



Guide de l'utilisateur

Amazon Verified Permissions



Amazon Verified Permissions: Guide de l'utilisateur

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce que les autorisations vérifiées par Amazon ?	1
Autorisation dans les autorisations vérifiées	1
Langage politique du cèdre	2
Avantages des autorisations vérifiées	2
Accélérez le développement des applications	2
Applications plus sécurisées	3
Fonctionnalités destinées à l'utilisateur final	3
Services connexes	3
Accès aux autorisations vérifiées	3
Tarification des autorisations vérifiées	5
Commencer à utiliser Policy Stores	7
Conditions préalables	8
Inscrivez-vous pour un Compte AWS	8
Étape 1 : créer un magasin PhotoFlash de politiques	8
Étape 2 : créer une politique	9
Étape 3 : Test d'un magasin de politiques	10
Étape 4 : Nettoyer les ressources	11
Conception d'un modèle d'autorisation	13
Il n'y a pas de modèle correct	14
Erreurs de renvoi	15
Concentrez-vous sur les ressources	15
Envisagez la multilocation	17
Comparaison des magasins de politiques partagés et des magasins de politiques par locataire	18
Comment choisir	20
Boutiques Policy	21
Création de magasins de politiques	22
Création d'un magasin de politiques à l'aide de Rust	31
Magasins de politiques liés à l'API	36
Comment ça marche	38
Considérations	40
Ajouter ABAC	41
Passage à la production	42
Résolution des problèmes	45

Supprimer des magasins de politiques	48
Alias du Policy Store	51
Propriétés des alias du Policy Store	51
Création d'alias de magasin de politiques	55
Récupération des alias du magasin de politiques	56
Obtenir un alias de Policy Store	56
Politique de référencement : alias de magasin	56
Supprimer les alias du Policy Store	58
Suppression progressive d'un alias de magasin de politiques	59
Suppression définitive d'un alias de magasin de politiques	59
Utilisation d'alias du Policy Store	60
Utilisation des alias du Policy Store dans Operations	61
Utilisation des alias du Policy Store Across Régions AWS	61
Contrôle de l'accès	62
autorisations vérifiées : CreatePolicyStoreAlias	63
autorisations vérifiées : GetPolicyStoreAlias	64
autorisations vérifiées : ListPolicyStoreAliases	64
autorisations vérifiées : DeletePolicyStoreAlias	64
Limitation des autorisations relatives aux alias du magasin Policy	65
Schéma du Policy Store	67
Schéma d'édition	69
Mode de validation des politiques	72
Stratégies	74
Création de politiques statiques	75
Modification de politiques statiques	78
.....	81
Évaluer un exemple de contexte	83
Politiques de test	89
Exemples de politiques	92
Utilise la notation entre crochets pour référencer les attributs des jetons	92
Utilise la notation par points pour référencer les attributs	93
Reflète Amazon Cognito Attributs des jetons d'identification	93
Reflète les attributs du jeton d'identification OIDC	94
Reflète Amazon Cognito attributs du jeton d'accès	94
Reflète les attributs du jeton d'accès OIDC	94
Modèles de politiques et politiques liées à des modèles	96

Création de modèles de politiques	97
Création de politiques liées à un modèle	99
Modification de modèles de politiques	101
Exemples de politiques liées à un modèle	103
PhotoFlash exemples	104
DigitalPetStore exemples	105
TinyToDo exemples	105
Sources d'identité	107
Choisir le bon fournisseur d'identité	108
Travailler avec des sources Amazon Cognito d'identité	108
Création de sources d'identité	111
Modification des sources d'identité	115
Mappage des jetons au schéma	117
Validation du client et du public	129
Travailler avec les sources d'identité OIDC	132
Création de sources d'identité	133
Modification des sources d'identité	136
Mappage des jetons au schéma	138
Validation du client et du public	146
Intégrations	150
Utilisation d'Express	150
Conditions préalables	151
Configuration de l'intégration	151
Configuration de l'autorisation	152
Implémentation du middleware d'autorisation	155
Tester l'intégration	156
Résolution des problèmes	156
Étapes suivantes	156
Autoriser les demandes	157
opérations d'API	158
Modèle de test	159
Intégration aux applications	161
Effectuer des demandes d'API	164
Points de terminaison d'autorisations vérifiés	164
Paramètres de demande	164
Identifiants de demande	164

Authentification par API de requête	164
Bibliothèques disponibles	164
Effectuer des demandes d'API à l'aide de la méthode POST	165
Sécurité	168
Protection des données	168
Chiffrement des données	170
Clés gérées par le client	171
Gestion des identités et des accès	190
Public ciblé	191
Authentification par des identités	191
Gestion de l'accès à l'aide de politiques	194
Comment fonctionne Amazon Verified Permissions avec IAM	196
IAM politiques relatives aux autorisations vérifiées	202
Exemples de politiques basées sur l'identité	205
AWS politiques gérées	208
Résolution des problèmes	212
Validation de conformité	214
Résilience	214
Contrôle	216
CloudTrail journaux	216
Informations sur les autorisations vérifiées dans CloudTrail	217
Comprendre les entrées du fichier journal des autorisations vérifiées	218
Travailler avec AWS CloudFormation	236
Autorisations et CloudFormation modèles vérifiés	236
AWS Constructions CDK	237
En savoir plus sur CloudFormation	237
En utilisant AWS PrivateLink	238
Considérations	238
Création d'un point de terminaison d'interface	238
Création d'une politique de point de terminaison	239
Quotas	241
Quotas de ressources	241
Template-linked exemple de taille de politique	243
Quotas pour les hiérarchies	244
Quotas d'opérations par seconde	245
Termes et concepts	251

Modèle d'autorisation	252
Demande d'autorisation	252
Réponse d'autorisation	252
Politiques prises en compte	253
Données contextuelles	253
Déterminer les politiques	253
Données sur les entités	253
Permissions, autorisations et principes	253
Application des politiques	254
Boutique Policy	254
Alias du magasin Policy	254
Nom de la politique	255
Nom du modèle de politique	255
Politiques satisfaites	255
Différences avec le cèdre	255
Support des modèles de politiques	256
Support de schéma	256
Définition des groupes d'action	256
Formatage des entités	257
Limites de longueur et de taille	262
FAQ sur Cedar v4	263
Pourquoi est-ce que je reçois un message d'erreur indiquant que Cedar 2 n'est plus pris en charge lorsque j'appelle Amazon Verified Permissions ?	264
Pourquoi certaines politiques, certains modèles de politiques et certains schémas ne sont-ils pas compatibles avec Cedar 4 ?	264
Comment savoir si ma boutique utilise Cedar 2 ou Cedar 4 ?	264
Comment passer à Cedar 4 ?	266
Puis-je rétrograder ma boutique en ligne de Cedar 4 à Cedar 2 ?	267
Pourquoi est-ce que je reçois un message d'erreur indiquant que mon magasin de politiques est configuré pour Cedar 2 ?	267
Comment rendre mon schéma compatible avec Cedar 4 ?	267
Comment puis-je rendre mes politiques et mes modèles compatibles avec Cedar 4 ?	269
Historique de la documentation	270
.....	cclxxii

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 politique de Cedar pour définir des autorisations précises afin de protéger les ressources de votre application.

Pour obtenir des conseils et des exemples sur la configuration d'un point de décision politique (PDP) à l'aide d'autorisations vérifiées, consultez [Implémentation d'un PDP en utilisant les autorisations vérifiées d'Amazon dans les](#) directives AWS prescriptives.

Rubriques

- [Autorisation dans les autorisations vérifiées](#)
- [Langage politique du cèdre](#)
- [Avantages des autorisations vérifiées](#)
- [Services connexes](#)
- [Accès aux autorisations vérifiées](#)
- [Tarification des autorisations vérifiées](#)

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 votre application. 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 que ou une autre solution Amazon Cognito d'authentification. Les autorisations vérifiées sont indépendantes de l'endroit où le principal est géré et de la manière dont il a été authentifié.

Les autorisations vérifiées sont un service qui permet aux clients de créer, de gérer et de tester des politiques dans le AWS Management Console, par le biais de programmes à l'aide des autorisations vérifiées APIs, ou par le biais de solutions de code telles que l'infrastructure. CloudFormation Les autorisations sont exprimées en utilisant le langage de politique de Cedar. L'application cliente

appelle l'autorisation APIs 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 principaux autorisés, les utilisateurs humains ou les machines peuvent accéder à l'application et ne peuvent faire que ce pour quoi ils sont autorisés. Avec Cedar, vous pouvez dissocier votre logique métier de la logique d'autorisation. Dans le code de votre application, vous devez préfacier 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 4.7 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.

Verified Permissions propose des intégrations avec les frameworks de développement les plus courants, ce qui facilite la mise en œuvre de l'autorisation dans vos applications avec un minimum de modifications de code. Ces intégrations vous permettent de vous concentrer sur votre logique métier principale tandis que Verified Permissions gère les décisions d'autorisation.

- Express.js : intégration basée sur un intergiciel qui vous permet de protéger les points de terminaison d'API dans vos applications Express sans modifier les gestionnaires de route existants. Pour de plus amples informations, veuillez consulter [the section called “Utilisation d'Express”](#).

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 et les AWS informations d'identification OAuth 2.0. Lorsque vous créez un magasin de politiques, vous avez la possibilité de créer vos principaux et vos groupes à partir d'un groupe Amazon Cognito d'utilisateurs. 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 protocoles REST, HTTP, et ce, WebSocket APIs 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 à ce sujet API Gateway, consultez le [guide du API Gateway développeur](#).
- 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 Outils AWS pour PowerShell l'utilisateur](#).

- [autorisations vérifiées](#) dans la référence des commandes AWS CLI
- [Autorisations vérifiées par Amazon](#) dans AWS Tools for Windows PowerShell

AWS SDKs

AWS fournit SDKs (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.). Ils SDKs fournissent un moyen pratique de créer un accès programmatique aux autorisations vérifiées et AWS. Par exemple, ils se SDKs chargent de tâches telles que la signature cryptographique des demandes, la gestion des erreurs et le renouvellement automatique des demandes.

Pour en savoir plus et télécharger AWS SDKs, consultez la section [Outils pour Amazon Web Services](#).

Vous trouverez ci-dessous des liens vers la documentation des différentes ressources relatives aux autorisations vérifiées AWS SDKs.

- [AWS SDK pour .NET](#)
- [AWS SDK pour C++](#)
- [AWS SDK pour Go](#)
- [AWS SDK pour Java](#)
- [AWS SDK pour JavaScript](#)

- [AWS SDK pour PHP](#)
- [AWS SDK pour Python \(Boto\)](#)
- [AWS SDK pour Ruby](#)
- [AWS SDK pour Rust](#)

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. CloudFormation Des constructions, ou des composants cloud réutilisables, peuvent être utilisés pour créer des CloudFormation modèles. Ces modèles peuvent ensuite être utilisés pour déployer votre infrastructure cloud.

Pour en savoir plus et télécharger le AWS 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](#).

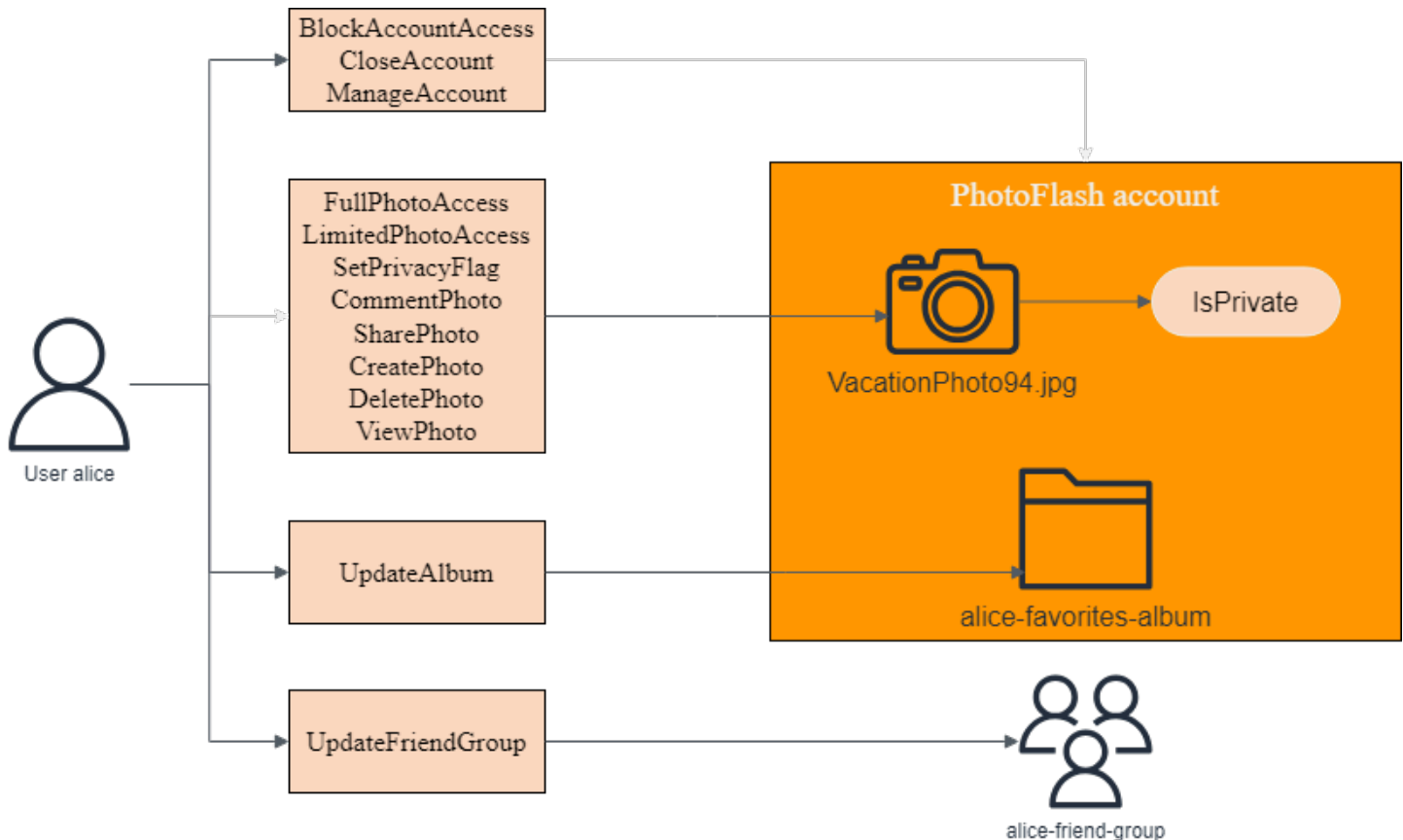
Si vous avez des questions concernant AWS la facturation, les comptes et les événements, [contactez Support](#).

Créez votre première boutique consacrée à la politique d'autorisations vérifiées par Amazon

Pour ce didacticiel, supposons que vous êtes le développeur d'une application de partage de photos et que vous recherchez un moyen de contrôler les actions que les utilisateurs de l'application peuvent effectuer. Vous souhaitez contrôler qui peut ajouter, supprimer ou afficher des photos et des albums photos. Vous souhaitez également contrôler les actions qu'un utilisateur peut effectuer sur son compte. Peuvent-ils gérer leur compte, que diriez-vous du compte d'un ami ? Pour contrôler ces actions, vous devez créer des politiques qui autorisent ou interdisent ces actions en fonction de l'identité de l'utilisateur. Verified Permissions propose [des magasins de politiques](#), ou conteneurs, pour héberger ces politiques.

Dans ce didacticiel, nous allons vous expliquer comment créer un exemple de boutique de politiques à l'aide de la console Amazon Verified Permissions. La console propose quelques exemples d'options de magasin de politiques et nous allons créer un magasin PhotoFlashde politiques. Ce magasin de règles permet aux principaux, tels que les utilisateurs, d'effectuer des actions, telles que le partage, sur des ressources, telles que des photos ou des albums.

Le schéma suivant illustre les relations entre une directrice et `User::alice` les actions qu'elle peut effectuer sur diverses ressources, à savoir son PhotoFlash compte, le `VactionPhoto94.jpg` fichier, l'album `alice-favorites-album photo` et le groupe d'utilisateurs `alice-friend-group`.



Maintenant que vous connaissez le magasin de PhotoFlashpolitiques, créons-le et explorons-le.

Conditions préalables

Inscrivez-vous pour un Compte AWS

Pour commencer AWS, vous avez besoin d'un Compte AWS. Pour plus d'informations sur la création d'un Compte AWS, voir [Getting started with an Compte AWS](#) dans le Guide de Gestion de compte AWS référence.

Étape 1 : créer un magasin PhotoFlash de politiques

Dans la procédure suivante, vous allez créer un magasin de PhotoFlashrègles à l'aide de la AWS console.

Pour créer un magasin PhotoFlash de politiques

1. Dans la [console Verified Permissions](#), choisissez Create new policy store.

2. Pour les options de démarrage, choisissez Démarrer à partir d'un exemple de magasin de politiques.
3. Pour Exemple de projet, sélectionnez PhotoFlash.
4. Choisissez Create Policy Store.

Lorsque le message « Magasin de politiques créé et configuré » s'affiche, choisissez Accéder à l'aperçu pour explorer votre magasin de politiques.

Étape 2 : créer une politique

Lorsque vous avez créé le magasin de politiques, une politique par défaut a été créée pour permettre aux utilisateurs d'avoir un contrôle total sur leurs propres comptes. Il s'agit d'une politique utile, mais pour nos besoins, créons une politique plus restrictive afin d'explorer les nuances des autorisations vérifiées. Si vous vous souvenez du schéma que nous avons examiné plus tôt dans le didacticiel `User::alice`, nous avons un directeur qui pouvait effectuer une action sur une ressource `alice-favorites-album`. `UpdateAlbum` Ajoutons la politique qui permettra à Alice, et uniquement à Alice, de gérer cet album.

Pour créer une politique

1. Dans la [console Verified Permissions](#), choisissez le magasin de politiques que vous avez créé à l'étape 1.
2. Dans la navigation, sélectionnez Politiques.
3. Choisissez Créer une politique, puis sélectionnez Créer une politique statique.
4. Pour l'effet de la politique, choisissez Permettre.
5. Dans le champ d'application des principes, sélectionnez Principal spécifique, puis pour Spécifier le type d'entité, choisissez PhotoFlash: :User, et pour Spécifier l'identifiant de l'entité, entrez **alice**.
6. Pour Étendue des ressources, choisissez Ressource spécifique, puis pour Spécifier le type d'entité, choisissez PhotoFlash: :Album, et pour Spécifier l'identifiant de l'entité, entrez **alice-favorites-album**.
7. Dans le champ d'application des actions, choisissez Ensemble d'actions spécifique, puis pour les actions auxquelles cette politique doit s'appliquer, sélectionnez UpdateAlbum.
8. Choisissez Suivant.

9. Sous Détails, pour Description de la politique, entrez facultatif **Policy allowing alice to update alice-favorites-album..**
10. Choisir Create policy (Créer une stratégie)

Maintenant que vous avez créé une politique, vous pouvez la tester dans la console des autorisations vérifiées.

Étape 3 : Test d'un magasin de politiques

Après avoir créé votre magasin de politiques et votre politique, vous pouvez les tester en exécutant une [demande d'autorisation](#) simulée à l'aide du banc de test des autorisations vérifiées.

Pour tester les politiques du magasin de politiques

1. Ouvrez la [console des autorisations vérifiées](#). Choisissez votre magasin de polices.
2. Dans le volet de navigation de gauche, choisissez Test bench.
3. Choisissez le mode visuel.
4. Pour Principal, procédez comme suit :
 - a. Pour que le principal passe à l'action, choisissez PhotoFlash: :User et pour Spécifier l'identifiant de l'entité, entrez **alice**.
 - b. Sous Attributs, pour Compte : Entité, assurez-vous que l'entité PhotoFlash::Account est sélectionnée, et pour Spécifier l'identifiant de l'entité, entrez **alice-account**.
5. Sous Ressource, pour Ressource sur laquelle le principal agit, choisissez le type de ressource PhotoFlash::Album et pour Spécifier l'identifiant de l'entité, entrez **alice-favorites-album**.
6. Pour Action, choisissez PhotoFlash: :Action : : » UpdateAlbum "dans la liste des actions valides.
7. En haut de la page, choisissez Exécuter une demande d'autorisation pour simuler la demande d'autorisation pour les politiques Cedar dans le magasin d'exemples de politiques. Le banc de test doit afficher Décision : Autoriser pour indiquer que notre politique fonctionne comme prévu.

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 sur la politique que vous avez créée à l'étape 2.

Valeur principale	Compte principal : valeur de l'entité	Valeur des ressources	Valeur parente de la ressource	Action	Décision d'autorisation
PhotoFlas h: :Utilisateur bob	PhotoFlas h: :Compte alice-account	PhotoFlash: : Album alice-favorit-album	N/A	PhotoFlas h: :Action : « » UpdateAlbum	Refuser
PhotoFlas h: :Utilisateur alice	PhotoFlas h: :Compte alice-account	PhotoFlas h: :Foto photo.jpeg	PhotoFlas h: :Compte bob-account	PhotoFlas h: :Action : « » ViewPhoto	Refuser
PhotoFlas h: :Utilisateur alice	PhotoFlas h: :Compte alice-account	PhotoFlas h: :Foto photo.jpeg	PhotoFlas h: :Compte alice-account	PhotoFlas h: :Action : « » ViewPhoto	Allow
PhotoFlas h: :Utilisateur alice	PhotoFlas h: :Compte alice-account	PhotoFlas h: :Foto bob-photo.jpeg	PhotoFlas h: :Album Bob-Vacation-Album	PhotoFlas h: :Action : « » DeletePhoto	Refuser

Étape 4 : Nettoyer les ressources

Une fois que vous avez fini d'explorer votre magasin de politiques, supprimez-le.

Pour supprimer un entrepôt de politiques

1. Dans la [console Verified Permissions](#), choisissez le magasin de politiques que vous avez créé à l'étape 1.
2. Dans le menu de navigation, sélectionnez Réglages.
3. Sous Supprimer le magasin de règles, choisissez Supprimer ce magasin de politiques.

4. Dans le magasin Supprimer cette politique ? dans une boîte de dialogue, entrez Supprimer, puis choisissez Supprimer.

Bonnes pratiques pour la conception d'un modèle d'autorisation

Lorsque vous vous préparez à utiliser le service Amazon Verified Permissions au sein d'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. 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 ? Comment sont-ils organisés ? Par exemple, les fichiers se trouvent-ils dans un dossier ?
- L'organisation des ressources joue-t-elle un rôle dans le modèle d'autorisations ?
- 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 rôles sont-ils globaux ou délimités ? Par exemple, un « opérateur » est-il limité au sein d'un seul tenant, ou le terme « opérateur » signifie-t-il un opérateur pour l'ensemble de l'application ?
- 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 peuvent faire 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.

Pour répondre aux questions et parvenir à un modèle optimal, procédez comme suit :

- Passez en revue [les modèles de conception de Cedar](#) dans le guide de référence sur le langage politique de Cedar.
- Tenez compte des [meilleures pratiques décrites](#) dans le Guide de référence du langage politique de Cedar.
- Tenez compte des meilleures pratiques présentées sur cette page.

Bonnes pratiques

- [Il n'existe pas de modèle canonique « correct »](#)
- [Renvoi 403 erreurs interdites plutôt que 404 erreurs non trouvées](#)
- [Concentrez-vous sur vos ressources au-delà des opérations d'API](#)
- [Considérations relatives à la location multiple](#)

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 ce n'est pas grave. 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.

Renvoie 403 erreurs interdites plutôt que 404 erreurs non trouvées

Il est préférable de renvoyer une erreur 403 Forbidden aux demandes qui incluent une entité, en particulier une ressource, qui ne correspond à aucune politique plutôt qu'une erreur 404 Not found. Cela fournit le plus haut niveau de sécurité, car vous n'indiquez pas si une entité existe ou non, mais simplement que la demande ne respectait les conditions d'aucune politique du magasin de politiques.

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

Dans la plupart des applications, les autorisations sont modélisées en fonction des ressources prises en charge. 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 `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.

Toutefois, au fur et à mesure que vous développez votre modèle, cette approche axée sur les API risque de ne pas convenir aux applications dotées de modèles d'autorisation très précis, car il s'agit simplement d'un proxy de ce que vos clients essaient réellement de protéger : les données et les ressources sous-jacentes. Si plusieurs personnes APIs contrôlent l'accès aux mêmes ressources, il peut être difficile pour les administrateurs de déterminer 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 de vous concentrer 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	Permissions
<code>listMembersOfGroup</code>	nécessite une <code>viewGroupMembership</code> autorisation sur le groupe
<code>listGroupMembershipsForUser</code>	nécessite <code>viewGroupMembership</code> l'autorisation de l'utilisateur
<code>isUserInGroups</code>	nécessite <code>viewGroupMembership</code> 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.

Considérations relatives à la location multiple

Vous souhaitez 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, élaborer 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. Pour plus d'informations, consultez les [considérations relatives à la conception multi-locataires d'Amazon Verified Permissions](#) dans les directives AWS prescriptives.

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 politiques 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 n'aura d'impact important sur votre AWS facture. 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

Lorsque votre application possède plusieurs magasins de politiques, 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 souhaitez 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 plus d'informations sur les quotas, consultez [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 politiques partagés et par locataire.

Considération	Niveau d'effort dans un magasin de politiques partagé	Niveau d'effort dans les magasins de politiques par locataire
Isolement des politiques relatives aux locataires	Moyen. Doit inclure les identifiants des locataires dans les politiques et les demandes d'autorisation.	Faible. L'isolation est le comportement par défaut. Les politiques spécifiques aux locataires ne sont

pas accessibles aux autres locataires.

Flux d'autorisation	Faible. Toutes les requêtes ciblent un magasin de politiques.	Moyen. Doit maintenir les mappages entre chaque locataire et son ID de magasin de politiques.
Gestion des modèles et des schémas	Faible. Doit faire fonctionner un schéma pour tous les locataires.	Haut. Les schémas et les modèles peuvent être moins complexes individuellement, mais les modifications nécessitent plus de coordination et de complexité.
Gestion des politiques globales	Faible. Toutes les politiques sont globales et peuvent être mises à jour de manière centralisée.	Haut. Vous devez ajouter des politiques globales à chaque magasin de politiques lors de l'intégration. Répliquez les mises à jour des politiques globales entre de nombreux magasins de politiques.
Débarquement du locataire	Haut. Doit identifier et supprimer uniquement les politiques spécifiques au locataire.	Faible. Supprimez le magasin de politiques.
Quotas de ressources de service	Haut. Les locataires partagent des quotas de ressources qui affectent les magasins de politiques, tels que la taille du schéma, la taille des politiques par ressource et les sources d'identité par magasin de politiques.	Faible. Chaque locataire dispose de quotas de ressources dédiés.

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.

Mais 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 débarquement 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.

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. Dans chaque magasin de politiques, vous pouvez créer un schéma qui est utilisé pour valider les politiques ajoutées au magasin de politiques. En outre, vous pouvez activer la validation des politiques. Si vous ajoutez une politique à un magasin de politiques avec la validation des politiques activée, les types d'entités, les types courants et les actions définis dans la politique sont validés par rapport au schéma et les politiques non valides sont rejetées.

La protection contre la suppression empêche la suppression accidentelle d'un magasin de politiques. La protection contre la suppression est activée sur tous les nouveaux magasins de politiques créés via le AWS Management Console. En revanche, il est désactivé pour tous les magasins de politiques créés via un appel d'API ou de SDK.

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](#). Vous pouvez également créer des alias de magasin de politiques pour faire référence à vos magasins de politiques par des noms conviviaux. Pour de plus amples informations, veuillez consulter [Alias de la boutique Amazon Verified Permissions Policy](#).

Nous vous recommandons d'utiliser des espaces de noms pour les entités Cedar dans vos magasins de politiques 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. Par exemple `MyApplicationNamespace::exampleType`. Les autorisations vérifiées prennent en charge jusqu'à 100 espaces de noms par magasin de politiques. Ces espaces de noms permettent de garder les choses en ordre lorsque vous travaillez avec plusieurs applications similaires. Par exemple, dans les applications à locataires multiples, l'utilisation d'un espace de noms pour ajouter le nom du locataire aux types définis dans le schéma les distinguera de leurs homologues similaires utilisés par les autres locataires. En consultant les journaux des demandes d'autorisation, vous pourrez facilement identifier le locataire qui a traité la demande d'autorisation. 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](#)

- [Magasins de politiques liés à l'API](#)
- [Supprimer des magasins de politiques](#)

Création de magasins de politiques d'autorisations vérifiées

Vous pouvez créer un magasin de règles à l'aide des 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 une source d'identité 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 avec l'appartenance à un groupe d'utilisateurs ou d'autres attributs.
- 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 règles à l'aide du Configuration guidée méthode de configuration

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 serez en mesure d'ajouter des éléments à votre magasin de politiques, d'étendre le schéma pour décrire d'autres types de ressources et de principaux types, et de 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 l'application Web Weather Updates.
4. Dans la section Détails, saisissez un espace de noms pour votre schéma. Pour plus d'informations sur les espaces de noms, consultez la section Espaces de [noms](#) dans la documentation de Cedar.
5. Choisissez Suivant.
6. Dans la fenêtre Type de ressource, saisissez le nom de votre type de ressource. Par exemple, il `currentTemperature` peut s'agir d'une ressource pour l'application Web Weather Updates.
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. Par exemple, il `temperatureFormat` peut s'agir d'un attribut de la `currentTemperature` ressource et être Fahrenheit ou Celsius. 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. Il `viewTemperature` peut s'agir, par exemple, d'une action dans l'application Web Weather Updates. 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. Par défaut, l'utilisateur est ajouté à ce champ mais peut être remplacé.
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. 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 Cognito User Pool 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 AWS et saisissez l'ID du groupe d' Amazon

Cognito utilisateurs 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.


- Choisissez un fournisseur OIDC externe si l'identifiant et les attributs du principal seront extraits d'un jeton d'and/or accès ID, généré par un fournisseur OIDC externe, puis ajoutez les détails du fournisseur et du jeton.

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. Il `user-id` peut s'agir, par exemple, d'un identifiant d'entité dans l'application **Web Weather Updates**.

 **Note**

Si vous utilisez Amazon Cognito, l'identifiant de l'entité doit être formaté comme `<userpool-id>|<sub>` suit.

- 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. Il `temperature-id` peut s'agir, par exemple, d'un identifiant d'entité dans l'application **Web Weather Updates**.
- 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 règles à l'aide du Configurez avec API Gateway et une source d'identité méthode de configuration

L' API Gateway option 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 Gateway API 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 Configurer avec API Gateway une source d'identité et sélectionnez Suivant.
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 les chemins ou méthodes de vos API dans la API Gateway console et sélectionnez Importer l'API pour voir les mises à jour.
 - 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 Amazon Cognito d'utilisateurs 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 API-linked d'autorisation.
 - 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 d' Amazon Cognito applications 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 un fournisseur OIDC externe :
 - 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 les sections [Mappage Amazon Cognito des jetons sur le schéma](#) et [Mappage des jetons OIDC sur le schéma](#).
 - c. (facultatif) Dans Réclamations de jetons - facultatif, choisissez Ajouter une réclamation de jeton, entrez le nom du jeton et sélectionnez un type de valeur.

- d. Dans Réclamations de jetons d'utilisateur et de groupe, procédez comme suit :
 - i. Entrez un nom de réclamation utilisateur dans le jeton 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.
 - ii. Entrez un nom de réclamation de groupe dans le jeton 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.
 - e. Dans Validation de l'audience, choisissez Add value et ajoutez une valeur que vous souhaitez que votre magasin de politiques accepte dans les demandes d'autorisation.
 - f. Choisissez Suivant.
7. Si vous le souhaitez 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 des applications, choisissez si vous souhaitez associer manuellement l'autorisateur Lambda ultérieurement ou si vous souhaitez que Verified Permissions le fasse pour vous dès maintenant et 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 Create policy store.
 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, si vous avez choisi d'associer l'autorisateur manuellement, 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 des autorisateurs [API Gateway Lambda dans le manuel Amazon API Gateway Developer Guide](#).


- 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 politiques. Ajoutez et affinez des politiques.

Sample policy store

Pour créer un magasin de règles à l'aide du Exemple de magasin de politiques méthode de configuration

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.
 - PhotoFlash est 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 n'importe qui 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 de compagnie 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 magasin DigitalPetStore d'exemples de politiques n'inclut pas de modèles de politiques. Les magasins de politiques PhotoFlashet TinyTodod'exemples incluent des modèles de politiques.

- TinyTodo est 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 d'autorisations Amazon Verified](#)

Empty policy store

Pour créer un magasin de règles à l'aide du Boutique de politiques vide méthode de configuration

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, consultez [Schéma 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](#) et [Création de politiques liées au modèle Amazon Verified Permissions](#).

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, voir [Schéma de la boutique Amazon Verified Permissions Policy](#).
- Pour ajouter des politiques, consultez [Création de politiques statiques relatives aux autorisations vérifiées par Amazon](#).
- Pour ajouter des modèles de politique, consultez [Création de modèles de politique d'autorisations vérifiées par Amazon](#).

```
$ aws verifiedpermissions create-policy-store \  
  --validation-settings "mode=STRICT" \  
{  
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/  
PEXAMPLEabcdefg111111",  
  "createdDate": "2023-05-16T17:41:29.103459+00:00",  
  "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",  
  "policyStoreId": "PEXAMPLEabcdefg111111"  
}
```

AWS SDKs

Vous pouvez créer un magasin de politiques à l'aide de l'`CreatePolicyStoreAPI`. Pour plus d'informations, consultez le guide [CreatePolicyStore](#) de référence de l'API Amazon Verified Permissions.

Implémentation des autorisations Amazon Verified dans Rust avec le AWS SDK

Cette rubrique fournit un exemple pratique de mise en œuvre des autorisations Amazon Verified Permissions dans Rust avec le AWS SDK. Cet exemple montre comment développer un modèle d'autorisation permettant de tester si un utilisateur est en mesure de voir une photo. L'exemple de code utilise la [aws-sdk-verifiedpermissions](#) caisse du [AWS SDK pour Rust](#), qui propose un ensemble robuste d'outils avec lesquels interagir Services AWS.

Conditions préalables

Avant de commencer, assurez-vous que la [AWS CLI](#) est configurée sur votre système et que vous connaissez Rust.

- Pour obtenir des instructions sur l'installation de la CLI AWS CLI, consultez le [guide d'installation de la AWS CLI](#).
- Pour obtenir des instructions sur la configuration du AWS CLI, consultez [les sections Configuration des paramètres AWS CLI](#) et [Paramètres des fichiers de configuration et d'identification dans le AWS CLI](#).
- Pour plus d'informations sur Rust, consultez [rust-lang.org](#) et le Guide du développeur du [AWS SDK pour Rust](#).

Une fois votre environnement préparé, voyons comment implémenter les autorisations vérifiées dans Rust.

Testez l'exemple de code

L'exemple de code effectue les opérations suivantes :

- Configure le client SDK pour communiquer avec AWS
- Crée un [magasin de politiques](#)
- Définit la structure du magasin de politiques en ajoutant un [schéma](#)
- Ajoute une [politique](#) pour vérifier les demandes d'autorisation
- Envoie une [demande d'autorisation](#) de test pour vérifier que tout est correctement configuré

Pour tester l'exemple de code

1. Créez un projet Rust.
2. Remplacez tout code existant `main.rs` par le code suivant :

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

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

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

//Function that creates a policy in the policy store in the client
async fn create_policy(client: &Client, ps_id: &str,
    policy_definition:&PolicyDefinition) -> CreatePolicyOutput {
    let create_policy =
    client.create_policy().definition(policy_definition.clone()).policy_store_id(ps_id).send().await;
    return create_policy.unwrap();
}
```

```

    return create_policy.unwrap();
}

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

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

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

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

//Add schema to policy store
    let schema= r#{
        "PhotoFlash": {
            "actions": {
                "ViewPhoto": {
                    "appliesTo": {
                        "context": {
                            "type": "Record",
                            "attributes": {}
                        },
                    },
                    "principalTypes": [
                        "User"
                    ],
                    "resourceTypes": [

```

```

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

//Create policy
let policy_text = r#"
  permit (
    principal in PhotoFlash::User::"alice",
    action == PhotoFlash::Action::"ViewPhoto",
    resource == PhotoFlash::Photo::"VacationPhoto94.jpg"
  );
"#;

```

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

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

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

3. Exécutez le code en entrant `cargo run` dans le terminal.

Si le code s'exécute correctement, le terminal s'affichera, `Decision: Allow` suivi de l'ID de politique de la politique déterminante. Cela signifie que vous avez créé avec succès un magasin de politiques et que vous l'avez testé à l'aide du AWS SDK pour Rust.

nettoyer des ressources ;

Une fois que vous avez fini d'explorer votre magasin de politiques, supprimez-le.

Pour supprimer un entrepôt de politiques

Vous pouvez supprimer un magasin de politiques en utilisant cette `delete-policy-store` opération, en le `PSEXAMPLEabcdefgh11111` remplaçant par l'ID du magasin de politiques que vous souhaitez supprimer.

```
$ aws verifiedpermissions delete-policy-store \  
  --policy-store-id PSEXAMPLEabcdefgh11111
```

Cette commande ne produit aucune sortie si elle réussit.

Magasins de politiques liés à l'API

Un cas d'utilisation courant consiste à utiliser les autorisations vérifiées par Amazon pour autoriser l'accès des utilisateurs à APIs l'hébergement sur Amazon API Gateway. À l'aide d'un assistant intégré à la AWS console, vous pouvez créer des politiques d'accès basées sur les rôles pour les utilisateurs gérés dans [Amazon Cognito](#) ou dans n'importe quel fournisseur d'identité OIDC (IdP), et déployer AWS Lambda un autorisateur qui appelle Verified Permissions pour évaluer ces politiques.

Pour terminer l'assistant, choisissez Configurer avec un fournisseur API Gateway d'identité lorsque vous [créez un nouveau magasin de politiques](#) et suivez les étapes indiquées.

Un magasin de politiques lié à une API est créé et fournit votre modèle d'autorisation et vos ressources pour les demandes d'autorisation. Le magasin de politiques possède une source d'identité et un autorisateur Lambda qui se connecte API Gateway aux autorisations vérifiées. Une fois le magasin de politiques créé, 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 portées OAuth 2.0 à l'aide du langage de politique [Cedar](#). Par exemple, les autorisations vérifiées ne peuvent accorder l'accès qu'aux utilisateurs possédant un `email` attribut dans le `domainemycompany.co.uk`.

Une fois que vous avez configuré 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 voir une démo, consultez [Amazon Verified Permissions - Quick Start Overview and Demo](#) sur la Amazon Web Services YouTube chaîne.

Rubriques

- [Comment les autorisations vérifiées autorisent-elles les demandes d'API](#)
- [Considérations relatives aux magasins de politiques liés aux API](#)
- [Ajouter un contrôle d'accès basé sur les attributs \(ABAC\)](#)
- [Passage à la production avec AWS CloudFormation](#)
- [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 Configurer avec API Gateway une source d'identité dans la console des autorisations vérifiées 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 par programmation avec le [AWS Cloud Development Kit \(AWS CDK\)](#) Pour de plus amples informations, veuillez consulter [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'[IsAuthorizedWithToken](#) API. 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.

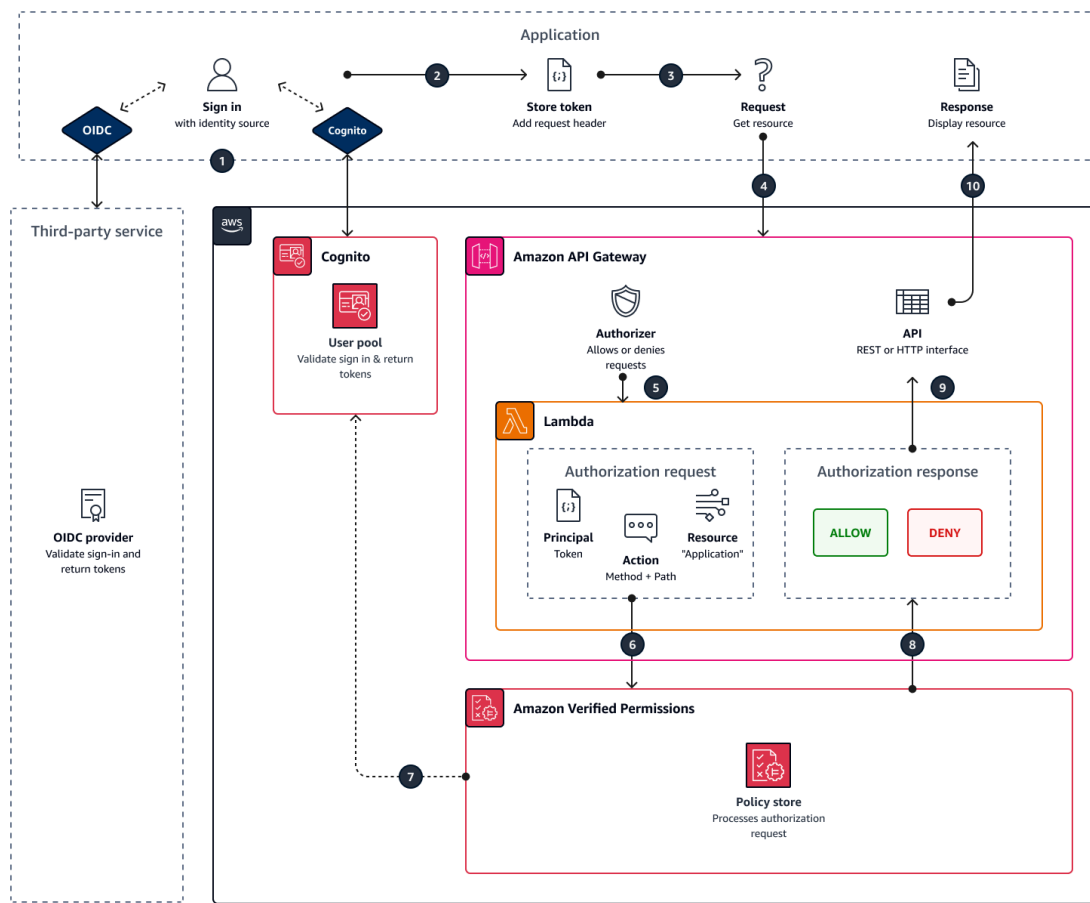
Composants de la source d'identité et de API Gateway l'autorisation avec autorisations vérifiées

- Un groupe [Amazon Cognito](#) d'utilisateurs 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 Gateway](#) API REST. 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 Configurer avec API Gateway une source d'identité crée automatiquement cet autorisateur Lambda pour vous.
- Un magasin de politiques d'autorisations vérifiées. La source d'identité du magasin de politiques est votre groupe Amazon Cognito d'utilisateurs ou votre groupe de fournisseurs OIDC. 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 API Gateway une source d'identité, Verified Permissions crée un schéma et des politiques de 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.](#)



1. Votre utilisateur se connecte à votre application via Amazon Cognito un autre IdP OIDC. L'IdP émet des identifiants et des jetons d'accès contenant les informations de l'utilisateur.
2. Votre application enregistre le JWTs. Pour plus d'informations, consultez la section [Utilisation de jetons avec des groupes d'utilisateurs](#) dans le Guide du Amazon Cognito développeur.
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'[IsAuthorizedWithToken](#) API. 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 Amazon Cognito jetons sont validés, consultez la section [Autorisation avec autorisations vérifiées par Amazon](#) dans le manuel du Amazon Cognito développeur.
 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 réponse `Allow OR Deny` 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.

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. Prenez en compte les 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.

Amazon Cognito les groupes 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 est lancée dans le VPC par défaut. Par défaut, APIs dont l'accès au réseau est limité au privé ne VPCs 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 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.

Ajouter 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 Guide du Amazon Cognito développeur.

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 souhaitez 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 un niveau élevé 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 des 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 [Mappage Amazon Cognito des jetons au schéma](#) et [Mappage des jetons OIDC au schéma](#) pour des exemples de politiques qui traitent les réclamations sous forme de jetons d'identification et d'accès.

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 Gateway API. 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'SchemaDefinitionobjet. 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::Policy](#) suivantes : Copiez votre déclaration de politique dans l'Definitionobjet. Escapez " les personnages en tant que\".
4. [AWS::Lambda::Function](#), [AWS::IAM::Role](#), [IAM::Policy](#), [AWS::IAM::Authorizer](#), [AWS::ApiGateway::LambdaPermission](#)

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](#)
- [J'ai joint l'autorisateur Lambda à mon API mais il ne génère pas de demandes d'autorisation](#)
- [J'ai reçu une décision d'autorisation inattendue et je souhaite revoir la logique d'autorisation](#)
- [Je souhaite trouver les journaux de mon autorisateur Lambda](#)
- [Mon autorisateur Lambda n'existe pas](#)
- [Mon API se trouve dans un VPC privé et ne peut pas invoquer l'autorisateur](#)
- [Je souhaite traiter des attributs utilisateur supplémentaires dans mon modèle d'autorisation](#)
- [Je souhaite ajouter de nouvelles actions, de nouveaux attributs de contexte d'action ou de nouvelles ressources](#)

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 qu'il mette 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 deAVPAuthorizerLambda. Si vous avez créé plusieurs magasins de politiques liés à l'API, utilisez l'heure de dernière modification de vos fonctions pour les corrélérer 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 API Gateway console, 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 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 revendications provenant de votre source

d'identité aux politiques, consultez les sections [Mappage Amazon Cognito des jetons sur le schéma](#) et [Mappage des jetons OIDC sur le schéma](#).

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 politiques 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.

Supprimer des magasins de politiques

Vous pouvez supprimer les boutiques Amazon Verified Permissions Policy à l'aide du AWS Management Console ou du AWS CLI. La suppression d'un magasin de politiques supprime définitivement le schéma ainsi que les politiques et modèles de politiques du magasin de politiques. Tous les alias de magasin de politiques associés au magasin de politiques supprimé seront également supprimés.

La protection contre la suppression empêche la suppression accidentelle d'un magasin de politiques. La protection contre la suppression est activée sur tous les nouveaux magasins de politiques créés via le AWS Management Console. En revanche, il est désactivé pour tous les magasins de politiques créés via un appel d'API ou de SDK.

Vous souhaitez peut-être supprimer des magasins de politiques pour les raisons suivantes :

- Vous avez atteint le quota de boutiques politiques disponibles dans une région donnée. Pour de plus amples informations, veuillez consulter [Quotas de ressources](#).

- Vous ne prenez plus en charge un locataire dans une application multi-locataires et, par conséquent, vous n'avez plus besoin de ce magasin de politiques.

AWS Management Console

Pour supprimer un entrepôt de politiques

1. Ouvrez la [console des autorisations vérifiées](#). Choisissez votre magasin de politiques.
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.

Note

Si la protection contre la suppression est activée, vous devez la désactiver avant de pouvoir choisir Supprimer. Pour le désactiver, sélectionnez Désactiver la protection contre la suppression.

AWS CLI

Pour supprimer un entrepôt de politiques

Vous pouvez supprimer un magasin de politiques en utilisant cette `delete-policy-store` opération, en le *PSEXAMPLEabcdefg111111* remplaçant par l'ID du magasin de politiques que vous souhaitez supprimer.

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

Cette commande ne produit aucune sortie si elle réussit.

Note

Si la protection contre la suppression est activée pour ce magasin de règles, vous devez d'abord exécuter l'`update-policy-store` opération et désactiver la protection contre la suppression.

```
aws verifiedpermissions update-policy-store \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

```
--deletion-protection "DISABLED" \  
--policy-store-id PSEXAMPLEabcdefgh111111
```

Alias de la boutique Amazon Verified Permissions Policy

Un alias de magasin de politiques est un nom convivial pour un magasin de politiques. Par exemple, les alias de magasin de politiques vous permettent de faire référence à un magasin de politiques en utilisant `policy-store-alias/example-policy-store` au lieu de `PSEXAMPLEabcdefg111111`. Les alias du Policy Store peuvent être utilisés dans toute opération Verified Permissions qui accepte un paramètre `policyStoreId` d'entrée.

Vous pouvez créer un alias de magasin de politiques pour un magasin de politiques à l'aide de l'`CreatePolicyStoreAliasAPI` ou de la `AWS::VerifiedPermissions::PolicyStoreAlias` CloudFormation ressource.

L'API Amazon Verified Permissions fournit un contrôle total des alias des magasins de politiques dans chaque région Compte AWS et dans chaque région. L'API inclut des opérations permettant de créer un alias de magasin de politiques (`CreatePolicyStoreAlias`), d'afficher les noms d'alias de magasin de politiques et les ARN d'alias de magasin de politiques (`GetPolicyStoreAlias`, `ListPolicyStoreAliases`), et de supprimer un alias de magasin de politiques (`DeletePolicyStoreAlias`).

Rubriques

- [Propriétés des alias du Policy Store](#)
- [Création d'alias de boutique Amazon Verified Permissions Policy](#)
- [Récupération des alias de la boutique Amazon Verified Permissions Policy](#)
- [Supprimer les alias de la boutique Amazon Verified Permissions Policy](#)
- [Utilisation de la politique d'autorisation vérifiée d'Amazon pour stocker les alias dans les opérations d'API](#)
- [Contrôle de l'accès aux alias du Policy Store](#)

Propriétés des alias du Policy Store

Comment fonctionnent les alias de la boutique Policy dans Amazon Verified Permissions.

Un alias de magasin de politiques est un alias indépendant AWS ressource

Un alias de magasin de politiques n'est pas une propriété d'un magasin de politiques. Les actions que vous effectuez sur l'alias du magasin de politiques n'affectent pas le magasin de politiques

associé. Vous pouvez supprimer l'alias du magasin de politiques sans aucun effet sur le magasin de politiques associé. Si vous supprimez un magasin de politiques, tous les alias de magasin de politiques associés à ce magasin de politiques sont également supprimés.

Chaque alias de magasin de politiques possède un Amazon Resource Name (ARN) qui identifie de manière unique l'alias de magasin de politiques. Si vous spécifiez un alias de magasin de politiques comme ressource dans une stratégie IAM, la politique fait référence à l'alias du magasin de politiques, et non au magasin de politiques associé.

Chaque alias de magasin de politiques possède deux formats

Lorsque vous créez un alias de magasin de politiques, vous spécifiez le nom de l'alias du magasin de politiques. Amazon Verified Permissions crée pour vous l'alias ARN de la boutique de politiques.

- Un ARN d'alias de magasin de politiques est un nom de ressource Amazon (ARN) qui identifie de manière unique l'alias du magasin de politiques.

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

- Un nom d'alias de magasin de politiques unique dans la région Compte AWS et. Dans l'API Amazon Verified Permissions, le nom d'alias du Policy Store est toujours préfixé par `policy-store-alias/`.

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

Les alias du Policy Store ne sont pas secrets

Les alias du Policy Store peuvent être affichés en texte clair dans les CloudTrail journaux et autres sorties. N'incluez pas d'informations confidentielles ou sensibles dans le nom d'alias du Policy Store.

Chaque alias de magasin de politiques est associé à un magasin de politiques à la fois

L'alias du magasin de politiques et le magasin de politiques associé doivent appartenir à Compte AWS la même région. Vous pouvez associer un alias de magasin de politiques à n'importe quel magasin de politiques de Compte AWS la même région.

Par exemple, cette `ListPolicyStoreAliases` sortie indique que l'alias du magasin de `example-policy-store` politiques est associé à exactement un magasin de stratégies cible, représenté par la `policyStoreId` propriété.

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

Plusieurs alias peuvent être associés au même magasin de politiques

Par exemple, vous pouvez associer les `example-policy-store-2` alias `example-policy-store` et au même magasin de politiques.

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

Un alias de magasin de politiques doit être unique dans un Compte AWS et région

Par exemple, vous ne pouvez avoir qu'un seul alias de magasin de politiques dont le nom se trouve `example-policy-store` dans chaque région Compte AWS et dans chaque région. Les alias du

Policy Store font la distinction majuscules/majuscules. Vous ne pouvez pas modifier le nom d'alias d'un magasin de politiques. Toutefois, vous pouvez supprimer l'alias du magasin de politiques et créer un nouvel alias du magasin de politiques portant le nom souhaité après l'expiration de la période de réservation de 24 heures.

Vous pouvez créer des alias de magasin de politiques portant le même nom dans différentes régions. Chaque alias de magasin de politiques aura un ARN unique. Si votre code fait référence à un alias de magasin de politiques `policy-store-alias/example-policy-store`, par exemple, vous pouvez l'exécuter dans plusieurs régions. Dans chaque région, il utilise un magasin de politiques différent.

Les alias du Policy Store prennent en charge les suppressions logicielles et les suppressions matérielles

Par défaut, lorsqu'un alias de magasin de politiques est supprimé, il est automatiquement supprimé. Le nom d'alias du magasin de politiques est réservé pour une période de 24 heures. Si vous tentez de créer un alias de magasin de politiques portant le même nom pendant cette période, la demande sera rejetée. Pendant cette période, `GetPolicyStoreAlias` renvoie l'alias du magasin de politiques avec l'`PendingDeletion` état. Vous pouvez également supprimer définitivement un alias de magasin de politiques en le spécifiant `HardDelete` comme `deletionMode`. Un alias de magasin de politiques supprimé définitivement est immédiatement supprimé et le nom d'alias du magasin de politiques peut être immédiatement réutilisé. Pour de plus amples informations, veuillez consulter [Supprimer les alias de la boutique Amazon Verified Permissions Policy](#).

Vous pouvez utiliser des alias pour identifier les magasins de politiques

Vous pouvez utiliser un alias de magasin de politiques pour identifier un magasin de politiques dans toutes les opérations qui acceptent un `policyStoreId` (par exemple, `IsAuthorized`). Dans de tels cas, le nom d'alias du magasin de politiques doit être préfixé par `policy-store-alias/`. Les alias du magasin de politiques ne peuvent pas être utilisés pour identifier un magasin de politiques pour l'`DeletePolicyStore` opération.

Vous ne pouvez pas utiliser un nom d'alias de magasin de politiques ou un ARN d'alias de magasin de politiques pour identifier un magasin de politiques dans l'`Resource` élément d'une stratégie IAM. Pour contrôler l'accès à un magasin de politiques lorsqu'il est référencé via un alias de magasin de politiques, consultez [Contrôle de l'accès aux alias du Policy Store](#).

Création d'alias de boutique Amazon Verified Permissions Policy

Vous pouvez créer un alias de magasin de politiques pour référencer un magasin de politiques à l'aide d'un nom convivial. Le nom d'un alias de magasin de politiques doit être unique par Compte AWS région. Les alias de magasin de politiques ne peuvent être associés qu'à des magasins de politiques appartenant à la même région Compte AWS et actifs dans la même région que l'alias du magasin de politiques. Les alias du Policy Store sont des ressources distinctes dotées de leurs propres autorisations ARNs et de celles de l'IAM.

Par défaut, seuls 10 alias de magasin de politiques peuvent être associés au même magasin de politiques.

Note

`CreatePolicyStoreAlias` est idempotent. Si vous appelez l'opération `CreatePolicyStoreAlias` avec un nom d'alias de magasin de politiques et un ID de magasin de politiques correspondant à un alias de magasin de politiques existant, l'opération réussit et renvoie l'alias de magasin de politiques existant. Toutefois, si vous appelez l'opération `CreatePolicyStoreAlias` avec un nom d'alias de magasin de politiques existant mais avec un identifiant de magasin de politiques différent, l'opération renvoie un `ConflictException`.

AWS CLI

Pour créer un alias de magasin de politiques

Vous pouvez créer un alias de magasin de politiques à l'aide de cette [CreatePolicyStoreAlias](#) opération. L'exemple suivant crée un alias de magasin de politiques portant le nom `example-policy-store`.

```
$ aws verifiedpermissions create-policy-store-alias \
  --alias-name policy-store-alias/example-policy-store \
  --policy-store-id PSEXAMPLEEabcdefg111111
{
  "aliasName": "policy-store-alias/example-policy-store",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store",
  "createdAt": "2024-01-15T12:30:00.000000+00:00"
```

```
}
```

Récupération des alias de la boutique Amazon Verified Permissions Policy

Vous pouvez récupérer des informations sur les alias de magasin de politiques à l'aide de l'GetPolicyStoreAlias opération visant à obtenir des détails sur un alias de magasin de politiques spécifique ou de l>ListPolicyStoreAliases opération visant à répertorier tous les alias de magasin de politiques de votre région Compte AWS et de votre région.

Obtenir un alias de Policy Store

Utilisez cette GetPolicyStoreAlias opération pour récupérer des informations sur un alias de magasin de politiques spécifique, y compris l'ID de magasin de politiques associé.

AWS CLI

Pour récupérer les informations relatives à un alias de magasin de politiques

Vous pouvez récupérer un alias de magasin de politiques à l'aide de cette [GetPolicyStoreAlias](#) opération. L'exemple suivant récupère les détails d'un alias de magasin de politiques portant le nom `example-policy-store`.

```
$ aws verifiedpermissions get-policy-store-alias \  
  --alias-name policy-store-alias/example-policy-store  
{  
  "aliasName": "policy-store-alias/example-policy-store",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",  
  "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-  
alias/example-policy-store",  
  "createdAt": "2024-01-15T12:30:00.000000+00:00",  
  "state": "Active"  
}
```

Politique de référencement : alias de magasin

Utilisez cette ListPolicyStoreAliases opération pour répertorier tous les alias du Policy Store de votre région Compte AWS et de votre région. Vous pouvez utiliser le `filter` paramètre pour

répertorier uniquement les alias de magasin de politiques associés à un magasin de politiques spécifique.

AWS CLI

Pour répertorier tous les alias du Policy Store

Vous pouvez répertorier les alias du policy store à l'aide de cette [ListPolicyStoreAliases](#) opération. L'exemple suivant répertorie tous les alias de magasin de politiques détenus par le 123456789012 Compte AWS dans la région us-west-2.

```
$ aws verifiedpermissions list-policy-store-aliases
{
  "policyStoreAliases": [
    {
      "aliasName": "policy-store-alias/example-policy-store",
      "policyStoreId": "PSEXAMPLEEabcdefg111111",
      "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store",
      "createdAt": "2024-01-15T12:30:00.000000+00:00",
      "state": "Active"
    },
    {
      "aliasName": "policy-store-alias/example-policy-store-2",
      "policyStoreId": "PSEXAMPLEEabcdefg111111",
      "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store-2",
      "createdAt": "2024-01-16T09:15:00.000000+00:00",
      "state": "Active"
    },
    {
      "aliasName": "policy-store-alias/example-policy-store-3",
      "policyStoreId": "PSEXAMPLEEabcdefg222222",
      "aliasArn": "arn:aws:verifiedpermissions:us-west-2:123456789012:policy-store-alias/example-policy-store-3",
      "createdAt": "2024-01-17T14:45:00.000000+00:00",
      "state": "Active"
    }
  ]
}
```

Pour répertorier les alias d'un magasin de politiques spécifique

Utilisez le `filter` paramètre pour répertorier uniquement les alias associés à un magasin de règles spécifique.

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

Supprimer les alias de la boutique Amazon Verified Permissions Policy

Vous pouvez supprimer un alias de magasin de politiques lorsqu'il n'est plus nécessaire. La suppression d'un alias de magasin de politiques n'affecte pas le magasin de politiques associé. La suppression d'un magasin de politiques supprime tous les alias de magasin de politiques associés à ce magasin de politiques.

Amazon Verified Permissions prend en charge deux modes de suppression pour les alias de la boutique Policy :

- Suppression logicielle (par défaut) : l'alias du magasin de politiques entre dans un `PendingDeletion` état. Le nom d'alias du magasin de politiques est réservé pendant 24 heures et ne peut pas être réutilisé pendant cette période. Pendant cette période,

`GetPolicyStoreAlias` renvoie l'alias du magasin de politiques avec l'`PendingDeletion` état. Il s'agit du comportement par défaut lorsque vous ne spécifiez pas `deletionMode`, ou lorsque vous spécifiez `SoftDelete`.

- Suppression définitive : l'alias du magasin de politiques est immédiatement supprimé. Le nom d'alias du magasin de politiques peut être réutilisé immédiatement. Pour effectuer une suppression définitive, spécifiez `HardDelete` comme `deletionMode`.

Suppression progressive d'un alias de magasin de politiques

Par défaut, la suppression d'un alias de magasin de politiques entraîne une suppression progressive. L'alias du magasin de politiques entre dans l'`PendingDeletion` état et le nom de l'alias du magasin de politiques est réservé pendant 24 heures.

AWS CLI

Pour supprimer progressivement un alias de magasin de politiques

Vous pouvez supprimer en douceur un alias de magasin de politiques à l'aide de [DeletePolicyStoreAlias](#) cette opération. L'exemple suivant supprime progressivement un alias de magasin de politiques portant le nom `example-policy-store`.

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

Vous pouvez également spécifier de manière explicite le mode de suppression logicielle.

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

Suppression définitive d'un alias de magasin de politiques

Pour supprimer immédiatement un alias de magasin de politiques, spécifiez-le `HardDelete` sous la forme `deletionMode`. Un alias de magasin de politiques supprimé définitivement n'entre pas dans l'`PendingDeletion` état et le nom de l'alias du magasin de politiques peut être immédiatement réutilisé.

Si vous supprimez définitivement un alias de banque de politiques précédemment supprimé automatiquement, l'alias de banque de politiques est immédiatement supprimé.

Important

Les autorisations vérifiées par Amazon sont finalement cohérentes. Si vous supprimez définitivement un alias de magasin de politiques et que vous le recréez immédiatement pour qu'il pointe vers un autre magasin de politiques, les demandes faisant référence à l'alias du magasin de politiques peuvent continuer à être résolues vers le magasin de politiques précédemment associé pendant une courte période. Pour éviter des résultats d'autorisation inattendus, laissez le temps à la suppression de se propager avant de recréer un alias de magasin de politiques portant le même nom.

AWS CLI

Pour supprimer définitivement un alias de magasin de politiques

Vous pouvez supprimer définitivement un alias de magasin de politiques en utilisant l'[DeletePolicyStoreAlias](#) opération dont le `deletion-mode` paramètre est défini sur `HardDelete`. L'exemple suivant supprime immédiatement un alias de magasin de politiques portant ce nom `example-policy-store`.

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

Utilisation de la politique d'autorisation vérifiée d'Amazon pour stocker les alias dans les opérations d'API

Toute opération Amazon Verified Permissions qui accepte un `policyStoreId` paramètre, tel que [IsAuthorized](#), et [IsAuthorizedWithTokenGetPolicyStore](#), peut accepter un nom d'alias de magasin de politiques à la place de l'ID de magasin de politiques.

⚠ Important

Lorsque vous utilisez un alias de magasin de politiques comme valeur d'un `policyStoreId` paramètre, vous devez inclure le `policy-store-alias/` préfixe. Par exemple, utilisez `policy-store-alias/example-policy-store`, non `example-policy-store`.

Utilisation des alias du Policy Store dans Operations

La `IsAuthorized` commande suivante utilise un alias de magasin de politiques dont le nom permet `example-policy-store` d'identifier un magasin de politiques.

AWS CLI

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

📘 Note

Vous ne pouvez pas utiliser un alias de magasin de politiques à la place du `policyStoreId` champ pour l'[DeletePolicyStore](#) opération.

Utilisation des alias du Policy Store Across Régions AWS

L'une des utilisations les plus performantes des alias est au niveau des applications qui s'exécutent dans plusieurs Régions AWS. Par exemple, vous pouvez avoir une application globale qui utilise différents magasins de politiques dans chaque région.

- Dans `us-east-1`, vous souhaitez utiliser. `PSEXAMPLEabcdefg111111`
- Dans `eu-west-1`, vous souhaitez utiliser. `PSEXAMPLEabcdefg222222`

Vous pouvez créer une version différente de votre application dans chaque région ou utiliser un dictionnaire ou une instruction `switch` pour sélectionner le magasin de politiques adapté à chaque

région. Mais il est beaucoup plus facile de créer un alias de magasin de politiques portant le même nom d'alias de magasin de politiques dans chaque région. N'oubliez pas que le nom d'alias du magasin de politiques distingue les majuscules et minuscules.

AWS CLI

```
$ aws --region us-east-1 verifiedpermissions create-policy-store-alias \  
  --alias-name policy-store-alias/my-app \  
  --policy-store-id PSEXAMPLEabcdefg111111  
  
$ aws --region eu-west-1 verifiedpermissions create-policy-store-alias \  
  --alias-name policy-store-alias/my-app \  
  --policy-store-id PSEXAMPLEabcdefg222222
```

Utilisez ensuite l'alias Policy Store dans votre code. Lorsque votre code s'exécute dans chaque région, l'alias du magasin de politiques fait référence au magasin de politiques associé dans cette région.

AWS CLI

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

Cependant, il existe un risque que l'alias du magasin de politiques soit supprimé. Dans ce cas, les tentatives de l'application pour utiliser le nom d'alias du magasin de politiques échoueront et vous devrez peut-être recréer ou mettre à jour l'alias du magasin de politiques. Pour atténuer ce risque, veillez à ne pas autoriser les principaux à gérer les alias du Policy Store que vous utilisez dans votre application.

Contrôle de l'accès aux alias du Policy Store

Les responsables qui gèrent les alias du magasin de politiques doivent être autorisés à interagir avec ces alias du magasin de politiques et, pour certaines opérations, avec le magasin de politiques auquel l'alias du magasin de politiques est associé. Vous pouvez fournir ces autorisations à l'aide IAM de politiques.

Les sections suivantes décrivent les autorisations requises pour créer et gérer des alias de magasin de politiques.

autorisations vérifiées : CreatePolicyStoreAlias

Pour créer un alias de magasin de politiques, le principal doit disposer des autorisations suivantes à la fois pour l'alias du magasin de politiques et pour le magasin de politiques associé.

- `verifiedpermissions:CreatePolicyStoreAlias` pour l'alias du Policy Store. Fournissez cette autorisation dans une IAM politique attachée au principal autorisé à créer l'alias du magasin de politiques.

L'exemple de déclaration de politique suivant spécifie un alias de magasin de politiques particulier dans un Resource élément. Mais vous pouvez répertorier plusieurs alias de magasin de politiques ARNs ou spécifier un modèle d'alias de magasin de politiques, tel que `"sample*"`. Vous pouvez également spécifier une Resource valeur de `"*"` pour permettre au principal de créer n'importe quel alias de magasin de politiques dans la région Compte AWS et.

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

- `verifiedpermissions:CreatePolicyStoreAlias` pour le magasin de politiques associé. Cette autorisation doit être fournie dans une IAM politique.

```
{
  "Sid": "PolicyStorePermissionForAlias",
  "Effect": "Allow",
  "Action": "verifiedpermissions:CreatePolicyStoreAlias",
  "Resource": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
}
```

autorisations vérifiées : GetPolicyStoreAlias

Pour obtenir des informations sur un alias de magasin de politiques spécifique, le principal doit être `verifiedpermissions:GetPolicyStoreAlias` autorisé à utiliser l'alias de magasin de politiques dans une IAM politique.

L'exemple de déclaration de politique suivant donne au principal l'autorisation d'obtenir un alias de magasin de politiques spécifique.

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

autorisations vérifiées : ListPolicyStoreAliases

Pour répertorier les alias du magasin de politiques dans la région Compte AWS et, le principal doit disposer d'une `verifiedpermissions:ListPolicyStoreAliases` autorisation dans une IAM politique. Étant donné que cette politique n'est liée à aucune ressource de magasin de politiques ou d'alias de magasin de politiques en particulier, la valeur de l'élément de ressource dans la stratégie doit être `"*"`.

Par exemple, la déclaration de IAM politique suivante donne au principal l'autorisation de répertorier tous les alias du magasin de politiques dans le Compte AWS.

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

autorisations vérifiées : DeletePolicyStoreAlias

Pour supprimer un alias de magasin de politiques, le principal a besoin d'une autorisation uniquement pour l'alias du magasin de politiques.

Note

La suppression d'un alias de magasin de politiques n'a aucun effet sur le magasin de politiques associé, même si les applications qui font référence à l'alias du magasin de politiques recevront des erreurs. Si vous supprimez par erreur un alias de magasin de politiques, vous pouvez le recréer après la période de réservation de 24 heures.

Le principal a besoin `verifiedpermissions:DeletePolicyStoreAlias` d'une autorisation pour l'alias du magasin de politiques. Fournissez cette autorisation dans une IAM politique attachée au principal autorisé à supprimer l'alias du magasin de politiques.

L'exemple de déclaration de politique suivant spécifie l'alias du magasin de politiques dans un `Resource` élément. Mais vous pouvez répertorier plusieurs alias de magasin de politiques ARNs ou spécifier un modèle d'alias de magasin de politiques, tel que `"sample*"`. Vous pouvez également spécifier une `Resource` valeur de `"*"` pour permettre au principal de supprimer tout alias de magasin de politiques dans la région Compte AWS et.

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

Limitation des autorisations relatives aux alias du magasin Policy

Vous pouvez utiliser un alias de magasin de politiques pour faire référence à un magasin de politiques dans toute opération qui accepte un `policyStoreId` champ en entrée. Lorsque vous le faites, Amazon Verified Permissions autorise par `verifiedpermissions:GetPolicyStoreAlias` rapport à l'alias du magasin de politiques et l'opération demandée par rapport au magasin de politiques associé.

Par exemple, si l'`IsAuthorized` opération est effectuée à l'aide d'un alias de magasin de politiques, le principal a besoin des deux éléments suivants :

- `verifiedpermissions:GetPolicyStoreAlias` autorisation pour l'alias du Policy Store
- `verifiedpermissions:IsAuthorized` autorisation pour le magasin de politiques associé

L'exemple de politique suivant accorde l'autorisation d'appeler `IsAuthorized` en utilisant un alias de magasin de politiques spécifique.

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

Pour limiter les alias de magasin de politiques qu'un principal peut utiliser, limitez l'`verifiedpermissions:GetPolicyStoreAlias` autorisation. Par exemple, la politique suivante autorise le principal à utiliser n'importe quel alias de magasin de politiques, à l'exception de ceux commençant par `Restricted`.

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

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 voir la différence entre la façon dont Verified Permissions et Cedar gèrent les schémas, consultez [Support de schéma](#).

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 les autorisations vérifiées 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](#). Choisissez votre magasin de polices.
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 préfacier 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 PSEXAMPLEabcdefgh111111  
{  
  "policyStoreId": "PSEXAMPLEabcdefgh111111",  
  "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'PutSchemaAPI. Pour plus d'informations, consultez le guide [PutSchema](#) de référence de l'API Amazon Verified Permissions.

Modification des schémas du magasin de politiques

Lorsque vous sélectionnez Schema dans la console Amazon Verified Permissions, les types d'entités et les actions qui constituent votre schéma sont affichés. Vous pouvez afficher et modifier votre schéma en mode visuel ou en mode JSON. Le mode visuel vous permet de mettre à jour le schéma en ajoutant de nouveaux types et actions à l'aide de différents assistants. En mode JSON, vous pouvez commencer à mettre à jour le code JSON du schéma directement dans l'éditeur JSON.

Visual Mode

L'éditeur de schéma visuel commence par une série de diagrammes illustrant les relations entre les entités de votre schéma. Choisissez Expand pour optimiser l'affichage des diagrammes. Deux diagrammes sont disponibles :

- Diagramme d'actions : la vue du diagramme d'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 sont autorisés à 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 une entité pour afficher les détails.

Pour modifier un schéma d'autorisations vérifiées en mode visuel

1. Ouvrez la [console des autorisations vérifiées](#). Choisissez votre magasin de polices.
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.

JSON mode

Lorsque vous effectuez des mises à jour, vous remarquerez que l'éditeur JSON valide votre code par rapport à la syntaxe JSON et qu'il identifie les erreurs et les avertissements lors de vos modifications, ce qui vous permet de détecter rapidement les problèmes. De plus, vous n'avez pas à vous soucier du formatage du JSON, il vous suffit de choisir Format JSON une fois vos mises à jour et le format sera mis à jour pour correspondre au formatage JSON attendu.

Pour modifier un schéma d'autorisations vérifiées en mode JSON

1. Ouvrez la [console des autorisations vérifiées](#). Choisissez votre magasin de polices.
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.

Activation du mode de validation de la politique d'autorisations Amazon Verified Permissions

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 tentative de demande si la validation échoue. Pour cette raison, nous vous recommandons de désactiver la validation pendant le développement de votre application, de l'activer à des fins de test et de la laisser activée pendant que votre application est en production.

AWS Management Console

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

1. Ouvrez la [console des autorisations vérifiées](#). Choisissez votre magasin de polices.
2. Sélectionnez 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'[UpdatePolicyStore](#) opération et en spécifiant une valeur différente pour le [ValidationSettings](#) paramètre.

```
$ 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 des autres politiques. 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.

Vous pouvez éventuellement attribuer un nom de stratégie à une stratégie. Les noms des politiques doivent être uniques pour toutes les politiques du magasin de politiques et être préfixés par `name/`. Vous pouvez utiliser un nom de stratégie à la place de l'ID de stratégie dans les opérations du plan de contrôle qui acceptent un `policyId` paramètre. L'exemple suivant utilise un nom de stratégie pour récupérer une politique avec `GetPolicy`.

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

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 identifiants uniques universels (UUID) pour tous les identifiants 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 == Rôle : « A1B2C3D4-E5F6-A1B2-C3D4-Example11111",//administrateurs

- 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.

Rubriques

- [Création de politiques statiques relatives aux autorisations vérifiées par Amazon](#)
- [Modification des politiques statiques d'Amazon Verified Permissions](#)
- [Ajouter du contexte](#)
- [Utilisation du banc de test Amazon Verified Permissions](#)
- [Exemples de politiques relatives aux autorisations vérifiées par Amazon](#)


Création de politiques statiques relatives aux autorisations vérifiées par Amazon

Vous pouvez créer une politique statique pour les principaux afin de leur permettre ou de leur interdire d'effectuer des actions spécifiques sur des ressources spécifiques pour votre application. Une politique statique inclut des valeurs spécifiques pour `principal` et `resource` et est prête à être utilisée dans les décisions d'autorisation.

AWS Management Console

Pour créer une politique statique

1. Ouvrez la [console des autorisations vérifiées](#). Choisissez votre magasin de polices.
2. Dans le panneau de navigation de gauche, sélectionnez Politiques (Politiques).
3. Choisissez Créer une politique, puis sélectionnez Créer une politique statique.

 Note

Si vous souhaitez utiliser une déclaration de politique, passez à l'étape 8 et collez-la dans la section Politique de la page suivante.

4. Dans la section Effet de la politique, choisissez si la politique autorisera ou interdira lorsqu'une demande correspond à la politique. Si vous choisissez Autoriser, la politique autorise les principaux à effectuer les actions sur les ressources. À l'inverse, si vous choisissez Interdire, la politique n'autorise pas les principaux à effectuer les actions sur les ressources.
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 Create Policy (Créer une politique).

AWS CLI

Pour créer une politique statique

Vous pouvez créer une politique statique à l'aide de l'[CreatePolicy](#) opération. 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\"  
}
```

Pour créer une stratégie avec un nom de stratégie

Vous pouvez éventuellement spécifier un nom de stratégie lors de la création d'une stratégie. Le nom doit être unique pour toutes les politiques du magasin de politiques et être préfixé parname/. Vous pouvez utiliser le nom à la place de l'ID de politique.

```
$ aws verifiedpermissions create-policy \  
  --strategy-name my-test-policy
```

```
--definition "{ \"static\": { \"Statement\": \"permit(principal, action, resource in Album:\\\\\"public_folder\\\\\")\"} }" \
--policy-store-id PSEXAMPLEabcdefg111111 \
--name name/example-policy
{
  "createdDate": "2023-06-12T20:33:37.382907+00:00",
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
  "policyId": "SPEXAMPLEabcdefg111111",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyType": "STATIC",
  "resource": {
    "entityId": "public_folder",
    "entityType": "Album"
  }
}
```

Note

Si vous spécifiez un nom déjà associé à une autre politique dans le magasin de politiques, vous recevez un `ConflictException` message d'erreur.

Modification des politiques statiques d'Amazon Verified Permissions

Vous pouvez modifier une politique statique existante dans votre magasin de politiques. Vous pouvez uniquement mettre à jour les politiques statiques directement. Pour modifier une politique liée à un modèle, vous devez mettre à jour le modèle de stratégie. Pour de plus amples informations, veuillez consulter [Modification des modèles de politique d'autorisations Amazon Verified](#).

Vous pouvez modifier les éléments suivants 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. Pour modifier l'un de ces éléments, vous devez supprimer et recréer la politique.

- Une politique allant d'une politique statique à une politique liée à un modèle.

- L'effet d'une politique statique émise par `permit ouforbid`.
- Le principal référencé par une politique statique.
- Le ressource référencé par une politique statique.

AWS Management Console

Pour modifier une politique statique

1. Ouvrez la [console des autorisations vérifiées](#). Choisissez votre magasin de polices.
2. Dans le panneau de navigation de gauche, sélectionnez Politiques (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` ou `condition` de votre politique statique. Vous ne pouvez pas mettre à jour l'effet de la politique `principal`, ou ressource 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'[UpdatePolicy](#) opération. 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 PSEXAMPLEabcdefgh111111

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

```

Pour mettre à jour le nom d'une politique

Vous pouvez définir ou mettre à jour le nom d'une politique lors de la mise à jour d'une politique. Le nom doit être unique pour toutes les politiques du magasin de politiques et être préfixé parname/. Si vous n'incluez pas le champ du nom dans la demande de mise à jour, le nom existant reste inchangé. Pour supprimer un nom, définissez-le sur une chaîne vide.

```

$ aws verifiedpermissions update-policy \
  --policy-id SPEXAMPLEabcdefgh111111 \
  --policy-store-id PSEXAMPLEabcdefgh111111 \
  --definition file://definition.txt \
  --name name/example-policy

{
  "createdDate": "2023-06-12T20:33:37.382907+00:00",
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",

```

```
"policyId": "SPEXAMPLEabcdefg111111",
"policyStoreId": "PSEXAMPLEabcdefg111111",
"policyType": "STATIC",
"principal": {
  "entityId": "janeFriends",
  "entityType": "UserGroup"
},
"resource": {
  "entityId": "vacationFolder",
  "entityType": "Album"
}
}
```

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. Les demandes de jetons d'accès sont contextuelles. Vous souhaitez 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 d'autorisations vérifiées doivent se JSON-formatted trouver dans un contextMap élément.

Les exemples illustrant ce contenu proviennent d'un [exemple de magasin de politiques](#). Pour suivre, créez le magasin DigitalPetStore d'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 DigitalPetStore de politiques.

```
"context": {
  "contextMap": {
    "AccountCodes": {
      "set": [
        {
          "long": 111122223333
        },
        {
          "long": 444455556666
        }
      ],
    }
  }
}
```

```
{
  "long": 123456789012
}
],
"approvedBy": {
  "entityIdentifier": {
    "entityId": "Bob",
    "entityType": "DigitalPetStore::User"
  }
},
"MfaAuthorized": {
  "boolean": true
},
"NetworkInfo": {
  "record": {
    "IPAddress": {
      "string": "192.0.2.178"
    },
    "Country": {
      "string": "United States of America"
    },
    "SSL": {
      "boolean": true
    }
  }
},
"RequestedOrderCount": {
  "long": 4
},
"UserAgent": {
  "string": "My UserAgent 1.12"
}
}
```

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.

String

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 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'`employéBob`.

Pour tester cet exemple de contexte dans l'exemple d'`DigitalPetStore`application, 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, `DigitalPetStore` 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. Pour des exemples de contexte avec les informations des jetons d'accès comme contexte, consultez

les sections [Cartographie des jetons Amazon Cognito d'accès](#) et [Cartographie des jetons d'accès OIDC](#).

Schema

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

```
"GetOrder": {
  "memberOf": [],
  "appliesTo": {
    "resourceTypes": [
      "Order"
    ],
    "context": {
      "type": "Record",
      "attributes": {
        "AccountCodes": {
          "type": "Set",
          "required": true,
          "element": {
            "type": "Long"
          }
        },
        "approvedBy": {
          "name": "User",
          "required": true,
          "type": "Entity"
        },
        "MfaAuthorized": {
          "type": "Boolean",
          "required": true
        },
        "NetworkInfo": {
          "type": "NetworkInfo",
          "required": true
        },
        "RequestedOrderCount": {
          "type": "Long",
          "required": true
        },
        "UserAgent": {
          "required": true,
```

```

        "type": "String"
      }
    }
  },
  "principalTypes": [
    "User"
  ]
}
}

```

Pour référencer le type de record données nommé `NetworkInfo` dans le contexte de votre demande, créez une construction [CommonType](#) dans votre schéma en ajoutant ce qui suit à votre schéma auparavant `actions`. Une `commonType` construction est un ensemble partagé d'attributs que vous pouvez appliquer à différentes entités.

```

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

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 chaîneMy UserAgent.
3. L'utilisateur doit avoir demandé à consulter 4 commandes ou moins.
4. L'un des codes de compte de l'utilisateur doit être111122223333.
5. L'adresse IP de l'utilisateur doit provenir des États-Unis d'Amérique, sa session doit être cryptée et son adresse IP doit commencer par. 192.0.2.
6. Un employé doit avoir approuvé sa commande. Dans l'entitiesélément de la demande d'autorisation, nous déclarerons un utilisateur Bob ayant le rôle deEmployee.

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 deBob, un utilisateur avec un rôle deEmployee.

```
{  
  "principal": {  
    "entityType": "DigitalPetStore::User",  
    "entityId": "Alice"  }}
```

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

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

```
    }
  }
},
"parents": []
}
]
},
"policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Utilisation du banc de test Amazon Verified Permissions

Utilisez le banc de test des autorisations vérifiées pour tester et 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 passer du mode visuel au 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 une présentation visuelle et une démonstration du banc de test, consultez [Amazon Verified Permissions - Policy Creation and Testing \(Primer Series #3\)](#) sur la AWS YouTube chaîne.

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](#). 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](#). 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.

Exemples de politiques relatives aux autorisations vérifiées par Amazon

Certains des exemples de politiques inclus ici sont des exemples de politiques de base de Cedar, tandis que d'autres sont vérifiés Permissions-specific. Les principaux renvoient au guide de référence du langage politique de Cedar et y sont inclus. 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](#)
- [Utilise la notation entre crochets pour référencer les attributs des jetons](#)
- [Utilise la notation par points pour référencer les attributs](#)
- [Reflète Amazon Cognito Attributs des jetons d'identification](#)
- [Reflète les attributs du jeton d'identification OIDC](#)
- [Reflète Amazon Cognito attributs du jeton d'accès](#)
- [Reflète les attributs du jeton d'accès OIDC](#)

Utilise la notation entre crochets pour référencer les attributs des jetons

L'exemple suivant montre comment créer une politique qui utilise la notation entre crochets pour référencer les attributs des jetons.

Pour plus d'informations sur l'utilisation des attributs de jeton dans les politiques des autorisations vérifiées, consultez les sections [Mappage Amazon Cognito des jetons sur le schéma](#) et [Mappage des jetons OIDC sur le 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.*"
};
```

Utilise la notation par points pour référencer les attributs

L'exemple suivant montre comment créer une politique qui utilise la notation par points pour référencer les attributs.

Pour plus d'informations sur l'utilisation des attributs de jeton dans les politiques des autorisations vérifiées, consultez les sections [Mappage Amazon Cognito des jetons sur le schéma](#) et [Mappage des jetons OIDC sur le schéma](#).

```
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"
};
```

Reflète Amazon Cognito Attributs des jetons d'identification

L'exemple suivant montre comment créer une politique qui fait référence aux attributs des jetons d'identification à partir de Amazon Cognito.

Pour plus d'informations sur l'utilisation des attributs de jeton dans les politiques des autorisations vérifiées, consultez les sections [Mappage Amazon Cognito des jetons sur le schéma](#) et [Mappage des jetons OIDC sur le schéma](#).

```
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"
};
```

Reflète les attributs du jeton d'identification OIDC

L'exemple suivant montre comment créer une politique faisant référence aux attributs des jetons d'identification d'un fournisseur OIDC.

Pour plus d'informations sur l'utilisation des attributs de jeton dans les politiques des autorisations vérifiées, consultez les sections [Mappage Amazon Cognito des jetons sur le schéma](#) et [Mappage des jetons OIDC sur le schéma](#).

```
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*"
};
```

Reflète Amazon Cognito attributs du jeton d'accès

L'exemple suivant montre comment créer une politique faisant référence aux attributs des jetons d'accès à partir de Amazon Cognito.

Pour plus d'informations sur l'utilisation des attributs de jeton dans les politiques des autorisations vérifiées, consultez les sections [Mappage Amazon Cognito des jetons sur le schéma](#) et [Mappage des jetons OIDC sur le schéma](#).

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

Reflète les attributs du jeton d'accès OIDC

L'exemple suivant montre comment créer une politique faisant référence aux attributs des jetons d'accès d'un fournisseur OIDC.

Pour plus d'informations sur l'utilisation des attributs de jeton dans les politiques des autorisations vérifiées, consultez les sections [Mappage Amazon Cognito des jetons sur le schéma](#) et [Mappage des jetons OIDC sur le schéma](#).

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

Modèles de politiques d'autorisations vérifiées par Amazon et politiques liées à des modèles

Dans Autorisations vérifiées, les modèles de politique sont des politiques comportant des espaces réservés pour `principal`resource, ou les deux. Les modèles de politique seuls ne peuvent pas être utilisés pour traiter les demandes d'autorisation. Pour traiter les demandes d'autorisation, une politique liée à un modèle doit être créée sur la base d'un modèle de politique. Les modèles de politique permettent de définir une politique une seule fois, puis de l'utiliser avec plusieurs principes et ressources. Les mises à jour du modèle de stratégie sont reflétées dans toutes les politiques qui utilisent le modèle. Pour plus d'informations, consultez les [modèles de politiques Cedar](#) dans le Guide de référence du langage politique de Cedar.

Vous pouvez éventuellement attribuer un nom de modèle de stratégie à un modèle de stratégie. Les noms des modèles de politiques doivent être uniques dans le magasin de politiques et être préfixés par `name/`. Vous pouvez utiliser un nom de modèle de stratégie à la place de l'ID du modèle de stratégie dans les opérations du plan de contrôle qui acceptent un `policyTemplateId` paramètre. Uniquement `GetPolicyTemplate` et `ListPolicyTemplates` renvoient le nom dans la sortie. L'exemple suivant utilise le nom d'un modèle de stratégie pour récupérer un modèle de stratégie `GetPolicyTemplate`.

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

Par exemple, le modèle de stratégie suivant fournit `Read``Edit`, et `Comment` des autorisations pour le principal et la ressource qui utilisent le modèle de stratégie.

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

Si vous deviez créer une politique nommée sur la `Editor` base de ce modèle, lorsqu'un principal est désigné comme éditeur pour une ressource spécifique, votre application créerait une politique qui autorise le principal à lire, modifier et commenter la ressource.

Contrairement aux politiques statiques, les politiques liées à un modèle sont dynamiques. Prenons l'exemple précédent, si vous deviez supprimer l'Commentation du modèle de stratégie, toute politique liée à ce modèle ou basée sur celui-ci serait mise à jour en conséquence et les principes spécifiés dans les politiques ne seraient plus en mesure de commenter les ressources correspondantes.

Pour d'autres exemples de politiques liés à un modèle, voir. [Exemples de politiques liées à un modèle d'autorisations Amazon Verified](#)

Création de modèles de politique d'autorisations vérifiées par Amazon

Vous pouvez créer des modèles de politique dans Permissions vérifiées en utilisant le AWS Management Console, le AWS CLI, ou le AWS SDKs. Les modèles de politique permettent de définir une politique une seule fois, puis de l'utiliser avec plusieurs principes et ressources. Une fois que vous avez créé un modèle de politique, vous pouvez créer des politiques liées à un modèle pour utiliser les modèles de politique avec des principes et des ressources spécifiques. Pour de plus amples informations, veuillez consulter [Création de politiques liées au modèle Amazon Verified Permissions](#).

AWS Management Console

Pour créer un modèle de politique

1. Ouvrez la [console des autorisations vérifiées](#). 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 [CreatePolicyTemplate](#) opération. 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"
}
```

Pour créer un modèle de stratégie avec un nom de modèle de stratégie

Vous pouvez éventuellement spécifier un nom de modèle de stratégie lors de la création d'un modèle de stratégie. Le nom doit être unique pour tous les modèles de politiques du magasin de politiques et être préfixé par `name/`. Vous pouvez utiliser le nom à la place de l'ID du modèle de politique.

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

Note

Si vous spécifiez un nom déjà associé à un autre modèle de stratégie dans le magasin de règles, vous recevez un `ConflictException` message d'erreur.

Création de politiques liées au modèle Amazon Verified Permissions

Vous pouvez créer des politiques liées à un modèle, ou des politiques basées sur un modèle de stratégie, en utilisant le AWS Management Console AWS CLI, ou le AWS SDKs 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.

Pour des exemples de politiques liés à un modèle, voir [Exemples de politiques liées à un modèle d'autorisations Amazon Verified](#)

AWS Management Console

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

1. Ouvrez la [console des autorisations vérifiées](#). Choisissez votre magasin de polices.
2. Dans le panneau de navigation de gauche, sélectionnez Politiques (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 == PhotoFlash::Action::"view",  
  resource == PhotoFlash::Photo::"VacationPhoto94.jpg"  
);
```

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

```
{  
  "templateLinked": {  
    "policyTemplateId": "PTEXAMPLEabcdefg111111",  
    "principal": {  
      "entityType": "PhotoFlash::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": "TPEXAMPLEabcdefgh111111",
"policyStoreId": "PSEXAMPLEabcdefgh111111",
"policyType": "TEMPLATELINKED",
"principal": {
  "entityId": "alice",
  "entityType": "PhotoFlash::User"
},
"resource": {
  "entityId": "VacationPhoto94.jpg",
  "entityType": "PhotoFlash::Photo"
}
}
```

Modification des modèles de politique d'autorisations Amazon Verified

Vous pouvez modifier ou mettre à jour les modèles de politique dans Permissions vérifiées en utilisant le AWS Management Console AWS CLI, le ou le AWS SDKs. La modification d'un modèle de politique met automatiquement à jour les politiques liées au modèle ou basées sur celui-ci. Faites donc attention lorsque vous modifiez les modèles de politique et assurez-vous de ne pas introduire accidentellement une modification susceptible de perturber votre application.

Vous pouvez modifier les éléments suivants d'un modèle de politique :

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

Vous ne pouvez pas modifier les éléments suivants d'un modèle de politique. Pour modifier l'un de ces éléments, vous devez supprimer et recréer le modèle de politique.

- L'effet d'un modèle de politique issu de `permit` ou `forbid`
- Le `principal` référencé par un modèle de politique
- Le `resource` référencé par un modèle de politique


```
--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"  
}
```

Pour mettre à jour le nom d'un modèle de politique

Vous pouvez définir ou mettre à jour le nom d'un modèle de stratégie lors de la mise à jour d'un modèle de stratégie. Le nom doit être unique pour tous les modèles de politiques du magasin de politiques et être préfixé parname/. Si vous n'incluez pas le champ du nom dans la demande de mise à jour, le nom existant reste inchangé. Pour supprimer un nom, définissez-le sur une chaîne vide.

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

Exemples de politiques liées à un modèle d'autorisations Amazon Verified

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.

PhotoFlash exemples

L'exemple suivant 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"  
);
```

L'exemple suivant 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"  
);
```

L'exemple suivant 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"  
);
```

L'exemple suivant 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"  
);
```

L'exemple suivant 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"  
);
```

L'exemple suivant 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 exemples

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

L'exemple suivant montre comment vous pouvez 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"  
);
```

L'exemple suivant montre comment créer une politique liée à un modèle qui utilise le modèle de stratégie 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"  
);
```

Sécurisez vos applications avec des sources d'identité et des jetons

Sécurisez rapidement vos applications en créant une source d'identité représentant 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. Vous pouvez ajouter un groupe d'utilisateurs Amazon Cognito ou un IdP OpenID Connect (OIDC) personnalisé comme source d'identité.

Vous pouvez utiliser les fournisseurs d'identité [OpenID Connect \(OIDC\)](#) (IdPs) dotés d'autorisations vérifiées. Votre application peut générer des demandes d'autorisation à l'aide de jetons Web JSON (JWTs) générés par un fournisseur d'identité conforme à l'OIDC. L'identité de l'utilisateur figurant dans le jeton est mappée à l'identifiant principal. Avec les jetons d'identification, Verified Permissions associe les revendications d'attributs aux attributs principaux. Avec les jetons d'accès, ces revendications sont mises en correspondance avec le [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).

Note

Verified Permissions prend des décisions d'autorisation sur la base des informations provenant d'un jeton IdP, mais n'interagit en aucune façon directement avec l'IdP.

Pour découvrir comment créer step-by-step une logique d'autorisation pour Amazon API Gateway REST à APIs l'aide d'un pool Amazon Cognito d'utilisateurs ou d'un fournisseur d'identité OIDC, consultez la section [Autoriser à l'aide des autorisations vérifiées par API Gateway APIs Amazon avec Amazon Cognito ou apportez votre propre fournisseur d'identité](#) sur le blog de AWS sécurité.

Rubriques

- [Choisir le bon fournisseur d'identité](#)
- [Travailler avec des sources Amazon Cognito d'identité](#)
- [Travailler avec les sources d'identité OIDC](#)

Choisir le bon fournisseur d'identité

Bien que les autorisations vérifiées fonctionnent avec de nombreuses IdPs options, tenez compte des points suivants lorsque vous décidez laquelle utiliser dans votre application :

À utiliser Amazon Cognito lorsque :

- Vous créez de nouvelles applications sans infrastructure d'identité existante
- Vous voulez des groupes d'utilisateurs AWS gérés avec des fonctionnalités de sécurité intégrées
- Vous avez besoin de l'intégration d'un fournisseur d'identité sociale
- Vous souhaitez une gestion simplifiée des jetons

Utilisez les fournisseurs OIDC lorsque :

- Vous disposez d'une infrastructure d'identité existante (Auth0, Okta, Azure AD)
- Vous devez maintenir une gestion centralisée des utilisateurs
- Vous avez des exigences de conformité spécifiques IdPs

Travailler avec des sources Amazon Cognito d'identité

Verified Permissions travaille en étroite collaboration avec les groupes d'utilisateurs d'Amazon Cognito. Amazon Cognito JWTs 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 principal que vous souhaitez associer à votre source d'identité, par exemple `MyCorp::User`.
- Le type d'entité de groupe principal que vous souhaitez associer à votre source d'identité, par exemple `MyCorp::UserGroup`.
- Le client IDs 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` Dans ce cas, vous devez faire référence à un utilisateur de ce groupe d'utilisateurs dont l'identifiant est `a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 us-west-2_EXAMPLE | a1b2c3d4-5678-90ab-cdef-EXAMPLE22222`

Les demandes de jetons du pool d'utilisateurs peuvent contenir des attributs, des étendues, des groupes, des clients IDs et des données personnalisées. [Amazon Cognito JWTs](#) ont la capacité d'inclure une variété d'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, dans [Mappage Amazon Cognito des jetons au schéma](#).

Important

Bien que vous puissiez révoquer les Amazon Cognito jetons avant leur expiration, ils JWTs 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 un 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. Pour plus d'informations, voir [Fin des sessions utilisateur par révocation de jetons](#)

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

L'exemple suivant montre comment créer une politique qui fait référence à un principal qui est un utilisateur d'un groupe d'utilisateurs Cognito. Notez que l'identifiant principal prend la forme de "<userpool-id>|<sub>".

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

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 [IsAuthorizedWithToken](#) échouera avec un `ValidationException`.

Pour plus d'informations sur le mappage des réclamations, consultez [Mappage Amazon Cognito des jetons au schéma](#). Pour plus d'informations sur l'autorisation des Amazon Cognito utilisateurs, consultez la section [Autorisation avec autorisations vérifiées par Amazon](#) dans le manuel Amazon Cognito Developer Guide.

Rubriques

- [Création de sources d' Amazon Cognito identité Amazon Verified Permissions](#)
- [Modification des sources d' Amazon Cognito identité Amazon Verified Permissions](#)
- [Mappage Amazon Cognito des jetons au schéma](#)
- [Validation du client et du public pour Amazon Cognito](#)

Création de sources d' Amazon Cognito identité Amazon Verified Permissions

La procédure suivante ajoute une source d'identité à un magasin de politiques existant.

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 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 [create-identity-source](#) dans AWS CLI ou [CreateIdentitySource](#) dans l'API Verified Permissions. Toutefois, 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 Amazon Cognito identités 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.

AWS Management Console

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

1. Ouvrez la [console des autorisations vérifiées](#). 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, pour Type principal, choisissez le type d'entité pour les principaux à partir de cette source. 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 de l'application client, choisissez si vous souhaitez valider l'application client IDs.
 - Pour valider l'application client IDs, sélectionnez Accepter uniquement les jetons avec l'application client correspondante IDs. 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 l'application cliente IDs si vous ne souhaitez pas valider l'application cliente IDs.
8. Choisissez Créer une source d'identité.
9. (Facultatif) Si votre magasin de politiques possède un schéma, 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 des Amazon Cognito jetons aux attributs principaux de Cedar, consultez [Mappage Amazon Cognito des jetons au schéma](#).

Note

Lorsque vous créez un [magasin de politiques lié à une API](#) ou que vous utilisez Configurer avec API Gateway un fournisseur d'identité pour créer des magasins de politiques, 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.

10. Créez des politiques qui utilisent les informations des jetons pour prendre des décisions d'autorisation. Pour de plus amples informations, veuillez consulter [Création de politiques statiques relatives aux autorisations vérifiées par Amazon](#).

Maintenant que vous avez créé une source d'identité, mis à jour le schéma et créé des politiques, les décisions `IsAuthorizedWithToken` d'autorisation sont prises par Verified Permissions. Pour plus d'informations, consultez le guide [IsAuthorizedWithToken](#) de référence de l'API Amazon Verified Permissions.

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 cette [CreateIdentitySource](#) opération.

L'exemple suivant crée une source d'identité qui peut accéder aux identités authentifiées à partir d'un groupe Amazon Cognito d'utilisateurs.

1. Créez un `config.txt` fichier contenant les informations suivantes sur le groupe Amazon Cognito d'utilisateurs à utiliser par le `--configuration` paramètre de la `create-identity-source` commande.

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

2. Exécutez la commande suivante pour créer une source Amazon Cognito d'identité.

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

3. (Facultatif) Si votre magasin de politiques possède un schéma, 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 des Amazon Cognito jetons aux attributs principaux de Cedar, consultez [Mappage Amazon Cognito des jetons au schéma](#).

Note

Lorsque vous créez un [magasin de politiques lié à une API](#) ou que vous utilisez Configurer avec API Gateway un fournisseur d'identité pour créer des magasins de politiques, 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.

4. Créez des politiques qui utilisent les informations des jetons pour prendre des décisions d'autorisation. Pour de plus amples informations, veuillez consulter [Création de politiques statiques relatives aux autorisations vérifiées par Amazon](#).

Maintenant que vous avez créé une source d'identité, mis à jour le schéma et créé des politiques, les décisions `IsAuthorizedWithToken` d'autorisation sont prises par Verified Permissions. Pour plus d'informations, consultez le guide [IsAuthorizedWithToken](#) de référence de l'API Amazon Verified Permissions.

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.

Modification des sources d'identité Amazon Verified Permissions

Vous pouvez modifier certains paramètres de votre source d'identité après l'avoir créée. Vous ne pouvez pas modifier le type de source d'identité, vous devez supprimer la source d'identité et en créer une nouvelle pour passer de l'OIDC à l'OIDC ou Amazon Cognito à l'OIDC. Amazon Cognito 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.

AWS Management Console

Pour mettre à jour une source Amazon Cognito d'identité

1. Ouvrez la [console des autorisations vérifiées](#). 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 les groupes 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 de l'application client, choisissez si vous souhaitez valider l'application client IDs.
 - Pour valider l'application client IDs, sélectionnez Accepter uniquement les jetons avec l'application client correspondante IDs. 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 l'application cliente IDs si vous ne souhaitez pas valider l'application cliente IDs.
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 une source Amazon Cognito d'identité

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

1. Créez un `config.txt` fichier contenant les informations suivantes sur le groupe Amazon Cognito d'utilisateurs à utiliser par le `--configuration` paramètre de la `update-identity-source` commande.

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

2. Exécutez la commande suivante pour mettre à jour une source Amazon Cognito d'identité.

```
$ aws verifiedpermissions update-identity-source \
  --update-configuration file://config.txt \
  --policy-store-id 123456789012
{
```

```
"createdDate": "2023-05-19T20:30:28.214829+00:00",  
"identitySourceId": "ISEXAMPLEabcdefgh111111",  
"lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",  
"policyStoreId": "PSEXAMPLEabcdefgh111111"  
}
```

Note

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.

Mappage Amazon Cognito des jetons au schéma

Vous souhaitez peut-être ajouter une source d'identité à un magasin de politiques et mapper les revendications, ou jetons, des fournisseurs de polices au schéma de votre magasin de politiques. Vous pouvez automatiser ce processus en utilisant la [configuration guidée](#) pour créer votre magasin de politiques avec une source d'identité, ou en mettant à jour votre schéma manuellement une fois le magasin de politiques créé. Une fois que vous avez mappé les jetons au schéma, vous pouvez créer des politiques qui les référencent.

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 Amazon Cognito jetons 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 créés par le biais de 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 et créer un schéma renseigné avec les 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 Amazon Cognito jetons 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 l'utiliser Amazon Cognito comme source d'identité dans votre magasin de politiques d'autorisations vérifiées, vous devez avoir des attributs de fournisseur dans votre schéma. Le schéma est fixe et doit correspondre aux entités créées par les jetons du fournisseur [IsAuthorizedWithToken](#) ou dans les demandes [BatchIsAuthorizedWithToken](#) d'API. 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 au schéma des attributs de fournisseur qui correspondent aux entités créées à l'aide de demandes d'API. 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

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

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.

Amazon Cognito Les jetons d'identification fonctionnent avec la plupart des bibliothèques [dépendantes de l'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 Guide du Amazon Cognito développeur.

Réclamations utiles sous forme de jetons Amazon Cognito d'identification

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 *oudepartment*.

Dans les politiques qui font référence à des Amazon Cognito attributs dotés d'un `:` séparateur, référez les attributs dans le 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 Amazon Cognito développeur.

L'exemple de jeton d'identification suivant possède chacun des quatre types d'attributs. Elle inclut la réclamation Amazon Cognito spécifique `cognito:username`, la réclamation personnalisée `custom:employmentStoreCode`, la réclamation email standard et la réclamation 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 Amazon Cognito d'utilisateurs, vous spécifiez le type d'entité principale avec laquelle 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, consultez [Modification des schémas du magasin de politiques](#). 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
      }
    }
  }
}
```

Pour un exemple de politique qui sera validée par rapport à ce schéma, consultez [Reflète Amazon Cognito Attributs des jetons d'identification](#).

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.

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

Réclamations utiles concernant les jetons Amazon Cognito d'accès

client_id

L'ID de l'application cliente d'une partie utilisatrice de l'OIDC. Avec 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 oudepartement. 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 Guide du Amazon Cognito développeur.

Un jeton d' Amazon Cognito accès 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-TOKEN11111111",
  "event_id": "bda909cb-3e29-4bb8-83e3-ce6808f49011",
  "token_use": "access",
  "scope": "MyAPI/mydata.write",
  "auth_time": 1688092966,
  "exp": 1688096566,
  "iat": 1688092966,
  "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN22222222",
  "username": "alice"
}
```

L'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, consultez [Modification des schémas du magasin de politiques](#).

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


`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 Amazon Cognito d'identité. 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
      }
    }
  }
}
```

```
}  
  }  
}
```

Pour un exemple de politique qui sera validée par rapport à ce schéma et utilisera la notation par points, voir [Utilise la notation par points pour référencer les attributs](#).

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 provider 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 de plus amples informations, veuillez consulter [Cartographie des jetons d'accès.](#)

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 Amazon Cognito d'identité, vous avez le choix du 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 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:username 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"
  ]
}
```

Pour un exemple de politique qui sera validée par rapport à ce schéma, consultez [Utilise la notation entre crochets pour référencer les attributs des jetons](#).

Validation du client et du public pour Amazon Cognito

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, Amazon Cognito et entre les sources d'identité 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 l'instance du fournisseur d'identité utilisée par votre application, par exemple `1example23456789`. Une audience est un chemin d'URL associé à la partie utilisatrice prévue, ou à la destination, du jeton d'accès, par exemple `https://mytoken.example.com`. Lorsque vous utilisez des jetons d'accès, la aud réclamation est toujours associée au public.

Amazon Cognito Les jetons d'identification comportent une aud réclamation contenant l'ID du [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 de l'application cliente dans votre source d'identité, Verified Permissions compare cette liste de clients d'applications IDs 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 Amazon Cognito les sources d'identité.

Autorisation côté client pour JWTs

Vous souhaitez 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 une application utilisant un JWT.¹

```
async function authorizeUsingJwtToken(jwtToken) {  
  
    const payload = await verifier.verify(jwtToken);
```

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

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

    return authResult;
  }

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

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

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

¹ Cet exemple de code utilise la [aws-jwt-verify](#) bibliothèque pour vérifier la JWTs signature par OIDC compatible IdPs.

Travailler avec les 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 JWTs produisent leurs produits en tant que 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.
- Enregistrements CNAME qui n'incluent pas de caractères génériques. Par exemple, ne `a.example.com` peut pas être mappé à `*.example.net`. Inversement, ne `*.example.com` peut pas être mappé à `a.example.net`
- 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 à l'entité IDs utilisateur et 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 provider` 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"`

Pour plus d'informations sur l'utilisation des opérations d'API pour autoriser les demandes provenant de sources OIDC, consultez [Opérations d'API disponibles pour l'autorisation](#).

L'exemple suivant 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.

Notez que lorsque vous référencez un groupe dans le principal, vous devez utiliser l'opérateur `in` pour que la politique soit correctement évaluée.

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

Rubriques

- [Création de sources d'identité OIDC Amazon Verified Permissions](#)
- [Modification des sources d'identité OIDC d'Amazon Verified Permissions](#)
- [Mappage des jetons OIDC au schéma](#)
- [Validation du client et du public pour les fournisseurs OIDC](#)

Création de sources d'identité OIDC Amazon Verified Permissions

La procédure suivante ajoute une source d'identité à un magasin de politiques existant.

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 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 [create-identity-source](#) dans AWS CLI ou [CreateIdentitySource](#) 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 Amazon Cognito identités 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.

AWS Management Console

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

1. Ouvrez la [console des autorisations vérifiées](#). 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 de plus amples informations, veuillez consulter [Mappage des jetons OIDC au schéma](#).
7. Dans Map token claims to schema entities, choisissez une entité User et User claim 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. (Facultatif) Dans les revendications de jeton de carte relatives à des entités de schéma, choisissez une entité de groupe et une revendication 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 (facultatif), entrez le client IDs ou le public URLs 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. (Facultatif) Si votre magasin de politiques possède un schéma, 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 des jetons OIDC aux attributs principaux de Cedar, consultez [Mappage des jetons OIDC au schéma](#).
12. Créez des politiques qui utilisent les informations des jetons pour prendre des décisions d'autorisation. Pour de plus amples informations, veuillez consulter [Création de politiques statiques relatives aux autorisations vérifiées par Amazon](#).

Maintenant que vous avez créé une source d'identité, mis à jour le schéma et créé des politiques, les décisions `IsAuthorizedWithToken` d'autorisation sont prises par Verified Permissions. Pour plus d'informations, consultez le guide [IsAuthorizedWithToken](#) de référence de l'API Amazon Verified Permissions.

AWS CLI

Pour créer une source d'identité OIDC

Vous pouvez créer une source d'identité à l'aide de cette [CreateIdentitySource](#) opération. L'exemple suivant crée une source d'identité qui peut accéder aux identités authentifiées auprès d'un fournisseur d'identité OIDC (IdP).

1. Créez un `config.txt` fichier contenant les détails suivants d'un IdP OIDC à utiliser par `--configuration` le paramètre de `create-identity-source` la commande.

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

```
    }  
  }  
}
```

2. Exécutez la commande suivante pour créer une source d'identité OIDC.

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

3. (Facultatif) Si votre magasin de politiques possède un schéma, 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 des jetons OIDC aux attributs principaux de Cedar, consultez [Mappage des jetons OIDC au schéma](#).
4. Créez des politiques qui utilisent les informations des jetons pour prendre des décisions d'autorisation. Pour de plus amples informations, veuillez consulter [Création de politiques statiques relatives aux autorisations vérifiées par Amazon](#).

Maintenant que vous avez créé une source d'identité, mis à jour le schéma et créé des politiques, les décisions `IsAuthorizedWithToken` d'autorisation sont prises par Verified Permissions. Pour plus d'informations, consultez le guide [IsAuthorizedWithToken](#) de référence de l'API Amazon Verified Permissions.

Modification des sources d'identité OIDC d'Amazon Verified Permissions

Vous pouvez modifier certains paramètres de votre source d'identité après l'avoir créée. Vous ne pouvez pas modifier le type de source d'identité, vous devez supprimer la source d'identité et en créer une nouvelle pour passer de l'OIDC à l'OIDC ou Amazon Cognito à l'OIDC. Amazon Cognito 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.

AWS Management Console

Pour mettre à jour une source d'identité OIDC

1. Ouvrez la [console des autorisations vérifiées](#). 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 de1ete 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 [UpdateIdentitySource](#) cette opération. L'exemple suivant met à jour la source d'identité spécifiée pour utiliser un autre fournisseur OIDC.

1. Créez un `config.txt` fichier contenant les détails suivants d'un IdP OIDC à utiliser par -- configuration le paramètre de `update-identity-source` la commande.

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

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

2. Exécutez la commande suivante pour mettre à jour une source d'identité OIDC.

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

Note

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.

Mappage des jetons OIDC au schéma

Vous souhaitez peut-être ajouter une source d'identité à un magasin de politiques et mapper les revendications, ou jetons, des fournisseurs de polices au schéma de votre magasin de politiques. Vous pouvez automatiser ce processus en utilisant la [configuration guidée](#) pour créer votre magasin de politiques avec une source d'identité, ou en mettant à jour votre schéma manuellement une fois le magasin de politiques créé. Une fois que vous avez mappé les jetons au schéma, vous pouvez créer des politiques qui les référencent.

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 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 créés par le biais de 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 et créer un schéma renseigné avec les attributs utilisateur. Dans l'autorisation par jeton d'identification, Verified Permissions associe les revendications aux attributs d'une entité principale.

Pour utiliser 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. Le schéma est fixe et doit correspondre aux entités créées par les jetons du fournisseur [IsAuthorizedWithToken](#) ou dans les demandes [BatchIsAuthorizedWithToken](#) d'API. 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 au schéma des attributs de fournisseur qui correspondent aux entités créées à l'aide de demandes d'API. Vous pouvez ensuite écrire des politiques à l'aide des attributs du jeton du fournisseur.

Rubriques

- [Associer les jetons d'identification au schéma](#)
- [Cartographie des jetons d'accès](#)
- [Ce qu'il faut savoir sur le mappage de schémas](#)

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.

Travailler avec des jetons d'identification d'un fournisseur OIDC revient à peu près à travailler avec des jetons Amazon Cognito d'identification. 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 de plus amples informations, veuillez consulter [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"
      }
    }
  }
}
```

Pour un exemple de politique qui sera validée par rapport à ce schéma, consultez [Reflète les attributs du jeton d'identification OIDC](#).

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.

La plupart des jetons d'accès provenant de fournisseurs OIDC externes s'alignent étroitement sur les jetons d' Amazon Cognito accès. 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, consultez [Modification des schémas du magasin de politiques](#).

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

Pour un exemple de politique qui sera validée par rapport à ce schéma, consultez [Reflète les attributs du jeton d'accès OIDC](#).

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 provider 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 de plus amples informations, veuillez consulter [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 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 politiques. 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:username 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"
      ]
    }
  }
}
```

Pour un exemple de politique qui sera validée par rapport à ce schéma, consultez [Utilise la notation entre crochets pour référencer les attributs des jetons](#).

Validation du client et du public pour les fournisseurs OIDC

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, Amazon Cognito et entre les sources d'identité 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 l'instance du fournisseur d'identité utilisée par votre application, par exemple `1example23456789`. Une audience est un chemin d'URL associé à la partie utilisatrice prévue, ou à la destination, du jeton d'accès, par exemple `https://mytoken.example.com`. Lorsque vous utilisez des jetons d'accès, la aud réclamation est toujours associée au public.

Les jetons d'identification OIDC ont une aud réclamation qui contient un client IDs, telle que `1example23456789`.

Les jetons d'accès OIDC ont une aud réclamation qui contient l'URL d'audience du jeton, telle que `https://myapplication.example.com`, et une `client_id` réclamation qui contient le client IDs, telle que `1example23456789`.

Lorsque vous configurez votre magasin de politiques, entrez une ou plusieurs valeurs pour la validation d'audience que votre magasin de politiques utilisera pour valider l'audience d'un jeton.

- **Jetons d'identification** — Verified Permissions valide l'identifiant du client en vérifiant qu'au moins un membre du client indiqué IDs dans la aud réclamation correspond à une valeur de validation d'audience.
- **Jetons d'accès** : les autorisations vérifiées valident l'audience en vérifiant que l'URL figurant dans la aud réclamation correspond à une valeur de validation de l'audience. Si aucune aud réclamation n'existe, l'audience peut être validée à l'aide des `client_id` revendications `cid` or. Vérifiez auprès de votre fournisseur d'identité la bonne déclaration d'audience et le bon format.

Autorisation côté client pour JWTs

Vous souhaitez 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 une application utilisant un JWT.¹

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

    let principalEntity = {
        entityType: "PhotoFlash::User", // the application needs to fill in the
relevant user type
        entityId: payload["sub"], // the application need to use the claim that
represents the user-id
    };
    let resourceEntity = {
        entityType: "PhotoFlash::Photo", //the application needs to fill in the
relevant resource type
        entityId: "jane_photo_123.jpg", // the application needs to fill in the
relevant resource id
    };
    let action = {
        actionType: "PhotoFlash::Action", //the application needs to fill in the
relevant action id
        actionId: "GetPhoto", //the application needs to fill in the relevant action
type
    };
    let entities = {
```

```
    entityList: [],
  };
  entities.entityList.push(...getUserEntitiesFromToken(payload));
  let policyStoreId = "PSEXAMPLEEabcdefg111111"; // set your own policy store id

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

  return authResult;
}

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

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

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

¹ Cet exemple de code utilise la [aws-jwt-verify](#) bibliothèque pour vérifier la JWTs signature par OIDC compatible IdPs.

Intégrations pour les autorisations vérifiées par Amazon

Les intégrations Amazon Verified Permissions vous aident à implémenter des autorisations précises dans vos applications tout en minimisant le code et en suivant les meilleures pratiques spécifiques au framework. Ces intégrations fournissent des composants intergiciels et des utilitaires qui connectent facilement votre application aux autorisations vérifiées.

Grâce aux intégrations, vous pouvez :

- Implémenter l'autorisation en quelques minutes
- Suivez les modèles et les conventions spécifiques au framework
- Réduisez les frais de maintenance
- Minimiser les erreurs potentielles de mise en œuvre de
- Concentrez-vous sur la logique métier plutôt que sur le code d'autorisation

Lorsqu'elles sont ajoutées à votre application, les intégrations effectuent les opérations suivantes :

1. Interceptez les demandes entrantes via un intergiciel spécifique au framework
2. Extraire le contexte d'autorisation pertinent des demandes
3. Déterminer les décisions d'autorisation à l'aide des autorisations vérifiées
4. Appliquer le contrôle d'accès en fonction des résultats d'autorisation

Verified Permissions prend actuellement en charge les frameworks suivants :

- [Express.js pour les applications Node.js](#)

Intégrer Express aux autorisations vérifiées d'Amazon

L'intégration de Verified Permissions Express fournit une approche basée sur un intergiciel pour implémenter l'autorisation dans vos applications Express.js. Grâce à cette intégration, vous pouvez protéger vos points de terminaison d'API à l'aide de politiques d'autorisation précises sans modifier vos gestionnaires de route existants. L'intégration gère automatiquement les contrôles d'autorisation en interceptant les demandes, en les évaluant par rapport à vos politiques définies et en garantissant que seuls les utilisateurs autorisés peuvent accéder aux ressources protégées.

Cette rubrique explique comment configurer l'intégration Express, de la création d'un magasin de politiques à la mise en œuvre et au test du middleware d'autorisation. En suivant ces étapes, vous pouvez ajouter des contrôles d'autorisation robustes à votre application Express avec un minimum de modifications de code.

Les GitHub dépôts suivants sont référencés dans cette rubrique :

- [cedar-policy/ authorization-for-expressjs](#) - Le middleware d'autorisation Cedar pour Express.js
- [verifiedpermissions/ authorization-clients-js](#) - Les clients d'autorisation Verified Permissions pour JavaScript
- [verifiedpermissions/examples/express-petstore](#) - Exemple d'implémentation à l'aide du middleware Express.js

Conditions préalables

Avant de mettre en œuvre l'intégration Express, assurez-vous de disposer des éléments suivants :

- Un [AWS compte](#) avec accès aux autorisations vérifiées
- [Node.js](#) et [npm installés](#)
- Une application [Express.js](#)
- Un fournisseur d'identité OpenID Connect (OIDC) (tel que) [Amazon Cognito](#)
- [AWS CLI](#) configuré avec les autorisations appropriées

Configuration de l'intégration

Étape 1 : créer un magasin de politiques

Créez un magasin de politiques à l'aide de AWS CLI :

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

Note

Enregistrez l'ID du magasin de politiques renvoyé dans la réponse pour l'utiliser lors des étapes suivantes.

Étape 2 : Installation des dépendances

Installez les packages requis dans votre application Express :

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

Configuration de l'autorisation

Étape 1 : générer et télécharger le schéma Cedar

Un schéma définit le modèle d'autorisation d'une application, y compris les types d'entités de l'application et les actions que les utilisateurs sont autorisés à effectuer. Nous vous recommandons de définir un espace de [noms](#) pour votre schéma. Dans cet exemple, nous utilisons `YourNamespace`. Vous attachez votre schéma à vos magasins de politiques d'autorisations vérifiées, et lorsque des politiques sont ajoutées ou modifiées, le service valide automatiquement les politiques par rapport au schéma.

Le `@cedar-policy/authorization-for-expressjs` package peut analyser les [spécifications OpenAPI](#) de votre application et générer un schéma Cedar. Plus précisément, l'objet `paths` est requis dans votre spécification.

Si vous n'avez pas de spécification OpenAPI, vous pouvez suivre les instructions rapides du [express-openapi-generator](#) package pour générer une spécification OpenAPI.

Générez un schéma à partir de votre spécification OpenAPI :

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

Ensuite, formatez le schéma Cedar pour l'utiliser avec AWS CLI. Pour plus d'informations sur le format spécifique requis, consultez [Schéma du Policy Store](#). Si vous avez besoin d'aide pour formater le schéma, un script est appelé `prepare-cedar-schema.sh` dans le dépôt [GitHubverifiedpermissions/examples](#). Voici un exemple d'appel à ce script qui génère le schéma au format Verified Permissions dans le `v2.cedarschema.forAVP.json` fichier.

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

Téléchargez le schéma formaté dans votre magasin de politiques, en le `policy-store-id` remplaçant par l'ID de votre magasin de politiques :

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

Étape 2 : créer des politiques d'autorisation

Si aucune politique n'est configurée, Cedar refuse toutes les demandes d'autorisation. L'intégration du framework Express permet de démarrer ce processus en générant des exemples de politiques basés sur le schéma généré précédemment.

Lorsque vous utilisez cette intégration dans vos applications de production, nous vous recommandons de créer de nouvelles politiques à l'aide des outils d'infrastructure en tant que code (iAAC). Pour de plus amples informations, veuillez consulter [Travailler avec AWS CloudFormation](#).

Générez des exemples de politiques Cedar :

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

Cela générera des exemples de politiques dans le `/policies` répertoire. Vous pouvez ensuite personnaliser ces politiques en fonction de vos cas d'utilisation. Par exemple :

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

```
resource
);
```

Formatez les politiques à utiliser avec AWS CLI. Pour plus d'informations sur le format requis, voir [create-policy](#) dans la AWS CLI référence. Si vous avez besoin d'aide pour formater les politiques, un script est appelé `convert_cedar_policies.sh` dans le référentiel [GitHubverifiedpermissions/examples](#). Voici un appel à ce script :

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

Téléchargez les politiques formatées dans Verified Permissions, en les `policy_1.json` remplaçant par le chemin et le nom de votre fichier de règles et `policy-store-id` par l'ID de votre magasin de politiques :

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

Étape 3 : Connecter un fournisseur d'identité

Par défaut, le middleware d'autorisation Verified Permissions lit un jeton Web JSON (JWT) fourni dans l'en-tête d'autorisation de la demande d'API pour obtenir des informations sur l'utilisateur. Les autorisations vérifiées peuvent valider le jeton en plus d'évaluer la politique d'autorisation.

Créez un fichier de configuration de source d'identité nommé `identity-source-configuration.txt` qui ressemble à ce qui suit avec votre `userPoolArn` et `clientId` :

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

Créez la source d'identité en exécutant la AWS CLI commande suivante, en la `policy-store-id` remplaçant par votre identifiant de magasin de politiques :

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

Implémentation du middleware d'autorisation

Mettez à jour votre application Express pour inclure le middleware d'autorisation. Dans cet exemple, nous utilisons des jetons d'identité, mais vous pouvez également utiliser des jetons d'accès. Pour plus d'informations, voir [authorization-for-expressjs](#) ci-dessous GitHub.

```
const { ExpressAuthorizationMiddleware } = require('@cedar-policy/authorization-for-expressjs');  
  
const { AVPAuthorizationEngine } = require('@verifiedpermissions/authorization-clients');  
  
const avpAuthorizationEngine = new AVPAuthorizationEngine({  
  policyStoreId: 'policy-store-id',  
  callType: 'identityToken'  
});  
  
const expressAuthorization = new ExpressAuthorizationMiddleware({  
  schema: {  
    type: 'jsonString',  
    schema: fs.readFileSync(path.join(__dirname, '../v4.cedarschema.json'),  
      'utf8'),  
  },  
  authorizationEngine: avpAuthorizationEngine,  
  principalConfiguration: { type: 'identityToken' },  
  skippedEndpoints: [],  
  logger: {  
    debug: (s) => console.log(s),  
    log: (s) => console.log(s),  
  }  
});  
  
// Add the middleware to your Express application  
app.use(expressAuthorization.middleware);
```

Tester l'intégration

Vous pouvez tester la mise en œuvre de votre autorisation en envoyant des demandes à vos points de terminaison d'API avec différents jetons utilisateur. Le middleware d'autorisation évaluera automatiquement chaque demande par rapport à vos politiques définies.

Par exemple, si vous avez configuré différents groupes d'utilisateurs dotés d'autorisations différentes :

- Administrateurs : accès complet à toutes les ressources et fonctions de gestion
- Employés : peuvent consulter, créer et mettre à jour des ressources
- Clients : peuvent uniquement consulter les ressources

Vous pouvez vérifier que les politiques d'autorisation fonctionnent comme prévu en vous connectant avec différents utilisateurs et en tentant diverses opérations. Dans le terminal de l'application Express, vous pouvez voir le résultat du journal qui fournit des informations supplémentaires sur les décisions d'autorisation.

Résolution des problèmes

Si vous rencontrez des problèmes d'autorisation, essayez ce qui suit :

- Vérifiez que l'identifiant de votre magasin de polices est correct
- Assurez-vous que votre source d'identité est correctement configurée
- Vérifiez que vos politiques sont correctement formatées
- Validez que vos jetons JWT sont valides

Étapes suivantes

Après avoir implémenté l'intégration de base, pensez à :

- Implémentation de mappers personnalisés pour des scénarios d'autorisation spécifiques
- Configuration de la surveillance et de la journalisation des décisions d'autorisation
- Création de politiques supplémentaires pour différents rôles d'utilisateur

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 d'autorisation 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 souhaitez 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 souhaitez 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 contiennent des informations provenant de l'extérieur APIs. 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 magasin DigitalPetStored'exemples de règles 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.

Rubriques

- [Opérations d'API disponibles pour l'autorisation](#)
- [Tester votre modèle d'autorisation](#)
- [Intégration de vos modèles d'autorisation aux applications](#)

Opérations d'API disponibles 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 évalue 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 des jetons Web JSON (JWTs). Verified Permissions fonctionne directement avec les fournisseurs OIDC, par exemple 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 une source d'identité.

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 dans le JWT que vous fournissez.

[BatchIsAuthorized](#)

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.

[BatchIsAuthorizedWithToken](#)

L'opération `BatchIsAuthorizedWithToken` 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 annuaire d'utilisateurs.

Comme pour le `IsAuthorizedWithToken` cas, 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 dans le JWT que vous fournissez.

Tester votre modèle d'autorisation

Pour comprendre l'effet de la décision d'autorisation Amazon Verified Permissions lorsque vous déployez votre application, vous pouvez évaluer vos politiques au fur [Utilisation du banc de test Amazon Verified Permissions](#) et à mesure que vous les développez à l'aide des demandes d'API REST HTTPS adressées à Verified Permissions. 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 [IsAuthorized](#) et [BatchIsAuthorized](#) sous forme de [demandes d' AWS API signées adressées](#) aux [points de terminaison des services](#) régionaux. [IsAuthorizedWithToken](#) 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 Permissions vérifiées dans la [console Permissions vérifiées](#). 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 [IsAuthorized](#) dans la référence de l'API Verified Permissions. Les détails suivants reproduisent les conditions de l'exemple 4 qui fait référence à l'DigitalPetStoreéchantillon.
 - 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 `parentDigitalPetStore::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 Permissions vérifiées dans la [console Permissions vérifiées](#). 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. [IsAuthorized](#) Dans la référence de l'API Verified Permissions, copiez le corps de la demande de l'exemple 4 qui fait référence à l'DigitalPetStoreexemple.

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 `policyStoreId` l'ID du magasin de politiques que vous avez indiqué précédemment.
6. Soumettez la demande et examinez les résultats. Dans un magasin `DigitalPetStore` de politiques 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'attribut `owner` 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 un utilisateur actif ou inactif.

Intégration de vos modèles d'autorisation aux applications

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. Rassemblez 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' Amazon Cognito identité](#) 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 SDKs. 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 SDKs, consultez la section [Outils pour Amazon Web Services](#).

Vous trouverez ci-dessous des liens vers la documentation des différentes ressources relatives aux autorisations vérifiées AWS SDKs.

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

- [AWS SDK pour Rust](#)

L' AWS SDK pour 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](#)

Effectuer des demandes d'API

Les demandes de requête pour les autorisations Amazon Verified sont des requêtes HTTP ou HTTPS qui utilisent cette POST méthode.

Points de terminaison d'autorisations vérifiés

Un point de terminaison est une URL qui sert de point d'entrée à un service Web. Vous pouvez sélectionner un point de Région AWS terminaison approprié lorsque vous faites vos demandes afin de réduire le temps de latence. Pour plus d'informations sur les points de terminaison utilisés par Verified Permissions, consultez [Amazon Verified Permissions](#) dans le Référence générale d'Amazon Web Services.

Paramètres de demande

Amazon Verified Permissions utilise un JSON-based protocole. Vous transmettez les paramètres de requête au format JSON dans le corps de votre requête HTTP. Les listes sont représentées sous forme de tableaux JSON. Pour plus d'informations, consultez la section [Paramètres communs](#) dans le manuel Amazon Verified Permissions API Reference.

Identifiants de demande

Dans chaque réponse, AWS renvoie un ID de demande via l'en-tête de réponse `x-amzn-RequestId` HTTP. Cette chaîne est un identifiant unique attribué pour fournir AWS des informations de suivi. Bien que l'ID de demande soit inclus dans chaque réponse, il n'est pas répertorié sur les pages de documentation de l'API individuelles afin d'améliorer la lisibilité et de réduire la redondance.

Authentification par API de requête

Vous envoyez des requêtes via HTTPS. Vous devez inclure une signature dans chaque demande de requête. Pour plus d'informations sur la création et l'inclusion d'une signature, consultez [la section Signature des demandes d' AWS API](#) dans le Référence générale d'Amazon Web Services.

Bibliothèques disponibles

AWS fournit des bibliothèques, des exemples de code, des didacticiels et d'autres ressources aux développeurs de logiciels qui préfèrent créer des applications à l'aide d'API spécifiques au langage

plutôt que des outils de ligne de commande et de l'API de requête. Ces bibliothèques fournissent des fonctions de base (non incluses dans les API), telles que l'authentification des demandes, les nouvelles tentatives et la gestion des erreurs, afin de faciliter le démarrage. Les bibliothèques et ressources d'autorisations vérifiées sont disponibles pour les langues et plateformes suivantes :

- [AWS SDK pour Go](#)
- [AWS SDK pour Java 2.x](#)
- [AWS SDK pour Java 1. x](#)
- [AWS SDK pour JavaScript](#)
- [AWS SDK pour .NET](#)
- [AWS SDK pour PHP](#)
- [AWS SDK pour Python \(Boto\)](#)
- [AWS SDK pour Ruby](#)

Pour plus d'informations sur les bibliothèques et les exemples de code dans tous les langages, voir [Exemples de code et bibliothèques](#).

Effectuer des demandes d'API à l'aide de la méthode **POST**

Si vous n'utilisez aucun des AWS SDK, vous pouvez effectuer des demandes d'autorisations vérifiées via HTTPS à l'aide de la méthode de POST demande. La POST méthode nécessite que vous spécifiez l'opération dans l'en-tête de la demande et que vous fournissiez les données de l'opération au format JSON dans le corps de la demande.

Nom de l'en-tête	Valeur d'en-tête
Host	Le point de terminaison Amazon Verified Permissions. Par exemple : <code>verifiedpermissions.us-east-1.amazonaws.com</code>
X-Amz-Date	Vous devez fournir l'horodatage dans l'en-tête HTTP Date ou dans l'en-tête AWS x-amz-date. Certaines bibliothèques clientes HTTP ne vous permettent pas de définir l'en-tête Date. Lorsqu'un en-tête x-amz-date est présent, le système ignore tout en-tête Date lors de l'authentification de la demande.

Nom de l'en-tête	Valeur d'en-tête
	L'en-tête x-amz-date doit être spécifié au format de base ISO 8601. Par exemple : 20130315T092054Z
Authorization	Ensemble de paramètres d'autorisation AWS utilisé pour garantir la validité et l'authenticité de la demande. Pour plus d'informations sur la création de cet en-tête, voir Processus de signature de la version 4 de Signature dans le Référence générale d'Amazon Web Services.
X-Amz-Target	Spécifie l'opération d'autorisations vérifiées que vous souhaitez effectuer. VerifiedPermissions. <i>API_Name</i> Par exemple, pour appeler l>CreatePolicy opération, utilisez la valeur cible suivante. VerifiedPermissions.CreatePolicy
Content-Type	Spécifie le format d'entrée. Utilisez la valeur suivante. application/x-amz-json-1.0
Accept	Spécifie le format de réponse. Utilisez la valeur suivante. application/x-amz-json-1.0
Content-Length	Taille de la charge utile en octets.
Content-Encoding	Spécifie le format de codage de l'entrée et de la sortie. Utilisez la valeur suivante. amz-1.0

Voici un exemple d'en-tête pour une requête HTTP renvoyant une liste de toutes les politiques du magasin de politiques spécifié, Compte AWS où elles font Principal référence à un User nomalice. Dans cet exemple, la Authorization ligne est encapsulée ici pour en faciliter la lecture. Ne l'inscrivez pas dans votre demande réelle.

```
POST / HTTP/1.1
```

```
Host: verifiedpermissions.us-east-1.amazonaws.com
X-Amz-Date: 20230101T200059Z
Accept-Encoding: identity
Content-Type: application/x-amz-json-1.0
X-Amz-Target: VerifiedPermissions.ListPolicies
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
  Signature=<Signature>
Content-Length: <PayloadSizeBytes>
```

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

Sécurité dans Amazon Verified Permissions

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de [AWS conformité Programmes](#) de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent aux autorisations vérifiées par Amazon, consultez [AWS Services concernés par programme de conformitéAWS](#) .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service 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 des autorisations vérifiées. Les rubriques suivantes expliquent comment configurer les autorisations vérifiées pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources d'autorisations vérifiées.

Rubriques

- [Protection des données dans Amazon Verified Permissions](#)
- [Gestion des identités et des accès pour Amazon Verified Permissions](#)
- [Validation de conformité pour les autorisations vérifiées par Amazon](#)
- [Résilience dans les autorisations vérifiées par Amazon](#)

Protection des données dans Amazon Verified Permissions

Le modèle de [responsabilité AWS partagée Le modèle](#) s'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 des AWS Cloud s'exécute. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Ce contenu inclut la configuration de la sécurité et les tâches de gestion pour le Services AWS produit que vous utilisez. Pour plus d'informations 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 d'AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécuritéAWS .

- À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (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.
 - SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous avons besoin du protocole TLS 1.2.
 - Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
 - Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
 - Utilisez des services de sécurité gérés avancés tels que Amazon Macie, qui aident à découvrir et à sécuriser les données sensibles stockées dans Amazon S3.
 - Si vous avez besoin de modules cryptographiques validés FIPS 140-2 lorsque vous accédez à AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations 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 autre Services AWS à l'aide de la console, de l'API ou AWS SDKs. AWS CLI Toutes les données que vous entrez 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 principes). Dans un environnement de test, vous pouvez choisir d'utiliser des identifiants d'entité simples, tels que `jane` ou `bob` pour le nom d'une entité de type `User`. Cependant, 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 les identificateurs uniques universels (UUIDs). Prenons l'exemple de l'utilisateur `jane` qui quitte l'entreprise. Plus tard, vous permettez à quelqu'un d'autre d'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"`. Permissions vérifiées et Cedar ne peut 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 sont jamais réutilisés afin de ne pas autoriser l'accès par inadvertance en raison de la présence d'un ancien identifiant dans une politique.

- Assurez-vous que les chaînes que vous souhaitez définir `Long` et `Decimal` les valeurs se situent dans la plage valide de chaque type. Assurez-vous également que l'utilisation d'opérateurs arithmétiques n'aboutit pas à 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 ne pas bloquer l'accès de manière inattendue.

Chiffrement des données

Amazon Verified Permissions chiffre automatiquement toutes les données des clients, telles que les politiques, avec un Clé gérée par AWS. Amazon Verified Permissions permet également aux clients d'utiliser un clé gérée par le client pour chiffrer leurs données.

Pour obtenir des informations détaillées sur l'utilisation des clés gérées par le client pour le chiffrement, consultez [the section called "Clés gérées par le client"](#).

Chiffrer les ressources dans les autorisations vérifiées d'Amazon

Amazon Verified Permissions fournit un chiffrement par défaut pour protéger les données sensibles des clients au repos à l'aide de clés de chiffrement AWS détenues par nos soins. Comme couche de protection supplémentaire, Amazon Verified Permissions vous permet de chiffrer vos magasins de politiques à l'aide de AWS Key Management Service (AWS KMS) clé gérée par le client s (CMK). Cette fonctionnalité garantit la protection des données sensibles grâce au chiffrement au repos, ce qui vous permet de :

- Réduisez la charge opérationnelle de votre application afin de protéger les données sensibles
- Gardez le contrôle sur qui peut consulter les détails de vos politiques d'autorisation via vos propres AWS KMS clé gérée par le client
- Créer des applications sécurisées qui répondent aux exigences réglementaires et de conformité strictes en matière de chiffrement

Les sections suivantes expliquent comment configurer le chiffrement pour les nouveaux magasins de politiques et comment gérer vos clés de chiffrement.

AWS KMS Types de clés pour les autorisations vérifiées par Amazon

Amazon Verified Permissions s'intègre AWS KMS pour gérer les clés de chiffrement utilisées pour les données des encrypting/decrypting clients. Pour plus d'informations sur les types de clés et leurs états, consultez [Concepts AWS Key Management Service](#) dans le Guide du développeur AWS KMS . Lorsque vous créez un nouveau magasin de politiques, vous pouvez choisir parmi les types de AWS KMS clés suivants pour chiffrer vos données :

AWS Clé possédée

Type de chiffrement par défaut. Amazon Verified Permissions détient la clé sans frais supplémentaires pour vous et chiffre les données des ressources au repos lors de leur création. Aucune configuration supplémentaire n'est requise dans votre code ou dans les applications de encrypt/decrypt vos données à l'aide de la clé détenue par Verified Permissions.

Clé gérée par le client

Vous créez, détenez et gérez la clé de votre AWS compte. Vous avez le contrôle total de la AWS KMS clé. AWS KMS des frais s'appliquent pour clé gérée par le client nous. Pour plus d'informations, consultez la page [Tarification d'AWS KMS](#). Pour plus d'informations sur les types de clés, consultez la section [Clés gérées par le client](#) dans le Guide du AWS KMS développeur.

Lorsque vous spécifiez un chiffrement clé gérée par le client pour les ressources de haut niveau (c'est-à-dire le magasin de politiques), Verified Permissions chiffre la ressource, ainsi que ses ressources secondaires, avec cette clé. Pour chiffrer un magasin de politiques à l'aide d'un clé gérée par le client, vous devez accorder l'accès aux autorisations vérifiées dans votre politique clé. Une politique clé est une politique [basée sur les ressources que vous associez à votre politique](#) pour contrôler clé gérée par le client l'accès à celle-ci. Pour plus d'informations, consultez [the section called "Autoriser l'utilisation de votre AWS KMS clé pour les autorisations vérifiées par Amazon"](#).

En outre, pour créer un magasin de règles chiffré avec un clé gérée par le client, ou pour effectuer des appels d'API vers un magasin de politiques chiffré par un clé gérée par le client, l'utilisateur ou le rôle IAM qui effectue l'appel doit également avoir accès à la clé. Si Verified Permissions ne parvient pas à accéder à la clé, toute décision d'autorisation impliquant des ressources chiffrées par cette clé peut être périmée ou inexacte. Si vous n'avez pas accès à la clé, vous ne pourrez pas accéder aux read/update/delete ressources chiffrées par cette clé, et tout appel de création visant à utiliser la clé à des fins de chiffrement échouera.

Note

Le chiffrement des autorisations vérifiées au repos est disponible dans toutes les AWS régions où les autorisations vérifiées sont disponibles.

Important

Une fois que clé gérée par le client a été utilisé pour chiffrer un magasin de règles, vous **NE POUVEZ PAS** mettre à jour la ressource pour utiliser une autre clé pour le chiffrement ou supprimer la clé de ce magasin de politiques.

Utilisation AWS KMS et clés de données avec autorisations vérifiées par Amazon

La fonctionnalité de chiffrement au repos d'Amazon Verified Permissions utilise une AWS KMS clé et une hiérarchie de clés de données pour protéger les données de vos ressources.

Note

Amazon Verified Permissions ne prend en charge que AWS KMS les clés symétriques. Vous ne pouvez pas utiliser de AWS KMS clé asymétrique pour chiffrer vos ressources Amazon Verified Permissions.

Utilisation de clés AWS détenues

Amazon Verified Permissions chiffre toutes les ressources par défaut à l'aide de clés AWS détenues. Ces clés sont gratuites et peuvent être modifiées chaque année afin de protéger les ressources de votre compte. Vous n'avez pas besoin de consulter, de gérer, d'utiliser ni d'auditer ces clés. Aucune action n'est donc requise pour la protection des données. Pour plus d'informations sur les clés AWS détenues, consultez la section [clés AWS détenues](#) dans le Guide du AWS KMS développeur.

Utilisation de clés gérées par le client

La sélection d'un clé gérée par le client pour le chiffrement offre les avantages suivants :

- Vous créez et gérez la AWS KMS clé, notamment en définissant les politiques clés et IAM les politiques pour contrôler l'accès à la AWS KMS clé. Vous pouvez activer et désactiver la AWS KMS clé, activer et désactiver la rotation automatique des touches et supprimer la AWS KMS clé lorsqu'elle n'est plus utilisée.
- Vous pouvez en utiliser un clé gérée par le client avec du matériel clé importé ou clé gérée par le client dans un magasin de clés personnalisé que vous possédez et gérez.
- Vous pouvez vérifier le chiffrement et le déchiffrement de vos ressources d'autorisations vérifiées en examinant les appels d'API Amazon Verified Permissions envoyés AWS KMS dans AWS CloudTrail les journaux.

Pour Amazon Verified Permissions, vous pouvez utiliser votre clé gérée par le client s pour les encryption/decryption, you will need to add specific key policies to allow Amazon Verified Permissions to encrypt/decrypt ressources en votre nom.

Autoriser l'utilisation de votre AWS KMS clé pour les autorisations vérifiées par Amazon

Amazon Verified Permissions nécessite au minimum les autorisations suivantes sur un clé gérée par le client :

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

Vous trouverez ci-dessous un exemple de politique clé :

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

Comprendre le contexte de la source

Le contexte source fournit des informations sur l'appelant source qui tente d'effectuer des AWS KMS actions sur une clé donnée. Cela permet d'éviter toute confusion ou toute utilisation abusive des données cryptées en liant le contexte à la source des données.

Les clients peuvent utiliser le contexte source comme conditions supplémentaires relatives à leur politique principale, telles que les déclarations de politique clés suivantes :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "Enable this account full access to this key",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": "kms:*",
    "Resource": "*"
  },
  {
    "Sid": "Enable AVP to retrieve this key's metadata",
    "Effect": "Allow",
    "Principal": {
      "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      },
      "StringLike": {
        "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
      }
    }
  },
  {
    "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
    "Effect": "Allow",
    "Principal": {
      "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": [
      "kms:Decrypt",
      "kms:ReEncryptTo",
      "kms:ReEncryptFrom",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:Encrypt"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      },

```

```
    "StringLike": {
      "aws:SourceArn": "arn:aws:verifiedpermissions::111122223333:policy-
store/*"
    }
  }
]
```

Cette politique clé permet à Verified Permissions de AWS KMS passer des appels en votre nom, si le compte source est le même que le compte sur AWS KMS auquel réside cette clé. Ces valeurs doivent être vérifiables lors de la AWS CloudTrail vérification des journaux d'audit pour la clé CMK. Pour plus d'informations sur les clés de AWS condition globales, consultez la section [Utilisation des clés de aws:SourceAccount condition aws:SourceArn ou](#) de condition.

Comprendre le contexte de chiffrement

Le contexte de chiffrement est un ensemble de paires clé-valeur qui contiennent des données authentifiées supplémentaires pour les contrôles d'intégrité du chiffrement. Lorsque vous incluez un contexte de chiffrement dans une demande de chiffrement de données, lie AWS KMS cryptographiquement le contexte de chiffrement aux données chiffrées. Pour déchiffrer les données, vous devez transmettre le même contexte de chiffrement.

Amazon Verified Permissions utilise le même contexte de chiffrement dans toutes les opérations AWS KMS cryptographiques et peut être vérifiée dans les AWS CloudTrail journaux lorsque Verified Permissions passe des AWS KMS appels en votre nom pour encryption/decryption des processus. Par défaut, Verified Permissions utilise les paires clé-valeur du contexte de chiffrement suivantes pour chiffrer vos ressources :

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

Amazon Verified Permissions vous permet également d'ajouter un contexte de chiffrement personnalisé dans le cadre des métadonnées supplémentaires que vous souhaitez inclure au cours des encryption/decryption processus. Cela signifie que votre politique clé peut être plus précise en matière d'octroi d'autorisations, comme dans l'exemple ci-dessous :

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "Enable this account full access to this key",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": "kms:*",
    "Resource": "*"
  },
  {
    "Sid": "Enable AVP to retrieve this key's metadata",
    "Effect": "Allow",
    "Principal": {
      "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
  },
  {
    "Sid": "Enable AVP to encrypt/decrypt resources utilizing this key",
    "Effect": "Allow",
    "Principal": {
      "Service": "verifiedpermissions.amazonaws.com"
    },
    "Action": [
      "kms:Decrypt",
      "kms:ReEncryptTo",
      "kms:ReEncryptFrom",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:Encrypt"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/*",
        "kms:EncryptionContext:policy_owner": "Tim"
      }
    }
  }
]

```

```
}
```

Cette politique de clé permet à Verified Permissions de passer des AWS KMS appels en votre nom, si la carte du contexte de chiffrement contient une clé `aws:verifiedpermissions:policy-store-arn` dont la valeur suit le format de `arn:aws:verifiedpermissions::111122223333:policy-store/*` et contient également une paire clé-valeur. `"policy_owner": "Tim"` Découvrez [the section called “Création d'un magasin de politiques crypté”](#) comment définir un contexte de chiffrement personnalisé.

Note

Il est recommandé que les politiques clés dont les conditions sont basées sur le contexte de chiffrement concernent un sous-ensemble de la carte du contexte de chiffrement, plutôt que de vérifier chaque paire clé-valeur. Le service et ses dépendances en amont peuvent ajouter des paires clé-valeur supplémentaires qui ne sont pas visibles pour vous, et peuvent affecter l'accès aux clés des autorisations vérifiées si la politique des clés l'autorise de manière conditionnelle en fonction de l'apparence exacte de la carte du contexte de chiffrement.

Compréhension kms: ViaService

La clé de `kms:ViaService` condition limite l'utilisation d'une AWS KMS clé aux demandes émanant de AWS services spécifiques. Cette clé de condition s'applique uniquement [aux sessions d'accès direct](#) (FAS). Pour plus d'informations sur `kms:ViaService`, voir [kms : ViaService](#) dans le manuel du AWS KMS développeur.

Par exemple, la déclaration de politique clé suivante utilise la clé de `kms:ViaService` condition pour autoriser l'utilisation de a [clé gérée par le client](#) pour les actions spécifiées uniquement lorsque la demande provient d'Amazon Verified Permissions dans la région USA Est (Virginie du Nord) au nom de `BrentRole`.

```
{
  "Sid": "Enable AVP to encrypt/decrypt resources using credentials of BrentRole",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/BrentRole"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKeyWithoutPlaintext",
```

```

    "kms:Encrypt",
    "kms:ReEncryptFrom",
    "kms:ReEncryptTo",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "verifiedpermissions.us-east-1.amazonaws.com"
      ]
    }
  }
}

```

Cela est nécessaire pour que Verified Permissions puisse transmettre votre identité, vos autorisations et vos attributs de session lorsque Verified Permissions fait une demande de AWS KMS chiffrement/déchiffrement en votre nom. Pour plus d'informations sur les demandes FAS, consultez la section [Sessions d'accès transféré](#) dans le guide de l'IAM utilisateur.

Politique AWS KMS clé complète

Sur la base des concepts présentés dans les sections précédentes, voici un exemple de politique clé qui permettra à Amazon Verified Permissions d'utiliser une clé CMK pour le chiffrement/déchiffrement :

```

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

```

```

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

```

```

    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/BrentRole"
    },
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:Encrypt",
      "kms:ReEncryptFrom",
      "kms:ReEncryptTo",
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": [
          "verifiedpermissions.us-east-1.amazonaws.com"
        ]
      },
      "StringLike": {
        "kms:EncryptionContext:aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/*",
        "kms:EncryptionContext:policy_owner": "Tim"
      }
    }
  }
}

```

Warning

Faites preuve de prudence lorsque vous modifiez AWS KMS les politiques relatives aux clés déjà utilisées par Amazon Verified Permissions. Bien que les autorisations vérifiées valident les autorisations de chiffrement et de déchiffrement lorsque vous configurez initialement une AWS KMS clé lors de la création de ressources de haut niveau, elles ne peuvent pas vérifier les modifications de politique ultérieures à la demande. La suppression involontaire des autorisations nécessaires peut perturber vos décisions d'autorisation et les flux de service réguliers des autorisations vérifiées. Pour obtenir des conseils sur la résolution des erreurs courantes liées à clé gérée par le client s dans Amazon Verified Permissions, reportez-vous à [the section called “Résoudre les problèmes liés aux clés gérées par le client dans les autorisations vérifiées par Amazon”](#).

IAM Politiques nécessaires pour les ressources chiffrées

Les clients qui appellent Verified Permissions via un IAM rôle au sein de leur compte devront s'assurer que la IAM politique correspondante dispose des autorisations appropriées clé gérée par le client pour les utiliser à des fins de chiffrement et de déchiffrement des ressources.

Pour créer des magasins de politiques chiffrés par un clé gérée par le client, la IAM politique suivante illustre le strict minimum nécessaire AWS KMS et les actions d'autorisations vérifiées pour ce faire :

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

Note

Pour récupérer (opérations Get* et List*) et supprimer les magasins de politiques chiffrés par un clé gérée par le client, aucune autorisation supplémentaire n'est nécessaire.

Pour mettre à jour un magasin de règles chiffré par a clé gérée par le client, récupérer (opérations Get* et List*), mettre à jour et supprimer les ressources enfants d'un magasin de politiques chiffré par

a clé gérée par le client, la IAM politique suivante illustre le strict minimum nécessaire AWS KMS et les actions d'autorisations vérifiées permettent de le faire :

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

Dans le cadre d'une IAM politique unique, les clients peuvent simplement ajouter les éléments suivants à leur politique de IAM rôle :

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

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

Gestion des magasins de politiques chiffrés

Les magasins de politiques constituent le conteneur d'entrée de gamme qui contiendra toutes les ressources de politiques associées. Pour plus d'informations sur les magasins de politiques et la hiérarchie des ressources pour enfants, consultez les [magasins de politiques Amazon Verified Permissions](#) dans le guide de l'utilisateur Amazon Verified Permissions.

Lorsque vous créez un magasin de politiques dans Verified Permissions, vous pouvez activer le chiffrement au repos à l'aide de AWS KMS clés. Cela garantit que :

- Toutes les opérations de lecture, de mise à jour et de suppression sur les magasins de politiques, et leurs ressources secondaires, utiliseront les processus clé gérée par le client de déchiffrement fournis
- Tout appel de décision d'autorisation (c'est-à-dire `IsAuthorized` `BatchIsAuthorized` `IsAuthorizedWithToken`, etc.) utilisera les processus clé gérée par le client de déchiffrement fournis

Création d'un magasin de politiques crypté

Avant de créer un magasin de règles chiffré, assurez-vous que les déclarations de politique clés que clé gérée par le client vous utilisez sont définies pour les autorisations vérifiées d'Amazon afin d'utiliser la clé pour le chiffrement/le déchiffrement. Découvrez [the section called "Autoriser l'utilisation de votre AWS KMS clé pour les autorisations vérifiées par Amazon"](#) quelles autorisations sont nécessaires.

En utilisant AWS CLI :

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

À quoi `encrypted.json` ressemble :

```
{
```

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

Assurez-vous de le key remplacer par votre clé gérée par le client ARN et de le remplacer par <ENCRYPTION_CONTEXT_KEY> les <ENCRYPTION_CONTEXT_VALUE> paires encryptionContext clé-valeur souhaitées. encryptionContext peut être complètement omis si aucun ajout de paire clé-valeur n'est souhaité.

Important

N'incluez pas la paire clé-valeur `aws:verifiedpermissions:policy-store-arn` dans votre contexte de chiffrement personnalisé. Ceci est automatiquement ajouté et entraînera des erreurs de validation s'il fait partie des paires clé-valeur du contexte de chiffrement personnalisé que vous avez transmises.

Pour plus d'informations sur la disponibilité APIs des ressources enfants d'un magasin de politiques, consultez la section [Actions](#) du guide de référence de l'API Amazon Verified Permissions.

Note

Si AWS KMS clé gérée par le client les ressources utilisées par vos autorisations vérifiées Amazon sont supprimées, désactivées ou inaccessibles en raison d'une politique de AWS KMS clé incorrecte, le déchiffrement des ressources échouera, ce qui entraînera des décisions d'autorisation périmées. La perte d'accès peut être temporaire (une politique de clé peut être corrigée) ou permanente (une clé supprimée ne peut pas être restaurée) selon les circonstances. Nous vous recommandons de [restreindre l'accès aux](#) opérations critiques, telles que la suppression ou la désactivation de la AWS KMS clé. Nous recommandons également à votre organisation de mettre en place des [procédures AWS d'accès rapides](#) pour garantir l'accès de vos utilisateurs privilégiés AWS dans le cas peu probable où Amazon Verified Permissions serait inaccessible.

Surveillance de l'interaction entre Amazon Verified Permissions et AWS KMS

Vous pouvez surveiller l'utilisation de votre clé gérée par le client canal par Amazon Verified Permissions. AWS CloudTrail Chaque demande envoyée AWS KMS via Verified Permissions inclut le contexte de chiffrement et la clé ARN utilisée (votre clé gérée par le client) dans les paramètres de la demande :

Exemple d'entrée de AWS CloudTrail journal pour GenerateDataKeyWithoutPlaintext :

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "verifiedpermissions.amazonaws.com"
  },
  "eventTime": "2025-09-28T16:51:04Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "verifiedpermissions.amazonaws.com",
  "userAgent": "verifiedpermissions.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwxyz",
    "encryptionContext": {
      "aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012",
      "policy_store_editor": "Janus"
    },
    ...
  },
  ...
}
```

Exemple d'entrée de AWS CloudTrail journal pour Decrypt :

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

```

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

```

Exemple d'entrée de AWS CloudTrail journal pour ReEncrypt :

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "verifiedpermissions.amazonaws.com"
  },
  "eventTime": "2025-09-28T16:51:04Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ReEncrypt",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "verifiedpermissions.amazonaws.com",
  "userAgent": "verifiedpermissions.amazonaws.com",
  "requestParameters": {
    "sourceKeyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwxyz",
    "destinationEncryptionContext": {
      "aws:verifiedpermissions:policy-store-arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
    },
    "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "destinationKeyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-
ijkl-4567-mnopqrstuvwxyz",
  }
}

```

```
    "sourceEncryptionContext": {
      "aws:verifiedpermissions:policy_store_arn":
"arn:aws:verifiedpermissions::111122223333:policy-store/PSt123456789012"
    },
    "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
    ...
  },
  ...
}
```

Notez que les entrées du journal font `invokedBy` référence au principal d'Amazon Verified Permissions et `encryptionContext/sourceEncryptionContext/destinationEncryptionContext` sont incluses dans la carte. `requestParameters`

Exemple d'entrée de AWS CloudTrail journal pour `DescribeKey` :

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "verifiedpermissions.amazonaws.com"
  },
  "eventTime": "2025-09-28T16:51:02Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "verifiedpermissions.amazonaws.com",
  "userAgent": "verifiedpermissions.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/abcdefgh-0123-ijkl-4567-
mnopqrstuvwxyz"
  },
  ...
}
```

Notez que l'entrée du journal inclut une `invokedBy` référence au principal d'Amazon Verified Permissions.

Pour plus d'informations sur les entrées du AWS CloudTrail journal, voir [Comprendre AWS CloudTrail les événements](#) dans le Guide de AWS CloudTrail l'utilisateur.

Limitations

Cette rubrique décrit les limites actuelles des autorisations vérifiées et l'utilisation de clé gérée par le client s pour le chiffrement des ressources.

- Vous ne pouvez pas désactiver le chiffrement pour un magasin de politiques une fois activé
- Après avoir créé un magasin de règles sans chiffrement, vous ne pouvez pas le mettre à jour pour qu'il soit chiffré par un clé gérée par le client
- Une fois que vous avez révoqué l'accès aux autorisations vérifiées à un magasin clé gérée par le client de politiques chiffré existant, il est possible que des décisions d'autorisation périmées soient prises.
- Après avoir créé un magasin de règles avec un clé gérée par le client, vous ne pouvez pas modifier les valeurs de contexte de chiffrement personnalisées ; il s'agit de valeurs statiques définies lors de la création du magasin de règles crypté

Résoudre les problèmes liés aux clés gérées par le client dans les autorisations vérifiées par Amazon

Cette rubrique décrit les erreurs clé gérée par le client associées courantes que vous pouvez rencontrer lors de l'utilisation d'Amazon Verified Permissions et fournit des étapes de dépannage pour les résoudre.

Accès refusé : problème AWS KMS d'autorisation

Erreur : « Le service ou l'appelant n'est pas autorisé à utiliser la AWS KMS clé fournie, car la ressource n'existe pas dans cette région, aucune politique basée sur les ressources n'autorise l'accès ou une politique basée sur les ressources refuse explicitement l'accès »

Cela peut signifier que le service ou l'appelant ne disposent pas des autorisations kms : * d'action requises dans leur IAM politique/politique AWS KMS clé ou que la clé référencée n'existe pas ou n'existe plus.

Résolution des problèmes avec AWS CloudTrail :

- Recherchez des kms . amazonaws . com événements dans AWS CloudTrail
- Recherchez le nom de l'événement de l' AWS KMS opération identifiée comme non autorisée (c'est-à-dire Decrypt ReEncryptGenerateDataKeyWithoutPlaintext,DescribeKey,, etc.)

- Passez en revue les `errorMessage` champs `errorCode` et
- Vérifiez `userIdentity` quel principal a tenté l'opération

Pour résoudre ce problème, accordez à l'utilisateur ou au IAM principal les autorisations d'accès aux AWS KMS opérations appropriées dans sa IAM politique et sa politique AWS KMS clé. Pour de plus amples informations, veuillez consulter [the section called "Politique AWS KMS clé complète"](#).

Exception de validation : configuration AWS KMS des clés

Erreur : « La AWS KMS clé configurée n'a pas de configuration valide »

Cela signifie que la clé référencée ne peut pas être utilisée par le service pour le clé gérée par le client chiffrement en raison de sa configuration actuelle. Les raisons peuvent inclure la désactivation de la clé, le type de clé non pris en charge `EncryptionAlgorithm` ou le type de clé non pris en charge `KeyUsage`.

Exception de limitation : AWS KMS limites de débit

Erreur : « Vous avez dépassé le débit auquel vous pouvez appeler AWS KMS »

Cette erreur signifie que vous avez dépassé la AWS KMS limite d'opérations cryptographiques pour votre clé : <https://docs.aws.amazon.com/kms/latest/developerguide/requests-per-second.html>.

Informations connexes

- [Gestion des magasins de politiques d'autorisations vérifiées](#)
- [AWS KMS Bonnes pratiques](#)
- [AWS KMS Contexte de chiffrement](#)
- [AWS CloudTrail Integration](#)
- [AWS CloudTrail Exemples d'entrées de journal](#)

Gestion des identités et des accès pour Amazon Verified Permissions

Gestion des identités et des accès AWS (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 stylo que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion de l'accès à l'aide de politiques](#)
- [Comment fonctionne Amazon Verified Permissions avec IAM](#)
- [IAM politiques relatives aux autorisations vérifiées](#)
- [Exemples de politiques basées sur l'identité pour Amazon Verified Permissions](#)
- [AWS politiques gérées pour les autorisations vérifiées par Amazon](#)
- [Résolution des problèmes liés à l'identité et à l'accès aux autorisations vérifiées par Amazon](#)

Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir [Résolution des problèmes liés à l'identité et à l'accès aux autorisations vérifiées par Amazon](#))
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir [Comment fonctionne Amazon Verified Permissions avec IAM](#))
- IAM administrateur : rédigez des politiques pour gérer l'accès (voir [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 en tant qu'utilisateur IAM ou en assumant un IAM rôle.

Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations

sur la connexion, consultez [Connexion à votre Compte AWS](#) dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez [AWS la version 4 de Signature pour les demandes d'API](#) dans le guide de IAM l'utilisateur.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches nécessitant des informations d'identification de l'utilisateur root, voir [Tâches nécessitant des informations d'identification de l'utilisateur root](#) dans le Guide de IAM l'utilisateur.

Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de l'annuaire de votre entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées 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. Pour plus d'informations, 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 qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir [Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires](#) dans le Guide de IAM l'utilisateur.

Un [IAM groupe](#) définit un ensemble d'utilisateurs IAM et facilite la gestion des autorisations pour de grands groupes d'utilisateurs. Pour plus d'informations, consultez la section [Cas d'utilisation pour IAM les utilisateurs](#) dans le Guide de IAM l'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 [Créer un rôle pour un fournisseur d'identité tiers \(fédération\)](#) 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 [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 [Utiliser un IAM rôle pour accorder des autorisations aux applications exécutées sur des Amazon EC2 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 de l'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 définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur les documents de politique JSON, consultez la section [Présentation des politiques JSON](#) dans le guide de IAM l'utilisateur.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant 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. Un IAM administrateur crée des IAM politiques et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. IAM les politiques définissent les autorisations quelle que soit la méthode utilisée pour effectuer l'opération.

Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir [Définir des IAM autorisations personnalisées avec des politiques gérées par le client](#) dans le Guide de l'IAM utilisateur.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour savoir comment choisir entre des politiques gérées et des politiques intégrées, 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 exemples incluent les politiques de confiance IAM dans les rôles et les politiques

de Amazon S3 compartiment. 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. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources.

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 (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 AWS WAF, et Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la [présentation de la liste de contrôle d'accès \(ACL\)](#) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité. Pour plus d'informations, consultez la section [Limites des autorisations pour IAM les entités](#) dans le Guide de IAM l'utilisateur.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez [Politiques de contrôle de service](#) dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir [Politiques de contrôle des ressources \(RCPs\)](#) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. 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
ACLs	Non
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Rôles 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, voir [Définir des IAM autorisations personnalisées avec des politiques gérées par le client](#) 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. 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 de Amazon S3 compartiment 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. 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. Intégration d'actions dans une politique 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. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, 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 caractéristiques ARNs, 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 Non
politique spécifiques au service

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` indique à quel moment les instructions s'exécutent en fonction de critères définis. 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. Pour voir toutes les clés de condition AWS globales, voir les [clés contextuelles de condition AWS globales](#) dans le guide de IAM l'utilisateur.

ACLs dans Autorisations vérifiées

Supports ACLs	Non
---------------	-----

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs 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

Prend en charge ABAC (étiquettes dans les politiques)	Oui
---	-----

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs appelés balises. Vous pouvez associer des balises aux IAM entités et aux AWS ressources, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

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 ABAC, voir [Définir les autorisations avec l'autorisation ABAC](#) dans le guide de l'IAM 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
---	-----

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. 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 la section [Informations d'identification de sécurité temporaires dans IAM](#) et [Services AWS compatibles IAM](#) dans le Guide de l'IAM utilisateur.

Autorisations principales interservices pour les autorisations vérifiées

Prend en charge les autorisations de principal Oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section [Sessions de transmission 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 la section [Créer un rôle pour déléguer des autorisations à un Service AWS](#) dans le guide de IAM l'utilisateur.

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 fichier 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.

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 autorisations vérifiées APIs ou pour que AWS Management Console les utilisateurs soient autorisés à gérer les politiques de 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
CreateIdentitySource	Action visant à créer une nouvelle source d'identité.
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.
CreatePolicyStore	Action visant à créer un nouveau magasin de politiques.

Action	Description
CreatePolicyTemplate	Action visant à créer un nouveau modèle de politique.
DeleteIdentitySource	Action visant à supprimer une source d'identité.
DeletePolicy	Action visant à supprimer une politique d'un magasin de politiques.
DeletePolicyStore	Action visant à supprimer un magasin de politiques.
DeletePolicyTemplate	Action visant à supprimer un modèle de politique.
GetIdentitySource	Action visant à obtenir une source d'identité.
GetPolicy	Action permettant de récupérer des informations relatives à une politique spécifiée.
GetPolicyStore	Action permettant de récupérer des informations sur un magasin de politiques spécifié.
GetPolicyTemplate	Action visant à obtenir un modèle de politique.
GetSchema	Action pour obtenir un schéma.
IsAuthorized	Action permettant d'obtenir une réponse d'autorisation en fonction des paramètres décrits dans la demande d'autorisation .
IsAuthorizedWithToken	Action permettant d'obtenir une réponse d'autorisation basée sur les paramètres décrits dans la demande d'autorisation lorsque le principal provient d'un jeton d'identité.
ListIdentitySources	Action permettant de répertorier toutes les sources d'identité dans le Compte AWS.

Action	Description
ListPolicies	Action permettant de répertorier toutes les politiques d'un magasin de politiques.
ListPolicyStores	Action visant à répertorier tous les magasins de politiques dans le Compte AWS.
ListPolicyTemplates	Action visant à répertorier tous les modèles de politiques dans le Compte AWS.
ListTagsForResource	Action permettant de répertorier tous les tags d'une ressource.
PutSchema	Action permettant d'ajouter un schéma à un magasin de politiques.
TagResource	Action permettant d'ajouter un tag à une ressource.
UpdateIdentitySource	Action visant à mettre à jour une source d'identité.
UpdatePolicy	Action visant à mettre à jour une politique dans un magasin de politiques.
UpdatePolicyStore	Action visant à mettre à jour un magasin de politiques.
UpdatePolicyTemplate	Action visant à mettre à jour un modèle de politique.
UntagResource	Action permettant de supprimer un tag d'une ressource.

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

JSON

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

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 l'API AWS IAM. Un administrateur IAM doit créer des politiques IAM 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'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par Verified Permissions, y compris le format des ARNs pour chacun des types de ressources, consultez la section [Actions, ressources et clés de condition pour les autorisations vérifiées d'Amazon](#) dans la référence d'autorisation de service.

Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la console d'autorisations vérifiées](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)

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 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 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 [Valider les politiques avec 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 la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez la section [Accès sécurisé aux API avec 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 politique 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 politique.

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:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

AWS politiques gérées pour les autorisations vérifiées par Amazon

Pour ajouter des autorisations aux utilisateurs, aux groupes et aux rôles, il est plus facile d'utiliser des politiques AWS gérées que de les rédiger vous-même. Il faut du temps et de l'expertise pour [créer des politiques gérées par les IAM clients](#) qui fournissent à votre équipe uniquement les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent des cas d'utilisation courants et sont disponibles dans votre Compte AWS. Pour plus d'informations sur les politiques AWS gérées, voir les [politiques AWS gérées](#) dans le Guide de IAM l'utilisateur.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services ajoutent occasionnellement des autorisations à une politique gérée par AWS pour prendre en charge de nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la

politique est attachée. Les services sont très susceptibles de mettre à jour une politique gérée par AWS quand une nouvelle fonctionnalité est lancée ou quand de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'une politique AWS gérée. Les mises à jour des politiques n'endommageront donc pas vos autorisations existantes.

En outre, AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique `ReadOnlyAccess` AWS gérée fournit un accès en lecture seule à tous les AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir une liste et une description des politiques relatives aux fonctions de travail, voir [les politiques AWS gérées pour les fonctions de travail](#) dans le Guide de IAM l'utilisateur.

AWS politique gérée : `AmazonVerifiedPermissionsFullAccess`

La politique `AmazonVerifiedPermissionsFullAccess` gérée accorde un accès complet aux autorisations vérifiées. Pour travailler avec des sources d'identité Amazon Cognito basées, vous devez joindre une politique distincte, telle que la [AmazonCognitoReadOnly](#) politique.

JSON

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

```
    ],
    "Resource": [
      "arn:aws:verifiedpermissions::*:policy-store/*"
    ]
  }
]
```

AWS politique gérée : AmazonVerifiedPermissionsReadOnlyAccess

La politique AmazonVerifiedPermissionsReadOnlyAccess gérée accorde un accès en lecture seule aux autorisations vérifiées.

Cette politique donne accès à toutes les opérations de lecture d'Amazon Verified Permissions, y compris la requête d'autorisation APIs IsAuthorized etIsAuthorizedWithToken.

Note

L'accès à BatchIsAuthorized et BatchIsAuthorizedWithToken est accordé automatiquement lorsque l'accès est accordé à IsAuthorized etIsAuthorizedWithToken, respectivement.

JSON

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

```

    "verifiedpermissions:GetIdentitySource",
    "verifiedpermissions:GetPolicy",
    "verifiedpermissions:GetPolicyStore",
    "verifiedpermissions:GetPolicyTemplate",
    "verifiedpermissions:GetSchema",
    "verifiedpermissions:IsAuthorized",
    "verifiedpermissions:IsAuthorizedWithToken",
    "verifiedpermissions:ListIdentitySources",
    "verifiedpermissions:ListPolicies",
    "verifiedpermissions:ListPolicyTemplates"
  ],
  "Resource": [
    "arn:aws:verifiedpermissions::*:policy-store/*"
  ]
}
]
}

```

Mises à jour des autorisations vérifiées apportées aux politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées pour les autorisations vérifiées depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au fil RSS sur la page d'historique du document d'autorisations vérifiées.

Modifier	Description	Date
AmazonVerifiedPermissionsFullAccess : nouvelle politique	Permissions vérifiées a ajouté une nouvelle politique permettant un accès complet aux autorisations vérifiées.	11 octobre 2024
AmazonVerifiedPermissionsReadOnlyAccess : nouvelle politique	Verified Permissions a ajouté une nouvelle politique pour autoriser l'accès à toutes les opérations de lecture d'Amazon Verified	11 octobre 2024

Modifier	Description	Date
	Permissions, y compris la requête d'autorisation APIs <code>IsAuthorized</code> et <code>IsAuthorizedWithToken</code> .	
Les autorisations vérifiées ont commencé à suivre les modifications	Verified Permissions a commencé à suivre les modifications apportées AWS à ses politiques gérées.	11 octobre 2024

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'action `iam:PassRole`, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle aux autorisations vérifiées.

Certains 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, vous devez 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 n'est pas autorisée à transmettre 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 (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si les autorisations vérifiées prennent en charge ces fonctionnalités, consultez [Comment 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 l'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. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez [AWS la documentation de sécurité](#).

Résilience dans les autorisations vérifiées par Amazon

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. 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 individu Région 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 en charge aucune réplication entre régions.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section [Infrastructure AWS globale](#).

Surveillance des appels d'API Amazon Verified Permissions

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'Amazon Verified Permissions et de vos autres AWS solutions. AWS fournit les outils suivants pour surveiller les autorisations vérifiées, signaler tout problème et prendre des mesures automatiques le cas échéant :

- AWS CloudTrail capture les appels d'API et les événements associés effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux dans un Amazon S3 compartiment que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, l'adresse IP source à partir de laquelle les appels ont été effectués et la date des appels. Pour plus d'informations, consultez le [Guide de l'utilisateur AWS CloudTrail](#).

Pour plus d'informations sur la surveillance des autorisations vérifiées avec CloudTrail, consultez [Journalisation des appels d'API Amazon Verified Permissions à l'aide de 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 Amazon S3 bucket, y compris des événements pour les autorisations vérifiées. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements relatifs aux actions de gestion les plus récents dans la CloudTrail console dans Historique des événements, mais pas les événements relatifs aux appels d'API tels que `isAuthorized`. 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 site Compte AWS, y compris des événements pour les autorisations vérifiées, créez une trace. Un suivi permet CloudTrail de transférer des fichiers journaux dans un Amazon S3 bucket. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et envoie les fichiers journaux dans le Amazon S3 compartiment que vous spécifiez. 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 plus d'informations, 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 faite avec les informations d'identification de l'utilisateur root ou Gestion des identités et des accès AWS (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'[élément userIdentity CloudTrail](#) .

Les événements liés aux données [IsAuthorizedWithToken](#) sont similaires [IsAuthorized](#) et 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 Amazon S3 compartiment 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.

Pour les appels d'API d'autorisation, les éléments de réponse, tels que la décision, sont inclus sous `additionalEventData` plutôt que `responseElements`.

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": "PSEXAMPLEEabcdefg111111"
```

```

    },
    "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-TOKEN11111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
      "entityType": "PhotoApp::Role",
      "entityId": "PhotoJudge"
    },
  },
  "resource": {
    "entityType": "PhotoApp::Application",
```

```

    "entityId": "PhotoApp"
  },
  "lastUpdatedDate": "2023-05-22T07:42:30.70852Z",
  "createdDate": "2023-05-22T07:42:30.70852Z"
},
"requestID": "93ffa151-3841-4960-9af6-30a7f817ef93",
"eventID": "30ab405f-3dff-43ff-8af9-f513829e8bde",
"readOnly": false,
"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

GetPolicy

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:29Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetPolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111"
  }
},

```

```

"responseElements": null,
"requestID": "23022a9e-2f5c-4dac-b653-59e6987f2fac",
"eventID": "9b4d5037-bafa-4d57-b197-f46af83fc684",
"readOnly": true,
"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

CreateIdentitySource

```

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

```

```

    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "principalEntityType": "User"
  },
  "responseElements": {
    "createdDate": "2023-07-14T15:05:01.599534Z",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-07-14T15:05:01.599534Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "requestID": "afcc1e67-d5a4-4a9b-a74c-cdc2f719391c",
  "eventID": "f13a41dc-4496-4517-aeb8-a389eb379860",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}

```

GetIdentitySource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:31Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",

```

```
"requestParameters": {
  "identitySourceId": "ISEXAMPLEEabcdefg111111",
  "policyStoreId": "PSEXAMPLEEabcdefg111111"
},
"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/
PSEXAMPLEEabcdefg111111"
  }
],
"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": "PSEXAMPLEEabcdefg111111"
  },
  "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 CloudFormation qui fournit et configure ces ressources pour vous.

Lorsque vous l'utilisez 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 une erreur CloudFormation concernant Amazon Cognito Identity, par exemple `Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient`, nous vous recommandons de créer le groupe Amazon Cognito d'utilisateurs et le client dans la zone géographique la plus proche Région AWS où 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 CloudFormation modèles vérifiés

Pour fournir et configurer des ressources pour les autorisations vérifiées et les services associés, vous devez comprendre les [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 CloudFormation piles. Si vous n'êtes pas familiarisé avec JSON ou YAML, vous pouvez utiliser CloudFormation Designer pour vous aider à démarrer avec les CloudFormation modèles. Pour plus d'informations, voir [Qu'est-ce que 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, de modèles de politiques et d'alias de magasins de politiques dans 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. CloudFormation Des constructions, ou des composants cloud réutilisables, peuvent être utilisés pour créer des CloudFormation modèles. Ces modèles peuvent ensuite être utilisés pour déployer votre infrastructure cloud.

Pour en savoir plus et télécharger le AWS 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 CloudFormation

Pour en savoir plus CloudFormation, consultez les ressources suivantes :

- [AWS CloudFormation](#)
- [AWS CloudFormation Guide de l'utilisateur](#)
- [CloudFormation API Reference](#)
- [AWS CloudFormation Guide de l'utilisateur de l'interface de ligne de commande](#)

Accédez aux autorisations Amazon Verified à l'aide de AWS PrivateLink

Vous pouvez l'utiliser AWS PrivateLink pour créer une connexion privée entre votre VPC et Amazon Verified Permissions. Vous pouvez accéder aux autorisations vérifiées comme si elles se trouvaient dans votre VPC, sans utiliser de passerelle Internet, de périphérique NAT, de connexion VPN ou Direct Connect de connexion. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour accéder aux autorisations vérifiées.

Vous établissez cette connexion privée en créant un point de terminaison d'interface optimisé par AWS PrivateLink. Nous créons une interface réseau de 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 les demandeurs qui servent de point d'entrée pour le trafic destiné aux autorisations vérifiées.

Pour plus d'informations, consultez [Accès aux Services AWS via AWS PrivateLink](#) dans le Guide AWS PrivateLink .

Considérations relatives aux autorisations vérifiées

Avant de configurer un point de terminaison d'interface pour les autorisations vérifiées, consultez les [considérations](#) du AWS PrivateLink guide.

Verified Permissions permet d'appeler toutes ses actions d'API via le point de terminaison de l'interface.

Les politiques de point de terminaison VPC ne sont pas prises en charge pour les autorisations vérifiées. Par défaut, l'accès complet aux autorisations vérifiées est autorisé via le point de terminaison de l'interface. Vous pouvez également associer un groupe de sécurité aux interfaces réseau du point de terminaison pour contrôler le trafic vers les autorisations vérifiées via le point de terminaison de l'interface.

Création d'un point de terminaison d'interface pour les autorisations vérifiées

Vous pouvez créer un point de terminaison d'interface pour les autorisations vérifiées à l'aide de la console Amazon VPC ou du AWS Command Line Interface (AWS CLI). Pour plus d'informations, consultez [Création d'un point de terminaison d'interface](#) dans le Guide AWS PrivateLink .

Créez un point de terminaison d'interface pour les autorisations vérifiées en utilisant le nom de service suivant :

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

Si vous activez le DNS privé pour le point de terminaison de l'interface, vous pouvez envoyer des demandes d'API à Verified Permissions en utilisant son nom DNS régional par défaut. Par exemple, `verifiedpermissions.us-east-1.amazonaws.com`.

Création d'une politique de point de terminaison pour votre point de terminaison d'interface

Une politique de point de terminaison est une ressource IAM que vous pouvez attacher à votre point de terminaison d'interface. La politique de point de terminaison par défaut permet un accès complet aux autorisations vérifiées via le point de terminaison de l'interface. Pour contrôler l'accès autorisé aux autorisations vérifiées depuis votre VPC, associez une politique de point de terminaison personnalisée au point de terminaison de l'interface.

Une politique de point de terminaison spécifie les informations suivantes :

- Les principaux qui peuvent effectuer des actions (Comptes AWS, utilisateurs IAM et rôles IAM).
- Les actions qui peuvent être effectuées.
- La ressource sur laquelle les actions peuvent être effectuées.

Pour plus d'informations, consultez [Contrôle de l'accès aux services à l'aide de politiques de point de terminaison](#) dans le Guide AWS PrivateLink .

Exemple : politique de point de terminaison VPC pour les actions d'autorisations vérifiées

Voici un exemple de politique de point de terminaison personnalisée. Lorsque vous attachez cette politique au point de terminaison de votre interface, elle accorde l'accès aux actions d'autorisations vérifiées répertoriées pour tous les principaux sur toutes les ressources.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
```

```
    "Action": [  
      "verifiedpermissions:IsAuthorized",  
      "verifiedpermissions:IsAuthorizedWithToken",  
      "verifiedpermissions:GetPolicy"  
    ],  
    "Resource": "*"    
  }  
]  
}
```

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 Region-specific. Vous pouvez demander des augmentations de certains quotas, tandis que 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](#)
- [Quotas pour les hiérarchies](#)
- [Quotas d'opérations par seconde](#)

Quotas de ressources

Nom	Par défaut	Ajusté	Description
Boutiques Policy par région et par compte	Chaque région prise en charge : 30 000	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 politiques 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	Ajusté	Description
Alias de magasin de politiques par magasin de politiques	10	Oui	Nombre maximal d'alias de magasin de politiques que vous pouvez associer à un seul magasin de politiques.
Taille de la demande d'autorisation ¹	1 Mo	Non	Taille maximale d'une demande d'autorisation.
Taille de la politique	10 000 octets	Oui	La taille maximale d'une police individuelle.
Taille du schéma	100,000 bytes	Oui	Taille maximale du schéma d'un magasin de politiques.
Espaces de noms par schéma de magasin de politiques	100	Oui	Nombre maximal d'espaces de noms que vous pouvez définir dans un schéma de magasin de politiques.
Taille de la politique par ressource	200 000 octets ²	Oui	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 [IsAuthorized](#) et [IsAuthorizedWithToken](#).

² La limite par défaut pour la taille totale de toutes les politiques définies pour une seule ressource est de 200 000 octets. De même, la taille totale de toutes les politiques, dont la portée laisse la ressource indéfinie, s'appliquant ainsi à toutes les ressources, est limitée par défaut à 200 000 octets. Notez que 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. Cette limite peut être augmentée, à condition que la conception de votre politique réponde à certaines contraintes. Si vous avez besoin d'explorer cette option, [contactez Support](#).

Template-linked exemple de taille de politique

Vous pouvez déterminer comment les politiques liées à un modèle contribuent à la taille de la politique par quota de ressource en faisant la somme de la longueur du principal et de la ressource. Si le principal ou la ressource n'est pas spécifié, la longueur de cette pièce est de 0. Si aucune ressource n'est spécifiée, sa taille est prise en compte dans le quota de "unspecified" ressources. La taille du corps du modèle lui-même n'a aucune incidence sur la taille de la politique.

Examinons le modèle suivant :

```
@id("template1")
permit (
  principal in ?principal,
  action in [Action::"view", Action::"comment"],
  resource in ?resource
)
unless {
  resource.tag == "private"
};
```

Créons les politiques suivantes à partir de ce modèle :

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

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

TemplateLinkedPolicy {
  policyId: "policy3",
  templateId: "template1",
```

```
principal: User::"jane",
resource: Photo::"car.jpg"

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

Calculons maintenant la taille de ces politiques en comptant les caractères dans le `principal` et `resource` pour chacune d'elles. Chaque caractère compte pour 1 octet.

La taille de `policy1` serait la longueur du principal `User::"alice"` (13) plus la longueur de la ressource `Photo::"car.jpg"` (16). En les additionnant, nous avons $13 + 16 = 29$ octets.

La taille de `policy2` serait la longueur du principal `User::"bob"` (11) plus la longueur de la ressource `Photo::"boat.jpg"` (17). En les additionnant, nous avons $11 + 17 = 28$ octets.

La taille de `policy3` serait la longueur du principal `User::"jane"` (12) plus la longueur de la ressource `Photo::"car.jpg"` (16). En les additionnant, nous avons $12 + 16 = 28$ octets.

La taille de `policy4` serait la longueur du principal `User::"jane"` (12) plus la longueur de la ressource (0). En les additionnant, nous avons $12 + 0 = 12$ octets.

Comme `policy2` il s'agit de la seule politique qui fait référence à la ressource `Photo::"boat.jpg"`, la taille totale de la ressource est de 28 octets.

Étant donné que `policy1` et `policy3` deux font référence à la ressource `Photo::"car.jpg"`, la taille totale de la ressource est de $29 + 28 = 57$ octets.

Comme `policy4` il s'agit de la seule politique qui fait référence à la "unspecified" ressource, la taille totale de la ressource est de 12 octets.

Quotas pour les hiérarchies

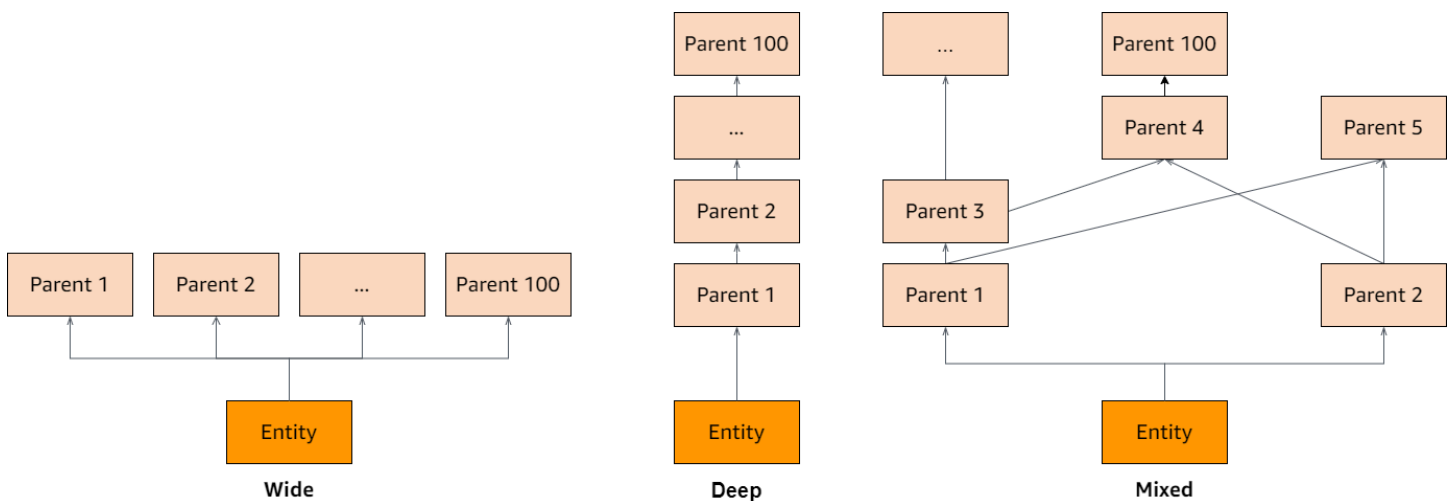
Note

Les quotas suivants sont agrégés, c'est-à-dire qu'ils sont additionnés. Le nombre maximum de parents transitifs pour le groupe est indiqué. Par exemple, si la limite de parents transitifs par directeur est de 100, cela signifie qu'il peut y avoir 100 parents de directeurs et aucun

parent à la fois pour les actions et les ressources, ou toute combinaison de parents totalisant 100 parents au total.

Nom	Par défaut	Ajusté	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.
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	Ajusté	Description
BatchGetPolicy demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de BatchGetPolicy demandes par seconde par magasin de politiques.
BatchIsAuthorized demandes par seconde, par région et par magasin de politiques	Chaque Région prise en charge : 30	Oui	Le nombre maximal de BatchIsAuthorized demandes par seconde par magasin de politiques.
BatchIsAuthorizedWithToken demandes par seconde, par région et par magasin de politiques	Chaque Région prise en charge : 30	Oui	Le nombre maximal de BatchIsAuthorizedWithToken demandes par seconde par magasin de politiques.
CreateIdentitySource demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 1	Oui	Le nombre maximal de CreateIdentitySource demandes par seconde par magasin de politiques.
CreatePolicy demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de CreatePolicy demandes par seconde par magasin de politiques.

Nom	Par défaut	Ajusté	Description
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 magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de CreatePolicyTemplate demandes par seconde par magasin de politiques.
DeleteIdentitySource demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 1	Oui	Le nombre maximal de DeleteIdentitySource demandes par seconde par magasin de politiques.
DeletePolicy demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de DeletePolicy demandes par seconde par magasin de politiques.
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 magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de DeletePolicyTemplate demandes par seconde par magasin de politiques.
GetIdentitySource demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de GetIdentitySource demandes par seconde par magasin de politiques.

Nom	Par défaut	Ajuste	Description
GetPolicy demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de GetPolicy demandes par seconde par magasin de politiques.
GetPolicyStore demandes par seconde, par région et par compte	Par région prise en charge : 10	Oui	Le nombre maximum de GetPolicyStore demandes par seconde.
GetPolicyTemplate demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de GetPolicyTemplate demandes par seconde par magasin de politiques.
GetSchema demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de GetSchema demandes par seconde par magasin de politiques.
IsAuthorized demandes par seconde, par région et par magasin de politiques	Chaque région prise en charge : 200	Oui	Le nombre maximal de IsAuthorized demandes par seconde par magasin de politiques.
IsAuthorizedWithToken demandes par seconde, par région et par magasin de politiques	Chaque région prise en charge : 200	Oui	Le nombre maximal de IsAuthorizedWithToken demandes par seconde par magasin de politiques.
ListIdentitySources demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de ListIdentitySources demandes par seconde par magasin de politiques.

Nom	Par défaut	Ajuste	Description
ListPolicies demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de ListPolicies demandes par seconde par magasin de politiques.
ListPolicyStores demandes par seconde, par région et par compte	Par région prise en charge : 10	Oui	Le nombre maximum de ListPolicyStores demandes par seconde.
ListPolicyTemplates demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de ListPolicyTemplates demandes par seconde par magasin de politiques.
PutSchema demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de PutSchema demandes par seconde par magasin de politiques.
UpdateIdentitySource demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 1	Oui	Le nombre maximal de UpdateIdentitySource demandes par seconde par magasin de politiques.
UpdatePolicy demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de UpdatePolicy demandes par seconde par magasin de politiques.
UpdatePolicyStore demandes par seconde, par région et par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de UpdatePolicyStore demandes par seconde.

Nom	Par défaut	Ajuste	Description
UpdatePolicyTemplate demandes par seconde, par région et par magasin de politiques	Par région prise en charge : 10	Oui	Le nombre maximal de UpdatePolicyTemplate demandes par seconde par magasin de politiques.

Termes et concepts linguistiques relatifs aux autorisations vérifiées par Amazon et à la politique de Cedar

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](#)
- [Alias du magasin Policy](#)
- [Nom de la politique](#)
- [Nom du modèle de politique](#)
- [Politiques satisfaites](#)
- [Différences entre les autorisations vérifiées d'Amazon et le langage politique de 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 IDs les 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.

Alias du magasin Policy

Un alias de magasin de politiques est un nom convivial pour un magasin de politiques. Vous pouvez utiliser un alias de magasin de politiques pour identifier un magasin de politiques dans toute opération d'autorisations vérifiées qui accepte un `policyStoreId` paramètre. Les alias du Policy Store sont des AWS ressources indépendantes dotées de leurs propres ARNs ressources. Chaque alias est associé à un magasin de politiques à la fois, et plusieurs alias peuvent être associés au même magasin de politiques. Pour de plus amples informations, veuillez consulter [Alias de la boutique Amazon Verified Permissions Policy](#).

Nom de la politique

Un nom de stratégie est un nom convivial facultatif pour une stratégie. Les noms des politiques doivent être uniques pour toutes les politiques du magasin de politiques et être préfixés par `name/`. Vous pouvez utiliser un nom de stratégie à la place de l'ID de stratégie dans les opérations du plan de contrôle qui acceptent un `policyId` paramètre. Les noms peuvent être définis lors de la création ou de la mise à jour d'une politique. Uniquement `GetPolicy` et `ListPolicies` renvoient le nom dans la sortie.

Nom du modèle de politique

Un nom de modèle de stratégie est un nom convivial facultatif pour un modèle de stratégie. Les noms des modèles de stratégie doivent être uniques pour tous les modèles de politiques du magasin de politiques et être préfixés par `name/`. Vous pouvez utiliser un nom de modèle de stratégie à la place de l'ID du modèle de stratégie dans les opérations du plan de contrôle qui acceptent un `policyTemplateId` paramètre. Les noms peuvent être définis lors de la création ou de la mise à jour d'un modèle de politique. Uniquement `GetPolicyTemplate` et `ListPolicyTemplates` renvoient le nom dans la sortie.

Politiques satisfaites

Les politiques satisfaisantes sont celles qui correspondent aux paramètres de la [demande d'autorisation](#).

Différences entre les autorisations vérifiées d'Amazon et le langage politique de 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 identifie ces différences.

Note

Depuis mai 2026, Verified Permissions s'aligne sur Cedar et prend désormais en charge plusieurs espaces de noms.

Support des modèles de politiques

Verified Permissions et Cedar autorisent les espaces réservés uniquement aux `principal` et `resource`. Cependant, les autorisations vérifiées nécessitent également que ni les `principal` et ni `resource` soient pas illimitées.

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

```
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` sont soumis à 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 ou les espaces de noms.

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 à laquelle vous passez en tant que paramètre des `IsAuthorizedWithToken` opérations `IsAuthorized` or.

Formatage des entités

Le formatage JSON des entités dans Verified Permissions utilisant le `entityList` paramètre 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
<code>uid</code>	<code>Identifier</code>
<code>type</code>	<code>EntityType</code>
<code>id</code>	<code>EntityId</code>
<code>attrs</code>	<code>Attributes</code>
<code>parents</code>	<code>Parents</code>

Exemple- Listes

Les exemples suivants montrent comment une liste d'entités est exprimée dans Cedar et Verified Permissions, respectivement.

Cedar

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

Verified Permissions

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

Exemple- Évaluation des politiques

Les exemples suivants montrent comment les entités sont formatées pour évaluer une politique dans une demande d'autorisation dans Cedar et Verified Permissions, respectivement.

Cedar

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

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

Verified Permissions

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

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

Limites de 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
Description de la politique en ligne	150	Non applicable au cèdre
Taille du modèle de politique	10 000	Aucune
Taille du schéma	100 000	Aucune
Type d'entité	200	Aucune
ID de stratégie	64	Aucune
ID du modèle de politique	64	Aucune
ID de l'entité	200	Aucune
ID du magasin Policy	64	Non applicable 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 est comptée une seule fois, plus la taille de chaque ensemble de paramètres utilisé pour instancier chaque politique liée au modèle.

FAQ sur la mise à niveau des autorisations vérifiées par Amazon vers Cedar 4

Amazon Verified Permissions est en train de mettre à niveau la version de Cedar qu'il utilise de la version 2 à la version 4. Cedar est le langage open source que vous utilisez pour écrire les politiques, les modèles de politiques et les schémas dans vos magasins de politiques. Grâce à la prise en charge des autorisations vérifiées par Cedar 4, vous pouvez utiliser de nouvelles fonctionnalités telles que l'isopérateur et les balises d'entité pour rédiger des politiques plus expressives.

Amazon Verified Permissions met automatiquement à niveau les boutiques politiques vers Cedar 4. Cependant, certaines politiques, schémas et demandes d'autorisation rédigés pour Cedar 2 sont incompatibles avec Cedar 4. Si tel est le cas pour votre magasin de politiques, nous ne le mettrons pas à niveau automatiquement. Vous devrez peut-être apporter des modifications à vos politiques, modèles de politiques, schémas ou code d'application avant de passer à Cedar 4.

Rubriques

- [Pourquoi est-ce que je reçois un message d'erreur indiquant que Cedar 2 n'est plus pris en charge lorsque j'appelle Amazon Verified Permissions ?](#)
- [Pourquoi certaines politiques, certains modèles de politiques et certains schémas ne sont-ils pas compatibles avec Cedar 4 ?](#)
- [Comment savoir si ma boutique utilise Cedar 2 ou Cedar 4 ?](#)
- [Comment passer à Cedar 4 ?](#)
- [Puis-je rétrograder ma boutique en ligne de Cedar 4 à Cedar 2 ?](#)
- [Pourquoi est-ce que je reçois un message d'erreur indiquant que mon magasin de politiques est configuré pour Cedar 2 ?](#)
- [Comment rendre mon schéma compatible avec Cedar 4 ?](#)
- [Comment puis-je rendre mes politiques et mes modèles compatibles avec Cedar 4 ?](#)

Pourquoi est-ce que je reçois un message d'erreur indiquant que Cedar 2 n'est plus pris en charge lorsque j'appelle Amazon Verified Permissions ?

À partir d'avril 2026, les API d'autorisation (`IsAuthorizedBatchIsAuthorized`, `IsAuthorizedWithToken` et `BatchIsAuthorizedWithToken`) seront désactivées pour les magasins de politiques qui utilisent Cedar 2. Dans ce cas, le message d'erreur suivant est renvoyé :

```
Your policy store uses Cedar 2.x, which is no longer supported. See https://docs.aws.amazon.com/verifiedpermissions/latest/userguide/cedar4-faq.html or contact AWS Support for more information.
```

Si vous recevez ce message d'erreur lorsque vous appelez Verified Permissions, cela signifie que votre magasin de politiques est incompatible avec Cedar 4 et n'a donc pas pu être migré vers la nouvelle version. Pour continuer à utiliser les autorisations vérifiées d'Amazon, créez un nouveau magasin de politiques, qui utilisera Cedar 4, ou contactez Support.

Pourquoi certaines politiques, certains modèles de politiques et certains schémas ne sont-ils pas compatibles avec Cedar 4 ?

L'équipe de Cedar a apporté plusieurs modifications incompatibles avec les versions antérieures depuis Cedar 2, afin de corriger des bogues et de simplifier le langage. Voici les modifications concernées :

- modifications de syntaxe pour les politiques, les modèles de politiques et les schémas
- un validateur de politique plus précis, qui détecte davantage d'erreurs
- modifications du comportement de fonctions intégrées telles que `isInRange`

[Pour une liste complète des modifications incompatibles avec les versions antérieures, recherchez les éléments marqués d'un \(* \) dans le journal des modifications de Cedar.](#)

Comment savoir si ma boutique utilise Cedar 2 ou Cedar 4 ?

Vous pouvez vérifier la version de Cedar utilisée par votre boutique Policy à l'aide de la console Amazon Verified Permissions ou à l'aide de l'opération `GetPolicyStore`.

Note

Toutes les boutiques Policy de la même Compte AWS région utilisent la même version de Cedar.

Console

Pour vérifier la version Cedar d'un magasin de politiques (console)

1. Connectez-vous à la console Amazon Verified Permissions AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/verifiedpermissions/>.
2. Dans le volet de navigation, choisissez Policy Stores, puis choisissez le Policy Store que vous souhaitez vérifier.
3. Choisissez Settings (Paramètres) dans le volet de navigation.
4. Dans la zone Détails, recherchez le champ de version de Cedar.

Le champ indique CEDAR_2 si votre magasin de polices utilise Cedar 2 et CEDAR_4 s'il utilise Cedar 4.

CLI

Pour vérifier la version Cedar d'un magasin de politiques (AWS CLI)

1. Installez et configurez le AWS Command Line Interface (AWS CLI), si ce n'est pas déjà fait. Pour de plus amples informations, consultez [Installation ou mise à jour de la version la plus récente de l' AWS CLI](#).
2. Utilisez la commande `get-policy-store`. Dans l'exemple suivant, remplacez *policy-store-id* par l'identifiant de votre magasin de polices :

```
aws verifiedpermissions get-policy-store \  
--policy-store-id policy-store-id
```

Le `cedarVersion` champ de sortie indique la version de Cedar utilisée par le Policy Store. Par exemple :

```
{
```

```
"policyStoreId": "ABCDEFG12345678abcdefg",
"arn": "arn:aws:verifiedpermissions::111122223333:policy-store/
ABCDEFG12345678abcdefg",
"validationSettings": {
  "mode": "STRICT"
},
"createdDate": "2025-06-03T13:09:47.752255+00:00",
"lastUpdatedDate": "2025-06-03T13:09:47.752255+00:00",
"deletionProtection": "ENABLED",
"cedarVersion": "CEDAR_2"
}
```

Le champ indique CEDAR_2 si votre magasin de polices utilise Cedar 2 et CEDAR_4 s'il utilise Cedar 4.

Comment passer à Cedar 4 ?

Amazon Verified Permissions a déjà fait passer la plupart de ses clients à Cedar 4. Si vous n'avez jamais créé de magasin de politiques, tout nouveau magasin de politiques que vous créerez utilisera Cedar 4. Si vous êtes déjà client, nous vous avons probablement déjà fait passer à Cedar 4. Consultez [Comment savoir si ma boutique utilise Cedar 2 ou Cedar 4 ?](#) la version de Cedar utilisée par vos magasins d'assurance.

Si vous n'avez pas été mis à niveau, Verified Permissions a détecté une politique, un modèle de politique, un schéma ou une demande d'autorisation incompatible avec Cedar 4 dans l'un de vos magasins de politiques. Nous vous enverrons une notification par e-mail décrivant les ressources incompatibles plus tard en 2025. Pour effectuer la mise à niveau plus tôt, ouvrez un dossier auprès de Support.

Important

Tous les magasins de polices d'assurance de la même version Compte AWS utilisent la même version de Cedar. Si l'une des boutiques de polices de votre compte est incompatible avec Cedar 4, vous ne pouvez pas utiliser Cedar 4 dans aucune des boutiques de polices de ce compte.

Puis-je rétrograder ma boutique en ligne de Cedar 4 à Cedar 2 ?

Non Si vous rencontrez des problèmes après la mise à niveau de votre boutique de polices vers Cedar 4, ouvrez un dossier auprès de Support.

Pourquoi est-ce que je reçois un message d'erreur indiquant que mon magasin de politiques est configuré pour Cedar 2 ?

Certaines fonctionnalités d'Amazon Verified Permissions reposent sur les nouvelles fonctionnalités de Cedar 4. Si votre magasin de politiques n'utilise pas Cedar 4, vous ne pouvez pas utiliser les champs d'API suivants :

- Dans les `BatchIsAuthorizedWithToken` opérations `IsAuthorizedBatchIsAuthorized`, `IsAuthorizedWithToken` et :
 - `datetime`, `decimal` ou `duration` des valeurs dans les `context` champs `attributes` ou

Vous ne pouvez pas utiliser de syntaxe ou de types de données dans les politiques, les modèles de politiques ou les schémas introduits après Cedar 2 tant que votre magasin de politiques n'est pas mis à niveau.

Comment rendre mon schéma compatible avec Cedar 4 ?

La console Verified Permissions peut résoudre automatiquement certains problèmes de compatibilité dans votre schéma. Si votre schéma ne peut pas être corrigé automatiquement, la console affiche une liste d'erreurs à corriger manuellement.

Important

L'éditeur de code de la console Amazon Verified Permissions affiche toujours les erreurs et les avertissements provenant de Cedar 4, même si votre boutique de polices utilise Cedar 2. Vous pouvez continuer à effectuer des mises à jour de schéma qui ne sont pas compatibles avec Cedar 4 à l'aide du bouton Enregistrer les modifications ou de l'API Verified Permissions.

Pour corriger un schéma à l'aide de la console

1. Connectez-vous à la console Amazon Verified Permissions AWS Management Console et ouvrez-la à l'adresse [verifiedpermissions](#).
2. Dans le volet de navigation, choisissez Policy Stores, puis choisissez le Policy Store que vous souhaitez vérifier.
3. Choisissez Schema dans le volet de navigation.
4. Si votre schéma peut être corrigé automatiquement, vous verrez une bannière indiquant « Cliquez sur « Corriger » pour prévisualiser une version compatible ». Sélectionnez Corriger.
5. Passez en revue les modifications apportées à votre schéma, puis cliquez sur Aperçu du schéma mis à jour.
6. Passez en revue le schéma mis à jour, puis cliquez sur Enregistrer les modifications.

Si votre schéma ne peut pas être corrigé automatiquement, vous pouvez voir une liste d'erreurs à corriger vous-même dans la console.

1. Ouvrez la page Modifier le schéma comme décrit ci-dessus.
2. Sélectionnez le mode JSON.
3. Passez la souris sur l'icône d'erreur rouge dans le caniveau sur le côté gauche de l'éditeur de code. Le message d'erreur s'affiche dans une infobulle.

Voici quelques erreurs courantes que vous pouvez rencontrer et comment les résoudre :

échec de l'analyse du schéma à partir de JSON : `` ***field-name***

Avec Cedar 2, vous pouvez inclure des champs arbitraires dans des parties de schémas telles que les définitions de type, même s'ils n'ont aucune signification dans le cadre d'un schéma Cedar. Dans Cedar 4, cela n'est plus autorisé. Pour résoudre cette erreur, supprimez le champ appelé ***field-name*** de votre schéma JSON. Pour une liste des champs de schéma valides, consultez la [documentation de Cedar](#).

type d'extension inconnu ` ***extension-name*** `

Dans Cedar 2, lorsque vous déclarez un attribut dont le nom type est `Extension`, vous pouvez spécifier n'importe quelle valeur pour le `name` champ, qu'il s'agisse ou non d'un nom de type d'extension valide. Il s'agit maintenant d'une erreur avec Cedar 4. Pour le résoudre, remplacez-le

extension-name par un nom de type d'extension valide. Vous trouverez une liste des noms de types d'extensions valides dans la [documentation de Cedar](#).

Si vous ne savez toujours pas comment résoudre les erreurs de votre schéma, contactez Support

Comment puis-je rendre mes politiques et mes modèles compatibles avec Cedar 4 ?

La console Verified Permissions vous indique toute erreur dans votre politique ou votre modèle qui le rend incompatible avec Cedar 4.

Pour afficher les erreurs d'une politique ou d'un modèle dans la console

1. Connectez-vous à la console Amazon Verified Permissions AWS Management Console et ouvrez-la à l'adresse [verifiedpermissions](#).
2. Dans le volet de navigation, choisissez Policy Stores, puis choisissez le Policy Store que vous souhaitez vérifier.
3. Choisissez Politiques ou Modèles de politiques dans le volet de navigation, selon le cas.
4. Sélectionnez la politique ou le modèle incompatible.
5. Sélectionnez Modifier
6. Passez la souris sur l'icône d'erreur rouge dans le caniveau sur le côté gauche de l'éditeur de code. Le message d'erreur s'affiche dans une infobulle.

Voici quelques erreurs courantes que vous pouvez rencontrer et comment les résoudre :

les littéraux vides sont interdits dans les politiques

Dans Cedar 2, vous pouvez utiliser la syntaxe `mySet == []` pour vérifier si un ensemble est vide. Avec Cedar 4, les politiques utilisant cette syntaxe ne sont plus validées par rapport à un schéma. Remplacez `mySet == []` dans votre police d'assurance par `mySet.isEmpty()`.

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
Suppression définitive pour les alias du Policy Store	Vous pouvez désormais supprimer définitivement les alias du Policy Store pour les supprimer immédiatement, en contournant ainsi la période de PendingDeletion réservation de 24 heures.	17 avril 2026
Noms des politiques et noms des modèles de politiques	Vous pouvez désormais attribuer des noms aux politiques et aux modèles de politiques pour les référencer par des noms conviviaux.	4 mars 2025
Alias du Policy Store	Vous pouvez désormais créer des alias de magasin de politiques pour faire référence à vos magasins de politiques par des noms conviviaux.	26 février 2025
Nouvelles politiques AWS gérées	Vous pouvez désormais utiliser les politiques AmazonVerifiedPermissionsReadOnlyAccess IAM gérées AmazonVerifiedPermissionsFullAccess et avec des autorisations vérifiées.	11 octobre 2024

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 Amazon Cognito d'utilisateurs dans 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 et Amazon Cognito d'un pool d'utilisateurs existants.	1er avril 2024
Concepts contextuels et exemple	Ajout d'informations sur le contexte des demandes d'autorisation avec autorisations vérifiées.	1er février 2024
Concepts d'autorisation et exemple	Ajout d'informations sur les demandes d'autorisation avec autorisations vérifiées.	1er février 2024
AWS CloudFormation intégré	Verified Permissions prend en charge la création de sources d'identité, de politiques, de magasins de politiques et de modèles de politiques dans CloudFormation.	30 juin 2023
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.