": "7cd0c677-733f-4602-bc03-248bae581fe5",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

GetSchema

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

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

CreatePolicyTemplate

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

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

DeletePolicyTemplate

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

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

CreatePolicy

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

```
  "requestID": "93ffa151-3841-4960-9af6-30a7f817ef93",
  "eventID": "30ab405f-3dff-43ff-8af9-f513829e8bde",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

GetPolicy

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

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

CreateIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-19T01:27:44Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreateIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "configuration": {
      "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:000011112222:us-east-1:userpool/us-
east-1_aaaaaaaaaa"
      }
    },
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "principalEntityType": "User"
  },
  "responseElements": {
    "createdDate": "2023-07-14T15:05:01.599534Z",
```

```
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-07-14T15:05:01.599534Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
  "requestID": "afcc1e67-d5a4-4a9b-a74c-cdc2f719391c",
  "eventID": "f13a41dc-4496-4517-aeb8-a389eb379860",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

GetIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:31Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
```

```
  "requestID": "7a6ecf79-c489-4516-bb57-9ded970279c9",
  "eventID": "fa158e6c-f705-4a15-a731-2cdb4bd9a427",
  "readOnly": true,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

ListIdentitySources

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T20:05:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListIdentitySources",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "95d2a7bc-7e9a-4efe-918e-97e558aacaf7",
  "eventID": "d3dc53f6-1432-40c8-9d1d-b9eeb75c6193",
  "readOnly": true,
  "resources": [
    {
```

```
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

DeleteIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeleteIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "d554d964-0957-4834-a421-c417bd293086",
  "eventID": "fe4d867c-88ee-4e5d-8d30-2fbc208c9260",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
```

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

Création de ressources Amazon Verified Permissions avec AWS CloudFormation

Amazon Verified Permissions est intégré à AWS CloudFormation un service qui vous aide à modéliser et à configurer vos AWS ressources afin que vous puissiez passer moins de temps à créer et à gérer vos ressources et votre infrastructure. Vous créez un modèle qui décrit toutes les AWS ressources que vous souhaitez (telles que les magasins de politiques), et AWS CloudFormation qui fournit et configure ces ressources pour vous.

Lorsque vous l'utilisez AWS CloudFormation, vous pouvez réutiliser votre modèle pour configurer vos ressources d'autorisations vérifiées de manière cohérente et répétée. Décrivez vos ressources une seule fois, puis distribuez les mêmes ressources encore et encore dans plusieurs Comptes AWS régions.

> **Important**
>
> Amazon Cognito Identity n'est pas disponible de la même manière qu' Régions AWS Amazon Verified Permissions. Si vous recevez un message d'erreur AWS CloudFormation concernant Amazon Cognito Identity, par exemple`Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient`, nous vous recommandons de créer le groupe d'utilisateurs et le client Amazon Cognito dans la zone géographique la plus proche où Région AWS Amazon Cognito Identity est disponible. Utilisez ce groupe d'utilisateurs nouvellement créé lors de la création de la source d'identité Verified Permissions.

Autorisations et AWS CloudFormation modèles vérifiés

Pour fournir et configurer des ressources pour les autorisations vérifiées et les services associés, vous devez comprendre les AWS CloudFormation modèles. Les modèles sont des fichiers texte formatés en JSON ou YAML. Ces modèles décrivent les ressources que vous souhaitez mettre à disposition dans vos AWS CloudFormation piles. Si vous n'êtes pas familiarisé avec JSON ou YAML, vous pouvez utiliser AWS CloudFormation Designer pour vous aider à démarrer avec les AWS CloudFormation modèles. Pour plus d'informations, voir Qu'est-ce que AWS CloudFormation Designer ? dans le guide de AWS CloudFormation l'utilisateur.

Verified Permissions prend en charge la création de sources d'identité, de politiques, de magasins de politiques et de modèles de politiques dans AWS CloudFormation. Pour plus d'informations, notamment des exemples de modèles JSON et YAML pour les ressources d'autorisations vérifiées, consultez la référence au type de ressource Amazon Verified Permissions dans le guide de l'AWS CloudFormation utilisateur.

AWS Constructions CDK

AWS Cloud Development Kit (AWS CDK) Il s'agit d'un framework de développement de logiciels open source permettant de définir l'infrastructure cloud dans le code et de la provisionner via ce dernier. AWS CloudFormation Des constructions, ou des composants cloud réutilisables, peuvent être utilisés pour créer des AWS CloudFormation modèles. Ces modèles peuvent ensuite être utilisés pour déployer votre infrastructure cloud.

Pour en savoir plus et télécharger AWS des CDK, consultez AWS Cloud Development Kit.

Vous trouverez ci-dessous des liens vers la documentation relative aux AWS CDK ressources d'autorisations vérifiées, telles que les constructions.

- Autorisations vérifiées par Amazon L2 CDK Construct

En savoir plus sur AWS CloudFormation

Pour en savoir plus AWS CloudFormation, consultez les ressources suivantes :

- AWS CloudFormation
- AWS CloudFormation Guide de l'utilisateur
- AWS CloudFormation API Reference
- AWS CloudFormation Guide de l'utilisateur de l'interface de ligne de commande

Accéder à Amazon VPC à l'aide d'un point de terminaison d'interface (AWS PrivateLink)

Vous pouvez utiliser AWS PrivateLink pour créer une connexion privée entre votre VPC et Amazon VPC et Amazon VPC. Vous pouvez accéder à PerAMI comme si le service se trouvait dans votre VPC, sans passerelle Internet, périphérique NAT, connexion VPN ou AWS Direct Connect connexion. Les instances de votre VPC ne nécessitent pas d'adresses IP publiques pour accéder à des autorisations vérifiées.

Vous établissez cette connexion privée en créant un point de terminaison d'interface à technologie AWS PrivateLink. Nous créons une interface réseau du point de terminaison dans chaque sous-réseau que vous activez pour le point de terminaison d'interface. Il s'agit d'interfaces réseau gérées par le demandeur qui servent de point d'entrée pour le trafic destiné

Pour plus d'informations, consultez Accès à Services AWS via AWS PrivateLink dans le Guide AWS PrivateLink.

ConsidérConsidérConsidérConsidérConsidérConsidérConsidérConsidér

ConsidérConsidérConsidérConsidérConsidérConsidérConsidérConsidérConsidérConsidérConsidérConsidér AWS PrivateLink

PerAMI VeriA prend en charge l'exécution d'appels en direction de toutes ses actions d'API via le point de terminaison d'interface pour faire passer des appels

Les stratégies de point de point de terminaison d'un VPC ne sont pas prises en charge pour le service pour Par défaut, l'accès complet des autorisations vérifiées est autorisé via le point de terminaison d'interface via le point de terminaison d'interface. Vous pouvez également associer un groupe de sécurité aux interfaces réseau du point de terminaison afin de contrôler le trafic vers le point de terminaison d'interface pour contrôler le trafic vers le point de terminaison d'interface pour accéder à

Créer un point de terminaison d'interface pour des autorisations relatives

Vous pouvez créer un point de terminaison d'interface pour des autorisations vérifiées à l'aide de la console Amazon VPC ou de l'AWS Command Line Interface(AWS CLI). Pour de plus amples

informations, veuillez consulter Créer un point de terminaison d'interface dans le Guide AWS PrivateLink.

Création d'un point de terminaison d'interface pour des autorisations relatives

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

Si vous activez le DNS privé pour le point de terminaison d'interface, vous pouvez adresser des demandes d'API à l'aide de son nom DNS régional par défaut. Par exemple, `verifiedpermissions.us-east-1.amazonaws.com`.

Quotas pour les autorisations vérifiées par Amazon

Vous Compte AWS disposez de quotas par défaut, anciennement appelés limites, pour chaque AWS service. Sauf indication contraire, chaque quota est spécifique à la région. Vous pouvez demander des augmentations pour certains quotas, et d'autres quotas ne peuvent pas être augmentés.

Pour consulter les quotas des autorisations vérifiées, ouvrez la console Service Quotas. Dans le volet de navigation, choisissez les AWS services, puis sélectionnez Autorisations vérifiées.

Pour demander une augmentation de quota, consultez Demander une augmentation de quota dans le Guide de l'utilisateur de Service Quotas. Si le quota n'est pas encore disponible dans Service Quotas, utilisez le Formulaire d'augmentation de limite de service.

Vous Compte AWS disposez des quotas suivants relatifs aux autorisations vérifiées.

Rubriques

Quotas de ressources

Nom	Par défaut	Ajusta	Description
Boutiques Policy par région et par compte	Chaque Région prise en charge : 1 000	Oui	Le nombre maximum de magasins de politiques.
Modèles de politiques par magasin de politiques	Chaque Région prise en charge : 40	Oui	Nombre maximal de modèles de politique s dans un magasin de politiques.
Sources d'identité par magasin de politiques	1	Non	Nombre maximal de sources d'identité que vous pouvez définir pour un magasin de politiques.

Nom	Par défaut	Ajusta	Description
Taille de la demande d'autorisation[1]	1 Mo	Non	Taille maximale d'une demande d'autorisation.
Taille de la politique	10 000 octets	Non	Taille maximale d'une police individuelle.
Taille du schéma	100,000 bytes	Non	Taille maximale du schéma d'un magasin de politiques.
Taille de la politique par ressource	200 000 octets[2]	Non	Taille maximale de toutes les politiques qui font référence à une ressource spécifique.

[1] Le quota pour une demande d'autorisation est le même pour IsAuthorizedles deux IsAuthorizedWithToken.

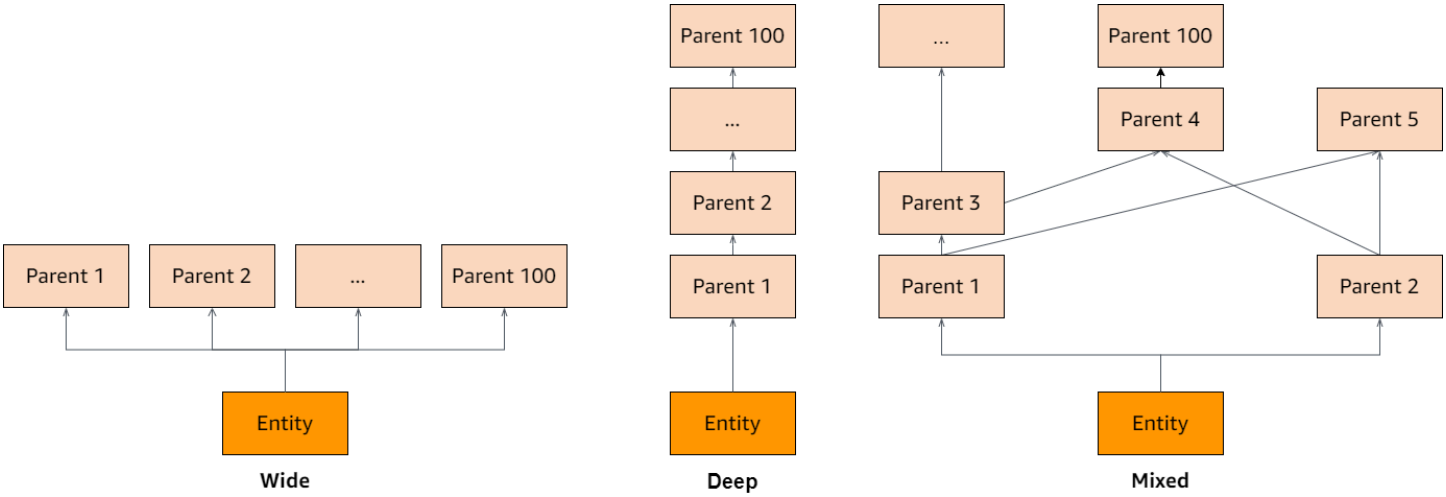
[2] La taille totale de toutes les politiques relatives à une seule ressource ne peut pas dépasser 200 000 octets. Pour les politiques liées à un modèle, la taille du modèle de stratégie n'est comptée qu'une seule fois, plus la taille de chaque ensemble de paramètres utilisé pour instancier chaque politique liée au modèle.

Quotas pour les hiérarchies

Nom	Par défaut	Ajusta	Description
Parents transitifs par directeur	100	Non	Le nombre maximum de parents transitifs pour chaque directeur.
Parents transitifs par action	100	Non	Le nombre maximum de parents transitifs pour chaque action.

Nom	Par défaut	Ajusta	Description
Parents transitifs par ressource	100	Non	Le nombre maximum de parents transitifs pour chaque ressource.

Le schéma ci-dessous illustre comment les parents transitifs peuvent être définis pour une entité (principal, action ou ressource).



Quotas d'opérations par seconde

Les autorisations vérifiées limitent les demandes adressées aux points de terminaison du service Région AWS lorsque les demandes d'application dépassent le quota d'une opération d'API. Les autorisations vérifiées peuvent renvoyer une exception lorsque vous dépassez le quota de demandes par seconde ou que vous tentez des opérations d'écriture simultanées. Vous pouvez consulter vos quotas RPS actuels dans Service Quotas. Pour empêcher les applications de dépasser le quota d'une opération, vous devez les optimiser en fonction des nouvelles tentatives et des retards exponentiels. Pour plus d'informations, consultez Réessayer avec un schéma d'interruption et Gestion et surveillance de la régulation des API dans vos charges de travail.

Nom	Par défaut	Ajusta	Description
BatchIsAuthorized demandes par seconde, par région et par compte	Chaque Région prise en charge : 30	Oui	Le nombre maximum de BatchIsAuthorized demandes par seconde.

Nom	Par défaut	Ajusta	Description
BatchIsAuthorizedWithToken demandes par seconde, par région et par compte	Chaque Région prise en charge : 30	Oui	Le nombre maximum de BatchIsAuthorizedWithToken demandes par seconde.
CreatePolicy demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de CreatePolicy demandes par seconde.
CreatePolicyStore demandes par seconde, par région et par compte	Par région prise en charge : 1	Non	Le nombre maximum de CreatePolicyStore demandes par seconde.
CreatePolicyTemplate demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de CreatePolicyTemplate demandes par seconde.
DeletePolicy demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de DeletePolicy demandes par seconde.
DeletePolicyStore demandes par seconde, par région et par compte	Par région prise en charge : 1	Non	Le nombre maximum de DeletePolicyStore demandes par seconde.
DeletePolicyTemplate demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de DeletePolicyTemplate demandes par seconde.
GetPolicy demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de GetPolicy demandes par seconde.
GetPolicyTemplate demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de GetPolicyTemplate demandes par seconde.

Nom	Par défaut	Ajusta	Description
GetSchema demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de GetSchema demandes par seconde.
IsAuthorized demandes par seconde, par région et par compte	Chaque région prise en charge : 200	Oui	Le nombre maximum de IsAuthorized demandes par seconde.
IsAuthorizedWithToken demandes par seconde, par région et par compte	Chaque région prise en charge : 200	Oui	Le nombre maximum de IsAuthorizedWithToken demandes par seconde.
ListPolicies demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de ListPolicies demandes par seconde.
ListPolicyStores demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de ListPolicyStores demandes par seconde.
ListPolicyTemplates demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de ListPolicyTemplates demandes par seconde.
PutSchema demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de PutSchema demandes par seconde.
UpdatePolicy demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de UpdatePolicy demandes par seconde.
UpdatePolicyTemplate demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Oui	Le nombre maximum de UpdatePolicyTemplate demandes par seconde.

Historique du document pour le guide de l'utilisateur Amazon Verified Permissions

Le tableau suivant décrit les versions de documentation relatives aux autorisations vérifiées.

Modification	Description	Date
Sources d'identité OIDC	Vous pouvez désormais autoriser les utilisateurs des fournisseurs d'identité OpenID Connect (OIDC).	8 juin 2024
Autorisation par lots avec jetons de source d'identité	Vous pouvez désormais autoriser les utilisateurs d'un groupe d'utilisateurs Amazon Cognito en une seule demande d'`BatchIsAuthorizedWithToken` API.	5 avril 2024
Création d'un magasin de politiques avec API Gateway	Vous pouvez désormais créer un magasin de politiques à partir d'une API existante et d'un pool d'utilisateurs Amazon Cognito.	1er avril 2024
Concepts contextuels et exemple	Ajout d'informations sur le contexte des demandes d'autorisation avec autorisations vérifiées.	1 février 2024
Concepts d'autorisation et exemple	Ajout d'informations sur les demandes d'autorisation avec autorisations vérifiées.	1 février 2024
AWS CloudFormation intégration	Verified Permissions prend en charge la création de sources d'identité, de politiques, de	30 juin 2023

	magasins de politiques et de modèles de politiques dans AWS CloudFormation.	
Première version	Publication initiale du guide de l'utilisateur Amazon Verified Permissions	13 juin 2023

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.