



Guide du développeur

Amazon Cognito



Amazon Cognito: Guide du développeur

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

Les marques déposées et la présentation commerciale d'Amazon ne peuvent pas être utilisées en relation avec un produit ou un service extérieur à Amazon, d'une manière susceptible d'entraîner une confusion chez les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques déposées 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 qu'Amazon Cognito ?	1
Groupes d'utilisateurs	2
Réserves d'identités	3
Caractéristiques d'Amazon Cognito	4
Groupes d'utilisateurs	4
Réserves d'identités	7
Comparaison des groupes d'utilisateurs et des réserves d'identités Amazon Cognito	9
Démarrer avec Amazon Cognito	14
Disponibilité par région	14
Tarification Amazon Cognito	15
Comment fonctionne l'authentification	15
Authentification du SDK	16
Authentification de l'interface utilisateur	19
Authentification par un fournisseur d'identité tiers	22
Authentification du pool d'identités	25
Termes relatifs à Amazon Cognito	28
Général	29
Groupes d'utilisateurs	31
Réserves d'identités	33
Utilisation des AWS SDK	35
Commencer avec AWS	36
Inscrivez-vous pour un Compte AWS	36
Création d'un utilisateur doté d'un accès administratif	36
Démarrage avec les groupes d'utilisateurs	39
Exemple de React SPA	39
Création d'une application	44
Création d'un environnement de développement Lightsail	46
Exemple d'application mobile Flutter	46
Création d'une application	52
Étapes suivantes	54
Créer un groupe d'utilisateurs	55
Ajouter un client d'application d'interface utilisateur hébergé	59
Ajout d'un fournisseur social	63
Ajout d'un fournisseur SAML	71

Démarrage avec les groupes d'identités	74
Créer un groupe d'identités dans Amazon Cognito	74
Configuration d'un kit SDK	76
Intégrer les fournisseurs d'identité	77
Obtenir des informations d'identification	77
Options de mise en route supplémentaires	78
Intégration aux applications	80
Authentification avec AWS Amplify	81
Créer une interface utilisateur (UI) avec Amplify	82
Authentification avec les kits AWS SDK	83
Autorisation avec Amazon Verified Permissions	84
Autorisation d'API avec autorisations vérifiées	86
Exemple de politique pour un utilisateur Amazon Cognito	89
Exemples de code	92
Amazon Cognito Identity	94
Actions	95
Exemples de services croisés	117
Fournisseur d'identité Amazon Cognito	119
Actions	128
Scénarios	248
Amazon Cognito Sync	373
Actions	374
Bonnes pratiques pour les applications multilocataires	376
Groupes d'utilisateurs par locataire	378
Clients d'applications par locataire	380
Groupes de groupes d'utilisateurs par locataire	382
Attributs personnalisés par locataire	384
Recommandations en matière de sécurité multilocataires	386
Scénarios Amazon Cognito courants	388
S'authentifier avec un groupe d'utilisateurs	388
Accès à vos ressources côté serveur	389
Accéder à des ressources avec API Gateway et Lambda	390
AWS Services d'accès avec un pool d'utilisateurs et un pool d'identités	391
S'authentifier avec un tiers et accéder aux services AWS avec un groupe d'identités	392
Accédez aux AWS AppSync ressources avec Amazon Cognito	393
Groupes d'utilisateurs Amazon Cognito	395

Fonctionnalités	396
Inscription	396
Connexion	397
Interface utilisateur hébergée	398
Sécurité	399
Expérience utilisateur personnalisée	399
Surveillance et analytique	400
Intégration des réserves d'identités Amazon Cognito	400
Authentification	401
Flux d'authentification de groupe d'utilisateurs	403
Clients d'application	414
Utilisation des appareils	426
Utilisation de l'API et des points de terminaison	433
Authentification de l'API des groupes d'utilisateurs	436
Mise à jour d'un groupe d'utilisateurs	445
Configuration des SMS	446
Mettre à jour un groupe d'utilisateurs à l'aide d'un AWS SDK ou d'une API REST AWS CDK	447
Interface utilisateur hébergée et serveur OAuth	449
Configuration de l'interface utilisateur hébergée avec AWS Amplify	450
Configuration de l'interface utilisateur hébergée avec la console Amazon Cognito	451
Affichage de votre page de connexion	454
Informations utiles concernant l'interface utilisateur hébergée des groupes d'utilisateurs Amazon Cognito	455
Configuration d'un domaine	457
Personnalisation des pages web intégrées	467
Comment utiliser l'interface utilisateur hébergée	474
Portées et serveurs de ressources	493
Autorisation Machine-to-machine (M2M)	494
À propos des portées	495
À propos des serveurs de ressources	497
Ajout de la connexion via un tiers	502
Fonctionnement de la connexion fédérée dans les groupes d'utilisateurs Amazon Cognito ..	502
Responsabilités d'une application en tant que fournisseur de services Amazon Cognito	503
Informations utiles concernant la connexion tierce aux groupes d'utilisateurs Amazon Cognito	504

Fournisseurs d'identité	505
Fournisseurs d'identité sociale	512
Fournisseurs SAML	521
Fournisseurs OIDC	555
Spécification des mappages d'attributs	566
Liaison d'utilisateurs fédérés à un profil utilisateur existant	571
Utilisation de déclencheurs Lambda	575
Considérations Importantes	578
Ajout d'un déclencheur au groupe d'utilisateurs	581
Événement déclencheur Lambda d'un groupe d'utilisateurs	582
Paramètres communs des déclencheurs Lambda de groupe d'utilisateurs	582
Sources de déclencheur Lambda par événement	584
Sources du déclencheur Lambda par fonction	590
Déclencheur Lambda Avant l'inscription	594
Déclencheur Lambda après confirmation.	604
Déclencheur Lambda avant l'authentification.	609
Déclencheur Lambda après l'authentification.	613
Déclencheurs Lambda de stimulation	618
Déclencheur Lambda avant la génération de jeton	634
Déclencheur Lambda de migration d'utilisateur	654
Déclencheur Lambda message personnalisé	661
Déclencheurs Lambda Expéditeur personnalisé	668
Utilisation de l'analytique Amazon Pinpoint	686
Rechercher des mappages de régions Amazon Cognito et Amazon Pinpoint	687
Intégration de votre application à Amazon Pinpoint	692
Analyse	693
Gestion des utilisateurs	695
Autorisation de l'inscription des utilisateurs	695
Inscription et confirmation des comptes d'utilisateur	698
Création d'utilisateurs en tant qu'administrateur	726
Ajout de groupes à un groupe d'utilisateurs	732
Gestion et recherche d'utilisateurs	735
Récupération de comptes d'utilisateurs	740
Importation d'utilisateurs dans un groupe d'utilisateurs	741
Attributs	760
Exigences de mot de passe	774

Paramètres de messagerie	776
Configuration du courrier électronique par défaut	777
Configuration de l'e-mail Amazon SES	778
Configuration du compte de messagerie	784
Paramètres des SMS	790
Configuration de l'envoi de SMS pour la première fois dans des groupes d'utilisateurs Amazon Cognito	792
Utilisation des jetons	799
Utilisation du jeton d'identification	801
Utilisation du jeton d'accès	806
Utilisation du jeton actualisation	810
Révocation de jetons	812
Vérification d'un jeton web JSON	814
Jetons de mise en cache	820
Accès à des ressources après la connexion	824
Accès aux ressources avec des autorisations vérifiées	389
Accès aux ressources avec API Gateway et AWS AppSync	827
Accès aux AWS ressources à l'aide d'un pool d'identités	829
Utiliser les fonctions de sécurité	834
Ajout de l'authentification MFA	835
Ajout de la sécurité avancée	848
AWS WAF ACL Web	866
Sensibilité à la casse	871
Deletion protection (Protection contre la suppression)	872
Gestion de la divulgation des utilisateurs	874
Groupes d'identités Amazon Cognito	881
Utilisation des groupes d'identités	883
Rôles IAM d'utilisateur	885
Identités authentifiées et non authentifiées	886
Activation ou désactivation de l'accès invité	886
Modification du rôle associé à un type d'identité	887
Modification des fournisseurs d'identité	888
Supprimer un groupe d'identités	890
Supprimer une identité d'un groupe d'identités	891
Utilisation d'Amazon Cognito Sync avec des groupes d'identités	891
Concepts de groupes d'identités	894

Flux d'authentification des groupes d'identités	895
Rôles IAM	905
Autorisations et approbation de rôle	920
Bonnes pratiques de sécurité	921
Bonnes pratiques en matière de configuration IAM	922
Bonnes pratiques en matière de configuration du pool d'identités	924
Utilisation d'attributs pour le contrôle d'accès	926
Utilisation d'attributs pour le contrôle d'accès avec les groupes d'identités Amazon Cognito	927
Exemple d'utilisation d'attributs pour une stratégie de contrôle d'accès	928
Désactiver les attributs pour le contrôle d'accès	931
Mappages de fournisseurs par défaut	931
Utilisation du contrôle d'accès basé sur les rôles	933
Création de rôles pour le mappage de rôles	933
Octroi d'une autorisation de transmission de rôle	934
Utilisation de jetons pour attribuer des rôles aux utilisateurs	935
Utilisation du mappage basé sur des règles pour attribuer des rôles aux utilisateurs	936
Demandes de jetons à utiliser dans le mappage basé sur des règles	938
Bonnes pratiques pour le contrôle d'accès basé sur les rôles	940
Obtention des informations d'identification	940
Accès aux AWS services	948
Fournisseurs d'identité externes aux groupes d'identités	951
Facebook	952
Login with Amazon	960
Google	966
Se connecter avec Apple	979
Fournisseurs OpenID Connect	987
Fournisseurs d'identité SAML	991
Identités authentifiées par le développeur	995
Présentation du flux d'authentification	996
Définir un nom de fournisseur de développement et l'associer à un groupe d'identités	997
Implémentation d'un fournisseur d'identité	997
Mise à jour de la carte de connexions (Android et iOS uniquement)	1005
Obtention d'un jeton (côté serveur)	1006
Connexion à une identité sociale existante	1008
Transition d'un fournisseur à un autre	1008

Changement d'identités	1013
Android	1013
iOS – objective-C	1013
iOS – swift	1014
JavaScript	1014
Unity	1015
Xamarin	1016
Amazon Cognito Sync	1017
Démarrer avec Amazon Cognito Sync	1018
Configurer un groupe d'identités dans Amazon Cognito	1018
Stocker et synchroniser les données	1018
Synchronisation des données	1018
Initialisation du client Amazon Cognito Sync	1019
Comprendre les jeux de données	1021
Lecture et écriture de données dans les jeux de données	1023
Synchronisation des données locales avec le magasin de synchronisation	1025
Gestion des rappels	1029
Android	1029
iOS : Objective-C	1032
iOS : Swift	1035
JavaScript	1039
Unity	1041
Xamarin	1044
Synchronisation en mode Push	1047
Créer une application Amazon Simple Notification Service (Amazon SNS)	1047
Activer la synchronisation en mode Push via la console Amazon Cognito.	1048
Utilisation de la synchronisation en mode Push dans votre application : Android	1049
Utilisation de la synchronisation en mode Push dans votre application : iOS - Objective-C .	1051
Utilisation de la synchronisation en mode Push dans votre application : iOS - Swift	1054
Flux Amazon Cognito	1057
Événements Amazon Cognito	1060
Utilisation de la console Amazon Cognito	1066
Console des groupes d'utilisateurs	1067
Console des réserves d'identités	1069
Sécurité	1071
Protection des données	1072

Chiffrement des données	1072
Gestion des identités et des accès	1074
Public ciblé	1074
Authentification par des identités	1075
Gestion des accès à l'aide de politiques	1079
Fonctionnement d'Amazon Cognito avec IAM	1082
Exemples de politiques basées sur l'identité	1092
Résolution des problèmes	1097
Utilisation des rôles liés à un service	1099
Journalisation et surveillance	1104
Coûts de surveillance	1105
Suivi des quotas, de l'utilisation CloudWatch et des Quotas de Service	1108
Journalisation des appels d'API Amazon Cognito avec AWS CloudTrail	1125
Validation de conformité	1152
Résilience	1153
Considérations sur les données régionales	1153
Sécurité de l'infrastructure	1154
Analyse de la configuration et des vulnérabilités	1155
AWS politiques gérées	1155
Mises à jour des politiques	1157
Étiquetage des ressources	1159
Ressources prises en charge	1160
Restrictions liées aux étiquettes	1160
Gestion des identifications avec la console	1160
Exemples AWS CLI	1161
Affectation de balises	1161
Affichage des balises	1162
Suppression de balises	1163
Application de balises au moment de créer des ressources	1164
Actions d'API	1165
Actions d'API pour les identifications de groupe d'utilisateurs	1165
Actions d'API pour les identifications de groupe d'identités	1165
Quotas	1166
Comprendre les quotas de taux de demandes d'API	1166
Catégorisation des quotas	1166

Opérations API des groupes d'utilisateurs Amazon Cognito avec traitement des taux de demandes spéciaux	1167
Monthly active users (Utilisateurs actifs mensuels)	1168
Gestion des quotas de taux de demandes d'API	1169
Identifier les besoins en matière de quota	1169
Optimisez les taux de demandes	1170
Suivre l'usage des quotas	1171
Suivez les utilisateurs actifs mensuels (MAU)	1172
Demande d'augmentation de quota	1172
Quotas de taux de demandes des groupes d'utilisateurs	1173
Quotas de taux de demande pour les groupes d'identités	1185
Quotas relatifs au nombre et à la taille des ressources	1187
Références API	1196
Référence des points de terminaison de groupe d'utilisateurs	1196
Référence des points de terminaison d'interface utilisateur hébergée	1198
Référence des points de terminaison de fédération	1206
Octrois OAuth 2.0	1232
Utilisation de PKCE	1234
Réponses aux erreurs de l'interface utilisateur hébergée et de fédération	1236
Référence d'API des groupes d'utilisateurs	1238
Référence d'API des groupes d'identités	1239
Référence d'API Cognito Sync	1239
Historique de la documentation	1240
.....	mcclix

Qu'est-ce qu'Amazon Cognito ?

Amazon Cognito est une plateforme d'identité pour les applications Web et mobiles. Il s'agit d'un annuaire d'utilisateurs, d'un serveur d'authentification et d'un service d'autorisation pour les jetons d'accès OAuth 2.0 et les informations d'identification AWS . Avec Amazon Cognito, vous pouvez authentifier et autoriser les utilisateurs à partir de l'annuaire d'utilisateurs intégré, de votre annuaire d'entreprise et de fournisseurs d'identité grand public tels que Google et Facebook.

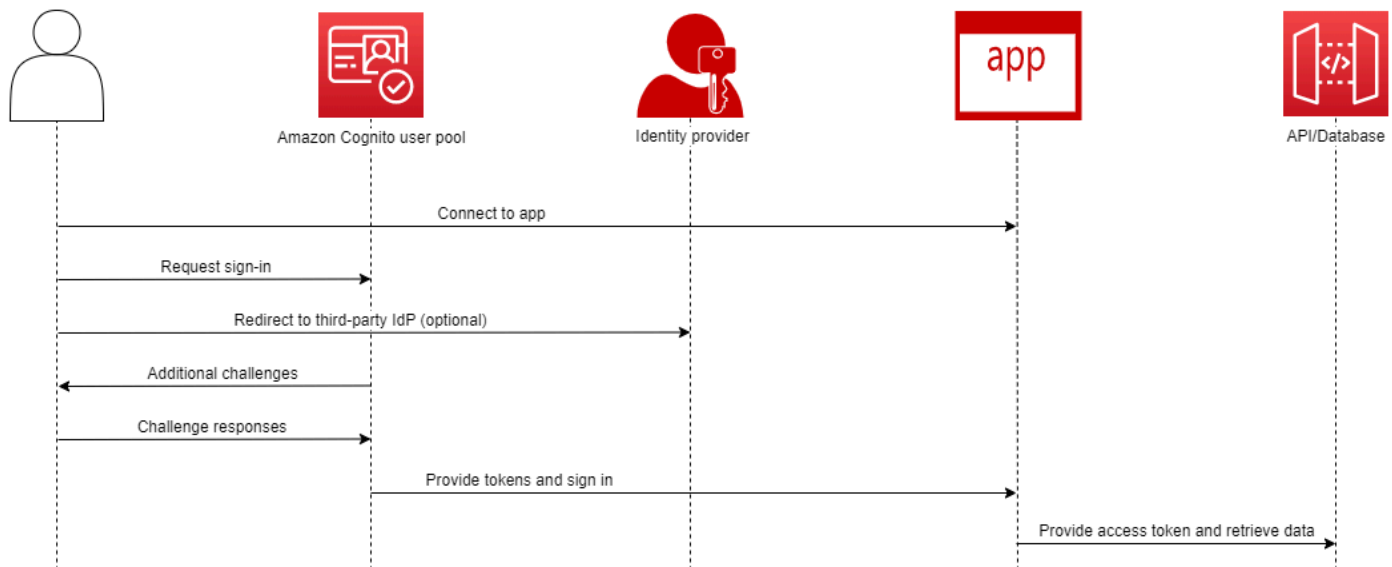
Rubriques

- [Groupes d'utilisateurs](#)
- [Réserves d'identités](#)
- [Caractéristiques d'Amazon Cognito](#)
- [Comparaison des groupes d'utilisateurs et des réserves d'identités Amazon Cognito](#)
- [Démarrer avec Amazon Cognito](#)
- [Disponibilité par région](#)
- [Tarification Amazon Cognito](#)
- [Comment fonctionne l'authentification avec les groupes d'utilisateurs et les groupes d'identités Amazon Cognito](#)
- [Termes relatifs à Amazon Cognito](#)
- [Utilisation de ce service avec un AWS SDK](#)
- [Commencer avec AWS](#)

Les deux composants qui suivent constituent Amazon Cognito. Ils fonctionnent indépendamment ou en tandem, en fonction des besoins d'accès de vos utilisateurs.

Groupes d'utilisateurs

Amazon Cognito user pools

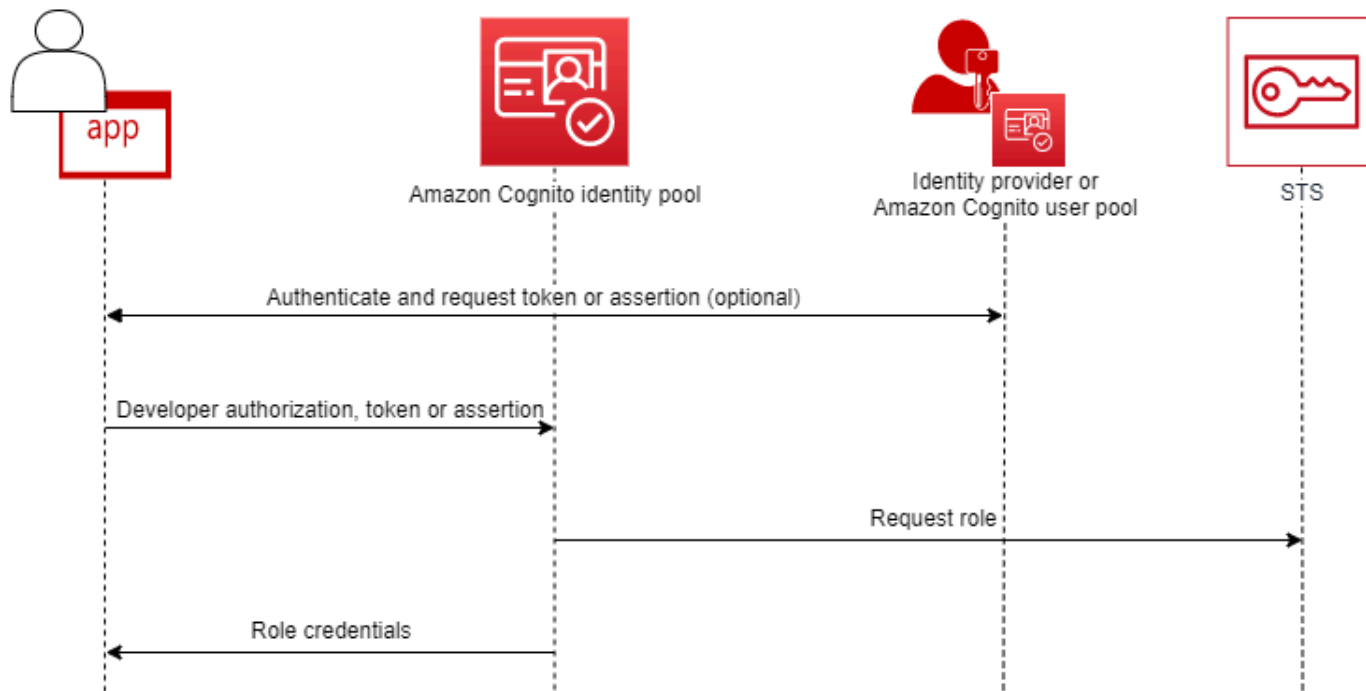


Créez un groupe d'utilisateurs lorsque vous souhaitez authentifier et autoriser les utilisateurs à accéder à votre application ou à votre API. Les groupes d'utilisateurs sont des annuaires d'utilisateurs permettant à la fois la création, la gestion et l'authentification des utilisateurs en libre-service et pilotées par l'administrateur. Votre groupe d'utilisateurs peut être un annuaire indépendant et un fournisseur d'identité (IdP) OIDC, ainsi qu'un fournisseur de services intermédiaire (SP) auprès de fournisseurs tiers d'identités du personnel et des clients. Vous pouvez fournir une authentification unique (SSO) dans votre application pour les identités du personnel de votre organisation dans SAML 2.0 et OIDC IdPs avec des groupes d'utilisateurs. Vous pouvez également fournir l'authentification unique dans votre application pour l'identité des clients de votre organisation dans les magasins d'identité publics OAuth 2.0 Amazon, Google, Apple et Facebook. Pour plus d'informations sur CIAM (gestion de l'identité et de l'accès des clients), consultez [Qu'est-ce que CIAM ?](#).

Les groupes d'utilisateurs ne nécessitent pas d'intégration à une réserve d'identités. À partir d'un groupe d'utilisateurs, vous pouvez émettre des jetons Web JSON authentifiés (JWT) directement vers une application, un serveur Web ou une API.

Réserves d'identités

Amazon Cognito federated identities (identity pools)



Configurez un pool d'identités Amazon Cognito lorsque vous souhaitez autoriser des utilisateurs authentifiés ou anonymes à accéder à vos ressources. AWS Un pool d'identités émet des AWS informations d'identification permettant à votre application de fournir des ressources aux utilisateurs. Vous pouvez authentifier les utilisateurs auprès d'un fournisseur d'identité de confiance, tel qu'un groupe d'utilisateurs ou un service SAML 2.0. Il peut également éventuellement émettre des informations d'identification pour les utilisateurs invités. Les pools d'identités utilisent à la fois le contrôle d'accès basé sur les rôles et les attributs pour gérer l'autorisation d'accès de vos utilisateurs à vos ressources. AWS

Les réserves d'identités ne nécessitent pas d'intégration à un groupe d'utilisateurs. Une réserve d'identités peut accepter des champs standard authentifiés émanant directement des fournisseurs d'identité du personnel et des consommateurs.

Un groupe d'utilisateurs et une réserve d'identités Amazon Cognito utilisés conjointement

Dans le diagramme au début de cette rubrique, vous utilisez Amazon Cognito pour authentifier votre utilisateur, puis lui accorder l'accès à un Service AWS.

1. L'utilisateur de votre application se connecte via un groupe d'utilisateurs et reçoit des jetons OAuth 2.0.
2. Votre application échange un jeton de pool d'utilisateurs avec un pool d'identités contre des AWS informations d'identification temporaires que vous pouvez utiliser avec les AWS API et le AWS Command Line Interface (AWS CLI).
3. Votre application attribue la session d'identification à votre utilisateur et fournit un accès autorisé à Amazon S3 et Amazon Services AWS DynamoDB, par exemple.

Pour d'autres exemples utilisant des réserves d'identités et des groupes d'utilisateurs, consultez [Scénarios Amazon Cognito courants](#).

Dans Amazon Cognito, l'obligation de sécurité du cloud dans le cadre du [modèle de responsabilité partagée](#) est conforme à SOC 1-3, PCI DSS et ISO 27001, et est éligible HIPAA-BAA. Dans Amazon Cognito, vous pouvez concevoir votre sécurité dans le cloud afin d'être conforme à SOC 1-3, ISO 27001 et HIPAA-BAA, mais pas PCI DSS. Pour plus d'informations, consultez [Services AWS concernés](#). Consultez également [Considérations sur les données régionales](#).

Caractéristiques d'Amazon Cognito

Groupes d'utilisateurs

Un groupe d'utilisateurs Amazon Cognito est un annuaire d'utilisateurs. Avec un groupe d'utilisateurs, vos utilisateurs peuvent se connecter à votre application web ou mobile via Amazon Cognito, ou se fédérer via un fournisseur d'identité tiers. Les utilisateurs fédérés et locaux ont un profil utilisateur dans votre groupe d'utilisateurs.

Les utilisateurs locaux sont ceux qui se sont inscrits ou que vous avez créés directement dans votre groupe d'utilisateurs. Vous pouvez gérer et personnaliser ces profils utilisateur dans le AWS Management Console, un AWS SDK ou le AWS Command Line Interface (AWS CLI).

Les groupes d'utilisateurs Amazon Cognito acceptent les jetons et les assertions provenant de tiers IdPs, et collectent les attributs utilisateur dans un JWT qu'il envoie à votre application. Vous pouvez standardiser votre application sur un ensemble de JWT pendant qu'Amazon Cognito gère les interactions avec eux IdPs, en mappant leurs revendications à un format de jeton central.

Un groupe d'utilisateurs Amazon Cognito peut être un fournisseur d'identité autonome. Amazon Cognito s'inspire de la norme OpenID Connect (OIDC) pour générer des JWT à des fins d'authentification et d'autorisation. Lorsque vous connectez des utilisateurs locaux, votre groupe

d'utilisateurs fait autorité pour ces utilisateurs. Vous avez accès aux fonctionnalités suivantes quand vous authentifiez des utilisateurs locaux.

- Implémentez votre propre interface Web qui appelle l'API des groupes d'utilisateurs Amazon Cognito pour authentifier, autoriser et gérer vos utilisateurs.
- Configurez une authentification multifactorielle (MFA) pour vos utilisateurs. Amazon Cognito prend en charge la MFA avec un mot de passe unique à durée limitée (TOTP) et par SMS.
- Sécurisez l'accès à partir de comptes utilisateurs contrôlés par des personnes malveillantes.
- Créez vos propres flux d'authentification personnalisés en plusieurs étapes.
- Recherchez des utilisateurs dans un autre annuaire et migrez-les vers Amazon Cognito.

Un groupe d'utilisateurs Amazon Cognito peut également jouer le double rôle de fournisseur de services (SP) pour votre IdPs application et d'IdP pour votre application. Les groupes d'utilisateurs d'Amazon Cognito peuvent se connecter à des clients IdPs tels que Facebook et Google, ou à des employés IdPs tels qu'Okta et Active Directory Federation Services (ADFS).

Avec les jetons OAuth 2.0 et OpenID Connect (OIDC) émis par un groupe d'utilisateurs Amazon Cognito, vous pouvez

- Accepter un jeton d'identification dans votre application qui authentifie un utilisateur et fournit les informations nécessaires pour configurer le profil de l'utilisateur
- Accepter un jeton d'accès dans votre API avec les étendues OIDC qui autorisent les appels d'API de vos utilisateurs.
- Récupérez les AWS informations d'identification d'un pool d'identités Amazon Cognito.

Fonctionnalités des groupes d'utilisateurs Amazon Cognito

Fonctionnalité	Description
IdP OIDC	Émettre des jetons d'identification pour authentifier les utilisateurs
Serveur d'autorisation	Émettre des jetons d'accès pour autoriser l'accès des utilisateurs aux API
SAML 2.0 SP	Transformez les assertions SAML en identifiants et jetons d'accès

OIDC SP	Transformez les jetons OIDC en jetons d'identification et d'accès
Système d'exploitation OAuth 2.0	Transformez les jetons d'identification d'Apple, Facebook, Amazon ou Google en vos propres jetons d'identification et d'accès
Service frontal d'authentification	Inscrivez, gérez et authentifiez les utilisateurs avec l'interface utilisateur hébergée
Support d'API pour votre propre interface utilisateur	Créez, gérez et authentifiez les utilisateurs via des requêtes d'API dans les SDK supportés ¹ AWS
MFA	Utilisez les SMS, les TOTP ou l'appareil de votre utilisateur comme facteur d'authentification supplémentaire ¹
Surveillance de la sécurité et réponse	Protégez-vous contre les activités malveillantes et les mots de passe peu sécurisés ¹
Personnaliser les flux d'authentification	Créez votre propre mécanisme d'authentification ou ajoutez des étapes personnalisées aux flux existants ¹
Groups	Créez des groupes logiques d'utilisateurs et une hiérarchie des revendications de rôles IAM lorsque vous transmettez des jetons à des pools d'identités
Personnalisez les jetons d'identification	Personnalisez vos jetons d'identification avec des demandes nouvelles, modifiées ou supprimées
Personnalisation des attributs utilisateur	Attribuez des valeurs aux attributs utilisateur et ajoutez vos propres attributs personnalisés

¹ La fonctionnalité est disponible uniquement pour les utilisateurs locaux.

Pour plus d'informations sur les groupes d'utilisateurs, consultez [Démarrage avec les groupes d'utilisateurs](#) et la [référence d'API des groupes d'utilisateurs Amazon Cognito](#).

Réserves d'identités

Un pool d'identités est un ensemble d'identifiants uniques, ou identités, que vous attribuez à vos utilisateurs ou invités et que vous autorisez à recevoir des AWS informations d'identification temporaires. Lorsque vous présentez une preuve d'authentification à une réserve d'identités sous la forme de champs standard fiables émanant d'un fournisseur d'identité (IdP) sociale SAML 2.0, OpenID Connect (OIDC) ou OAuth 2.0, vous associez votre utilisateur à une identité figurant dans la réserve d'identités. Le jeton créé par votre pool d'identités pour l'identité peut récupérer les informations d'identification de session temporaires depuis AWS Security Token Service (AWS STS).

Pour compléter les identités authentifiées, vous pouvez également configurer un pool d'identités pour autoriser l' AWS accès sans authentification IdP. Vous pouvez proposer votre propre preuve d'authentification personnalisée ou ne pas vous authentifier. Vous pouvez accorder des AWS informations d'identification temporaires à tout utilisateur de l'application qui en fait la demande, avec des identités [non authentifiées](#). Les réserves d'identités acceptent également les champs standard et émettent des informations d'identification en fonction de votre propre schéma personnalisé, avec des [identités authentifiées par le développeur](#).

Avec les réserves d'identités Amazon Cognito, vous pouvez intégrer les politiques IAM de deux manières dans votre Compte AWS. Vous pouvez utiliser ces deux fonctionnalités ensemble ou individuellement.

Contrôle d'accès basé sur les rôles

Lorsque votre utilisateur transmet les champs standard à votre réserve d'identités, Amazon Cognito choisit le rôle IAM qu'il demande. Pour personnaliser les autorisations du rôle en fonction de vos besoins, vous devez appliquer des politiques IAM à chaque rôle. Par exemple, si votre utilisateur démontre qu'il fait partie du service marketing, il reçoit des informations d'identification pour un rôle avec des politiques adaptées aux besoins d'accès du service marketing. Amazon Cognito peut demander un rôle par défaut, un rôle basé sur des règles qui interrogent les champs standard de votre utilisateur ou un rôle basé sur l'appartenance de votre utilisateur à un groupe d'utilisateurs. Vous pouvez également configurer la politique de confiance du rôle afin qu'IAM ne fasse confiance qu'à votre réserve d'identités pour générer des sessions temporaires.

Attributs pour le contrôle d'accès

Votre réserve d'identités lit les attributs à partir des champs standard de votre utilisateur et les mappe sur les balises de principal dans le cadre de la session temporaire de votre utilisateur. Vous pouvez ensuite configurer vos politiques basées sur les ressources IAM pour autoriser ou refuser l'accès aux ressources en fonction des principaux IAM qui contiennent les balises de session de votre réserve d'identités. Par exemple, si votre utilisateur prouve qu'il fait partie du service marketing, AWS STS balise sa session `Department: marketing`. Votre compartiment Amazon S3 autorise les opérations de lecture sur la base d'une `PrincipalTag` condition [aws](#) : qui nécessite une valeur de `marketing` pour la `Department` balise.

Fonctionnalités des réserves d'identités Amazon Cognito

Fonctionnalité	Description
Groupe d'utilisateurs Amazon Cognito SP	Échangez un jeton d'identification de votre groupe d'utilisateurs contre des informations d'identité Web provenant de AWS STS
SAML 2.0 SP	Échangez des assertions SAML pour des informations d'identification Web à partir de AWS STS
OIDC SP	Échangez des jetons OIDC contre des identifiants d'identité Web auprès de AWS STS
Système d'exploitation OAuth 2.0	Échangez des jetons OAuth d'Amazon, Facebook, Google, Apple et Twitter contre des identifiants d'identité Web provenant de AWS STS
SP personnalisé	Avec les AWS informations d'identification, échangez des demandes dans n'importe quel format contre des informations d'identification Web auprès de AWS STS
Accès non authentifié	Émettre des informations d'identification Web à accès limité sans authentification AWS STS
Contrôle d'accès basé sur les rôles	Choisissez un rôle IAM pour votre utilisateur authentifié en fonction de ses revendications, et configurez vos rôles pour qu'ils ne soient

assumés que dans le contexte de votre pool d'identités

Contrôle d'accès basé sur les attributs

Convertissez les demandes en balises principales pour votre session AWS STS temporaire et utilisez les politiques IAM pour filtrer l'accès aux ressources en fonction des balises principales

Pour plus d'informations sur les groupes d'utilisateurs, consultez [Commencer à utiliser les pools d'identités Amazon Cognito](#) et la [référence d'API des groupes d'identités Amazon Cognito](#).

Comparaison des groupes d'utilisateurs et des réserves d'identités Amazon Cognito

Fonctionnalité	Description	Groupes d'utilisateurs	Réserves d'identités
IdP OIDC	Émettez des jetons d'identification OIDC pour authentifier les utilisateurs de l'application	✓	
Serveur d'autorisation d'API	Émettez des jetons d'accès pour autoriser l'accès des utilisateurs aux API, aux bases de données et aux autres ressources qui acceptent les étendues d'autorisation OAuth 2.0	✓	
Serveur d'autorisation d'identité Web IAM	Générez des jetons que vous pouvez échanger contre des AWS STS AWS		✓

	informations d'identification temporaires	
SP SAML 2.0 et IdP OIDC	Émettez des jetons OIDC personnalisés en fonction des demandes d'un IdP SAML 2.0	✓
OIDC SP et OIDC IdP	Émettez des jetons OIDC personnalisés en fonction des demandes d'un IdP OIDC	✓
SP OAuth 2.0 et IdP OIDC	Émettez des jetons OIDC personnalisés en fonction des champs d'application fournis par les fournisseurs sociaux OAuth 2.0 tels qu'Apple et Google	✓
SP SAML 2.0 et courtier d'informations d'identification	Émettre des AWS informations d'identification temporaires basées sur les demandes d'un IdP SAML 2.0	✓
OIDC SP et courtier d'informations d'identification	Émettre des AWS informations d'identification temporaires sur la base des demandes d'un IdP OIDC	✓

SP OAuth 2.0 et courtier d'informations d'identification	Émettez des AWS informations d'identification temporaires basées sur les champs d'application fournis par les fournisseurs sociaux OAuth 2.0 tels qu'Apple et Google	✓
Groupe d'utilisateurs Amazon Cognito SP et courtier d'informations d'identification	Émettez des AWS informations d'identification temporaires basées sur des demandes OIDC émanant d'un groupe d'utilisateurs Amazon Cognito	✓
SP personnalisé et courtier d'informations d'identification	Émettre des AWS informations d'identification temporaires basées sur l'autorisation IAM du développeur	✓
Service frontal d'authentification	Inscrivez, gérez et authentifiez les utilisateurs avec l'interface utilisateur hébergée	✓

Support d'API pour votre propre interface utilisateur d'authentification	Créez, gérez et authentifiez les utilisateurs via des requêtes d'API dans les SDK supportés ¹ AWS	✓
MFA	Utilisez les SMS, les TOTP ou l'appareil de votre utilisateur comme facteur d'authentification supplémentaire ¹	✓
Surveillance de la sécurité et réponse	Protégez-vous contre les activités malveillantes et les mots de passe peu sécurisés ¹	✓
Personnaliser les flux d'authentification	Créez votre propre mécanisme d'authentification ou ajoutez des étapes personnalisées aux flux existants ¹	✓
Groups	Créez des groupes logiques d'utilisateurs et une hiérarchie des revendications de rôles IAM lorsque vous transmettez des jetons à des pools d'identités	✓

Personnalisez les jetons d'identification	Personnalisez vos jetons d'identification avec des demandes nouvelles, modifiées ou supprimées	✓
AWS WAF ACL Web	Surveillez et contrôlez les demandes adressées à votre environnement d'authentification avec AWS WAF	✓
Personnalisation des attributs utilisateur	Attribuez des valeurs aux attributs utilisateur et ajoutez vos propres attributs personnalisés	✓
Accès non authentifié	Émettre des informations d'identification Web à accès limité sans authentification AWS STS	✓
Contrôle d'accès basé sur les rôles	Choisissez un rôle IAM pour votre utilisateur authentifié en fonction de ses revendications, et configurez vos rôles pour qu'ils ne soient assumés que dans le contexte de votre pool d'identités	✓

Contrôle d'accès basé sur les attributs	Transformez les demandes des utilisateurs en balises principales pour votre session AWS STS temporaire et utilisez les politiques IAM pour filtrer l'accès aux ressources en fonction des balises principales	✓
---	---	---

¹ La fonctionnalité est disponible uniquement pour les utilisateurs locaux.

Démarrer avec Amazon Cognito

Par exemple, les applications de pool d'utilisateurs, voir [Démarrage avec les groupes d'utilisateurs](#).

Pour une présentation des pools d'identités, consultez [Commencer à utiliser les pools d'identités Amazon Cognito](#).

Pour des liens vers des expériences de configuration guidée avec des groupes d'utilisateurs et des groupes d'identités, consultez [Options de configuration guidée pour Amazon Cognito](#).

Pour accéder à des vidéos, à des articles, à de la documentation et à d'autres exemples d'applications, consultez les [ressources destinées aux développeurs Amazon Cognito](#).

Pour utiliser Amazon Cognito, vous devez avoir un Compte AWS. Pour plus d'informations, consultez [Commencer avec AWS](#).

Disponibilité par région

Amazon Cognito est disponible dans plusieurs AWS régions du monde. Dans chaque région, Amazon Cognito est réparti sur plusieurs zones de disponibilité. Ces zones de disponibilité sont physiquement isolées mais sont reliées par des connexions réseau privées, à latence faible, à débit élevé et à forte redondance. Ces zones de disponibilité permettent AWS de fournir des services,

notamment Amazon Cognito, avec des niveaux de disponibilité et de redondance très élevés, tout en minimisant le temps de latence.

Pour obtenir une liste de toutes les régions où Amazon Cognito est actuellement disponible, consultez [Régions et points de terminaison AWS](#) du Référence générale d'Amazon Web Services. Pour plus d'informations sur le nombre de zones de disponibilité disponibles dans chaque région, consultez [Infrastructure mondiale AWS](#).

Tarification Amazon Cognito

Pour plus d'informations sur la tarification Amazon Cognito, consultez [Tarification Amazon Cognito](#).

Comment fonctionne l'authentification avec les groupes d'utilisateurs et les groupes d'identités Amazon Cognito

Lorsque votre client se connecte à un groupe d'utilisateurs Amazon Cognito, votre application reçoit des jetons Web JSON (JWT).

Lorsque votre client se connecte à un pool d'identités, soit avec un jeton de groupe d'utilisateurs, soit avec un autre fournisseur, votre application reçoit des AWS informations d'identification temporaires.

Avec la connexion au groupe d'utilisateurs, vous pouvez implémenter l'authentification et l'autorisation entièrement à l'aide d'un AWS SDK. Si vous ne souhaitez pas créer vos propres composants d'interface utilisateur (UI), vous pouvez appeler une interface utilisateur Web prédéfinie (l'interface utilisateur hébergée) ou la page de connexion de votre fournisseur d'identité tiers (IdP).

Cette rubrique présente certaines des manières dont votre application peut interagir avec Amazon Cognito pour s'authentifier à l'aide de jetons d'identification, autoriser à l'aide de jetons d'accès et accéder à l'aide des informations d'identification du pool Services AWS d'identités.

Rubriques

- [Authentification et autorisation de l'API du groupe d'utilisateurs avec un AWS SDK](#)
- [Authentification du groupe d'utilisateurs avec l'interface utilisateur hébergée](#)
- [Authentification du groupe d'utilisateurs auprès d'un fournisseur d'identité tiers](#)
- [Authentification du pool d'identités](#)

Authentification et autorisation de l'API du groupe d'utilisateurs avec un AWS SDK

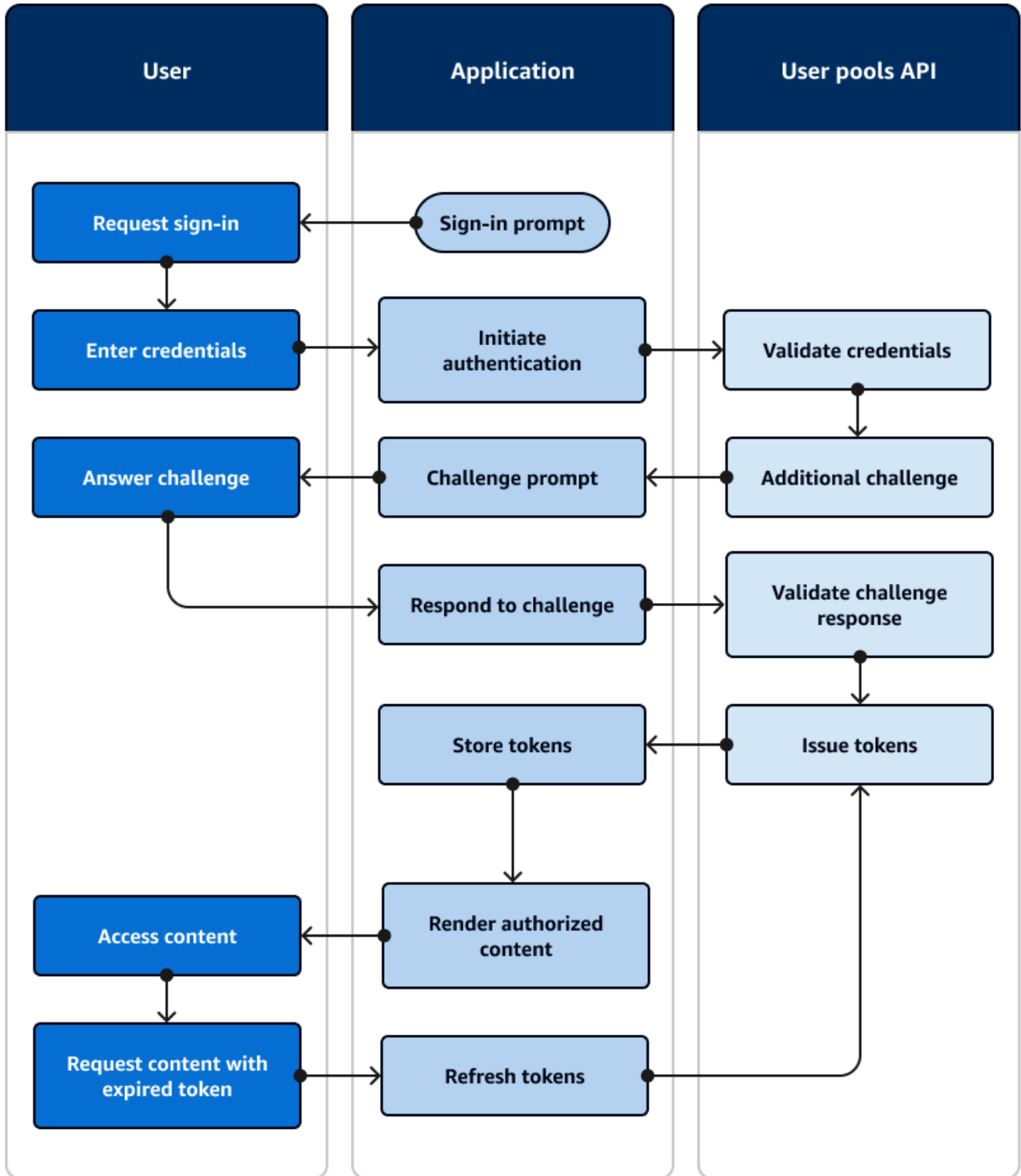
AWS a développé des composants pour les groupes d'utilisateurs Amazon Cognito, ou fournisseur d'identité Amazon Cognito, [dans divers frameworks de](#) développement. Les méthodes intégrées à ces kits de développement logiciel appellent l'API des groupes [d'utilisateurs Amazon Cognito](#). Le même espace de noms d'API de groupes d'utilisateurs comporte des opérations pour la configuration des groupes d'utilisateurs et pour l'authentification des utilisateurs. Pour une présentation plus complète, voir [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#).

L'authentification par API s'adapte au modèle dans lequel vos applications possèdent des composants d'interface utilisateur existants et s'appuient principalement sur le pool d'utilisateurs en tant qu'annuaire d'utilisateurs. Cette conception ajoute Amazon Cognito en tant que composant au sein d'une application plus vaste. Cela nécessite une logique programmatique pour gérer des chaînes complexes de défis et de réponses.

Cette application n'a pas besoin d'implémenter une implémentation complète d'OpenID Connect (OIDC) par une partie dépendante. Au lieu de cela, il a la capacité de décoder et d'utiliser des JWT. Si vous souhaitez accéder à l'ensemble complet des fonctionnalités du pool d'utilisateurs pour [les utilisateurs locaux](#), créez votre authentification avec le SDK Amazon Cognito dans votre environnement de développement.

L'authentification par API avec des étendues OAuth personnalisées est moins orientée vers l'autorisation d'API externe. Pour ajouter des étendues personnalisées à un jeton d'accès à partir de l'authentification par API, modifiez le jeton au moment de l'exécution avec un [Déclencheur Lambda avant génération de jeton](#).

Le schéma suivant illustre une session de connexion typique pour l'authentification par API.



Flux d'authentification par API

1. Un utilisateur accède à votre application.
2. Ils sélectionnent un lien « Se connecter ».
3. Ils saisissent leur nom d'utilisateur et leur mot de passe.
4. L'application invoque la méthode qui effectue une demande d'[InitiateAuth](#)API. La demande transmet les informations d'identification de l'utilisateur à un groupe d'utilisateurs.
5. Le groupe d'utilisateurs valide les informations d'identification de l'utilisateur et détermine que celui-ci a activé l'authentification multifactorielle (MFA).
6. Le groupe d'utilisateurs répond par un défi qui demande un code MFA.
7. L'application génère une invite qui collecte le code MFA auprès de l'utilisateur.
8. L'application invoque la méthode qui effectue une demande d'[RespondToAuthChallenge](#)API. La demande transmet le code MFA de l'utilisateur.
9. Le groupe d'utilisateurs valide le code MFA de l'utilisateur.
- 10 Le groupe d'utilisateurs répond avec les JWT de l'utilisateur.
- 11 L'application décode, valide et stocke ou met en cache les JWT de l'utilisateur.
- 12 L'application affiche le composant à accès contrôlé demandé.
- 13 L'utilisateur consulte leur contenu.
- 14 Plus tard, le jeton d'accès de l'utilisateur a expiré et l'utilisateur demande à consulter un composant dont l'accès est contrôlé.
- 15 L'application détermine que la session de l'utilisateur doit être maintenue. Il invoque à nouveau la [InitiateAuth](#) méthode avec le jeton d'actualisation et récupère de nouveaux jetons.

Variantes et personnalisation

Vous pouvez ajouter des défis supplémentaires à ce flux, par exemple vos propres défis d'authentification personnalisés. Vous pouvez restreindre automatiquement l'accès aux utilisateurs dont les mots de passe ont été compromis ou dont les caractéristiques de connexion inattendues peuvent indiquer une tentative de connexion malveillante. Ce flux est sensiblement le même pour les opérations d'inscription, de mise à jour des attributs utilisateur et de réinitialisation des mots de passe. La plupart de ces flux comportent des opérations d'API publiques (côté client) et confidentielles (côté serveur) dupliquées.

Ressources connexes

- [API des groupes d'utilisateurs Amazon Cognito](#)
- [Démarrage avec les groupes d'utilisateurs](#)
- [Intégration de l'authentification et de l'autorisation Amazon Cognito avec des applications Web et mobiles](#)
- [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#)

Authentification du groupe d'utilisateurs avec l'interface utilisateur hébergée

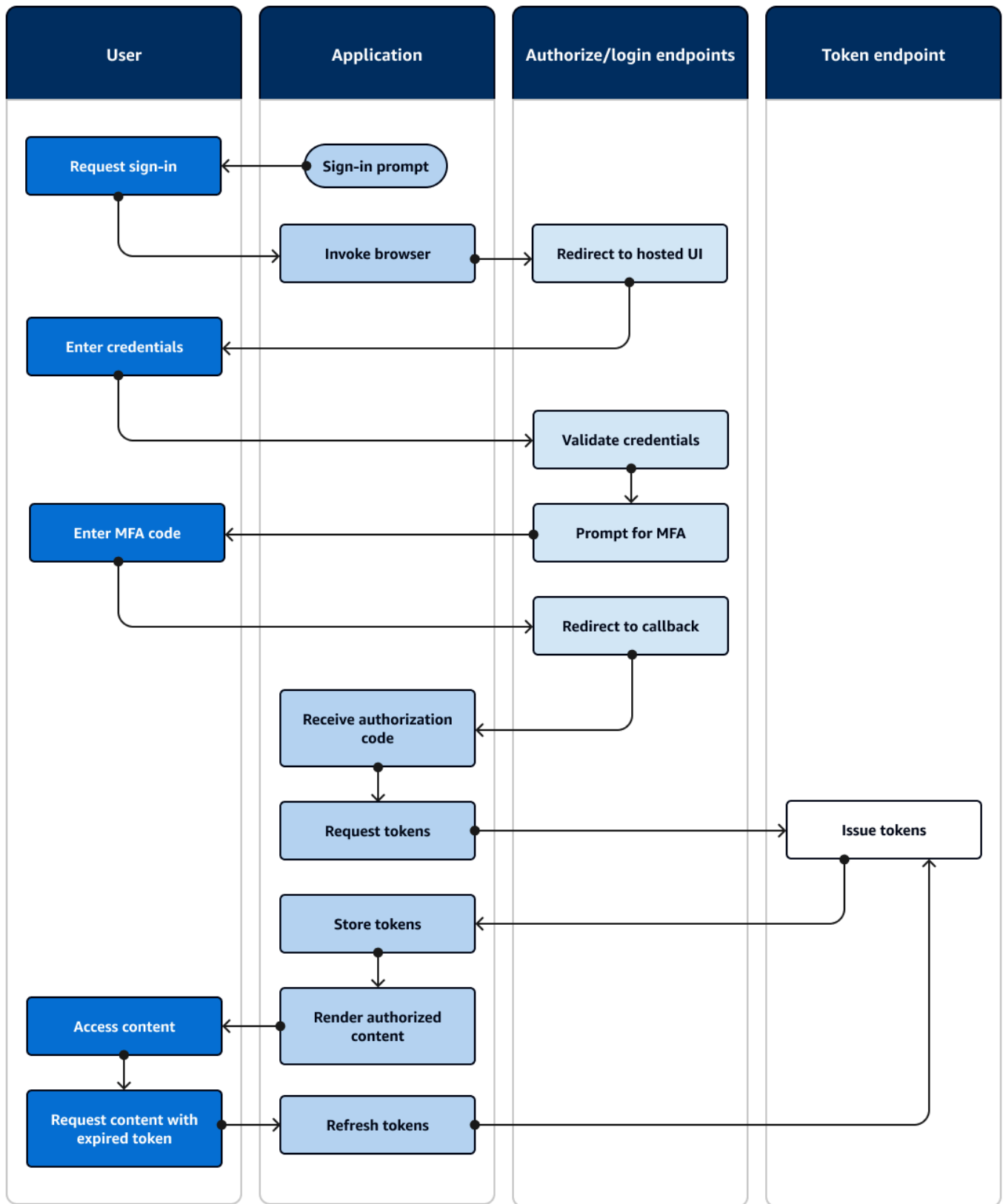
L'[interface utilisateur hébergée](#) est un site Web lié à votre groupe d'utilisateurs et à votre client d'application. Il peut effectuer des opérations de connexion, d'inscription et de réinitialisation du mot de passe pour vos utilisateurs. La mise en œuvre d'une application dotée d'un composant d'interface utilisateur hébergé pour l'authentification peut nécessiter moins d'efforts de développement. Une application peut ignorer les composants de l'interface utilisateur à des fins d'authentification et invoquer l'interface utilisateur hébergée dans le navigateur de l'utilisateur.

Les applications collectent les JWT des utilisateurs à l'aide d'un emplacement de redirection vers le Web ou une application. Les applications qui implémentent l'interface utilisateur hébergée peuvent se connecter aux groupes d'utilisateurs pour s'authentifier comme s'il s'agissait d'un IdP OpenID Connect (OIDC).

L'authentification par interface utilisateur hébergée s'adapte au modèle dans lequel les applications ont besoin d'un serveur d'autorisation, mais n'ont pas besoin de fonctionnalités telles que l'authentification personnalisée, l'intégration de groupes d'identités ou le libre-service des attributs utilisateur. Lorsque vous souhaitez utiliser certaines de ces options avancées, vous pouvez les implémenter avec un composant de groupes d'utilisateurs pour un SDK.

L'interface utilisateur hébergée et les modèles d'authentification IdP tiers, qui reposent principalement sur la mise en œuvre de l'OIDC, sont les meilleurs pour les modèles d'autorisation avancés dotés de champs d'application OAuth 2.0.

Le schéma suivant illustre une session de connexion typique pour l'authentification de l'interface utilisateur hébergée.



Flux d'authentification de l'interface utilisateur hébergée

1. Un utilisateur accède à votre application.
2. Ils sélectionnent un lien « Se connecter ».
3. L'application dirige l'utilisateur vers une invite de connexion à l'interface utilisateur hébergée.
4. Ils saisissent leur nom d'utilisateur et leur mot de passe.
5. Le groupe d'utilisateurs valide les informations d'identification de l'utilisateur et détermine que celui-ci a activé l'authentification multifactorielle (MFA).
6. L'interface utilisateur hébergée invite l'utilisateur à saisir un code MFA.
7. L'utilisateur saisit son code MFA.
8. L'interface utilisateur hébergée redirige l'utilisateur vers l'application.
9. L'application collecte le code d'autorisation à partir du paramètre de demande d'URL que l'interface utilisateur hébergée a ajouté à l'URL de [rappel](#).
- 10L'application demande des jetons avec le code d'autorisation.
- 11Le point de terminaison du jeton renvoie les JWT à l'application.
- 12L'application décode, valide et stocke ou met en cache les JWT de l'utilisateur.
- 13L'application affiche le composant à accès contrôlé demandé.
- 14L'utilisateur consulte leur contenu.
- 15Plus tard, le jeton d'accès de l'utilisateur a expiré et l'utilisateur demande à consulter un composant dont l'accès est contrôlé.
- 16L'application détermine que la session de l'utilisateur doit être maintenue. Il demande de nouveaux jetons au point de terminaison du jeton avec le jeton d'actualisation.

Variantes et personnalisation

Vous pouvez personnaliser l'apparence de l'interface utilisateur hébergée avec du CSS dans n'importe quel [client d'application](#). Vous pouvez également [configurer les clients d'applications](#) avec leurs propres fournisseurs d'identité, leurs propres étendues, leur accès aux attributs utilisateur et leur propre configuration de sécurité avancée.

Ressources connexes

- [Configuration et utilisation de l'interface utilisateur hébergée par Amazon Cognito et des points de terminaison de fédération](#)

- [Inscription et connexion à l'aide de l'interface utilisateur hébergée](#)
- [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#)
- [Référence de l'interface utilisateur hébergée et des points de terminaison de la fédération des groupes d'utilisateurs](#)

Authentification du groupe d'utilisateurs auprès d'un fournisseur d'identité tiers

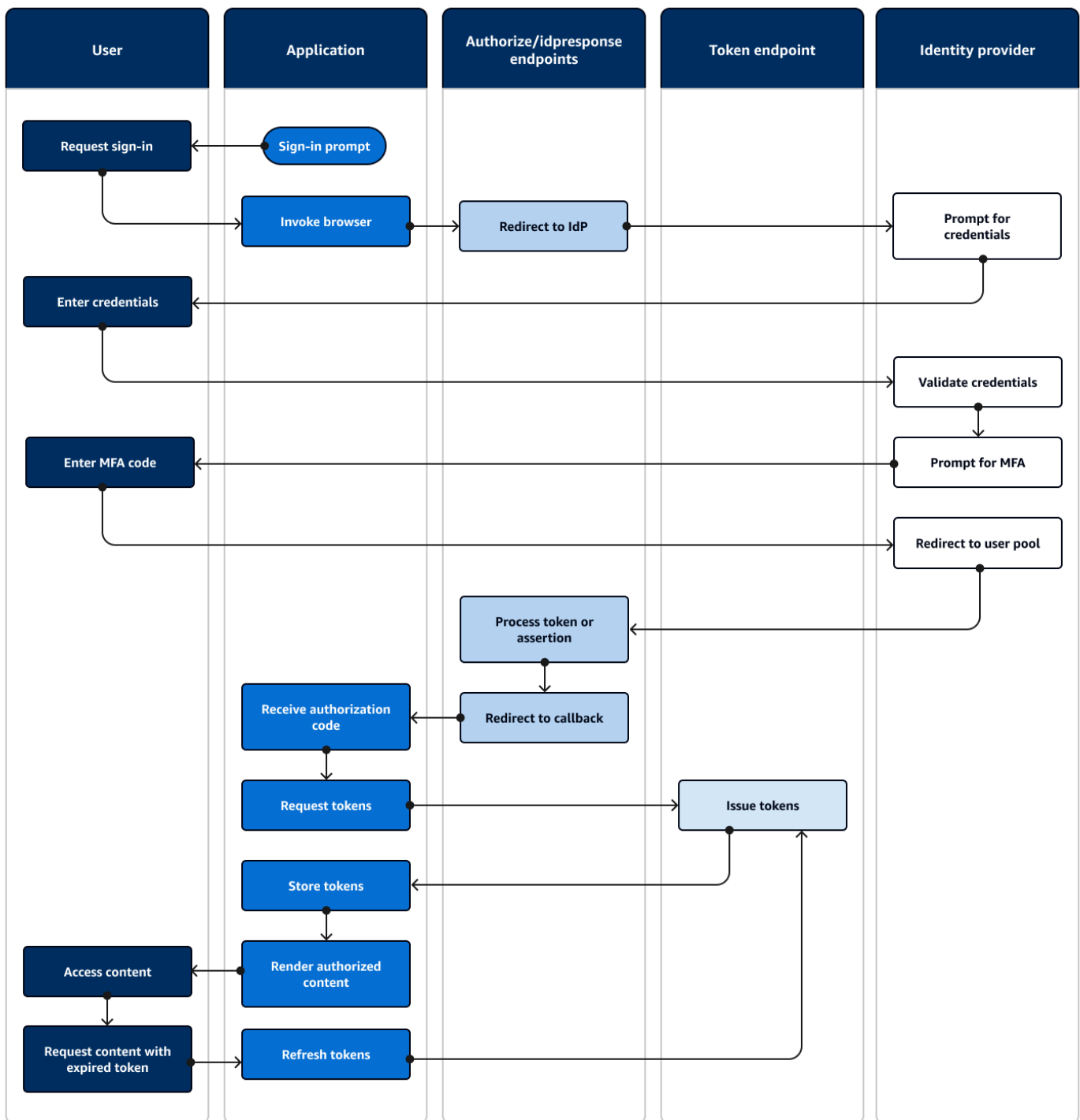
[La connexion avec un fournisseur d'identité externe \(IdP\), ou authentification fédérée, est un modèle similaire à l'interface utilisateur hébergée.](#) Votre application est une partie dépendante de l'OIDC pour votre groupe d'utilisateurs, tandis que votre groupe d'utilisateurs sert de relais à un IdP. L'IdP peut être un annuaire d'utilisateurs grand public tel que Facebook ou Google, ou un annuaire d'entreprise SAML 2.0 ou OIDC tel qu'Azure.

Au lieu de l'interface utilisateur hébergée dans le navigateur de l'utilisateur, votre application invoque un point de terminaison de redirection sur le [serveur d'autorisation du groupe d'utilisateurs](#). Du point de vue de l'utilisateur, il choisit le bouton de connexion dans votre application. Ensuite, leur IdP les invite à se connecter. Comme pour l'authentification par interface utilisateur hébergée, une application collecte des JWT à un emplacement de redirection dans l'application.

L'authentification auprès d'un IdP tiers correspond à un modèle dans lequel les utilisateurs peuvent ne pas vouloir créer un nouveau mot de passe lorsqu'ils s'inscrivent à votre application. L'authentification tierce peut être ajoutée sans effort à une application qui met en œuvre l'authentification par interface utilisateur hébergée. En effet, l'interface utilisateur hébergée et la tierce partie IdPs produisent un résultat d'authentification cohérent à partir de variations mineures de ce que vous invoquez dans les navigateurs des utilisateurs.

À l'instar de l'authentification par interface utilisateur hébergée, l'authentification fédérée est idéale pour les modèles d'autorisation avancés dotés de champs d'application OAuth 2.0.

Le schéma suivant illustre une session de connexion typique pour l'authentification fédérée.



Flux d'authentification fédéré

1. Un utilisateur accède à votre application.
2. Ils sélectionnent un lien « Se connecter ».

3. L'application dirige l'utilisateur vers une invite de connexion avec son IdP.
4. Ils saisissent leur nom d'utilisateur et leur mot de passe.
5. L'IdP valide les informations d'identification de l'utilisateur et détermine que celui-ci a activé l'authentification multifactorielle (MFA).
6. L'IdP invite l'utilisateur à saisir un code MFA.
7. L'utilisateur saisit son code MFA.
8. L'IdP redirige l'utilisateur vers le groupe d'utilisateurs avec une réponse SAML ou un code d'autorisation.
9. Si l'utilisateur a transmis un code d'autorisation, le groupe d'utilisateurs échange silencieusement le code contre des jetons IdP. Le groupe d'utilisateurs valide les jetons IdP et redirige l'utilisateur vers l'application avec un nouveau code d'autorisation.
- 10 L'application collecte le code d'autorisation à partir du paramètre de demande d'URL que le groupe d'utilisateurs a ajouté à l'URL de [rappel](#).
- 11 L'application demande des jetons avec le code d'autorisation.
- 12 Le point de terminaison du jeton renvoie les JWT à l'application.
- 13 L'application décode, valide et stocke ou met en cache les JWT de l'utilisateur.
- 14 L'application affiche le composant à accès contrôlé demandé.
- 15 L'utilisateur consulte leur contenu.
- 16 Plus tard, le jeton d'accès de l'utilisateur a expiré et l'utilisateur demande à consulter un composant dont l'accès est contrôlé.
- 17 L'application détermine que la session de l'utilisateur doit être maintenue. Il demande de nouveaux jetons au point de terminaison du jeton avec le jeton d'actualisation.

Variantes et personnalisation

Vous pouvez lancer l'authentification fédérée dans l'[interface utilisateur hébergée](#), où les utilisateurs peuvent choisir parmi une liste de celles IdPs que vous avez attribuées à votre [client d'application](#). L'interface utilisateur hébergée peut également demander une adresse e-mail et [acheminer automatiquement la demande d'un utilisateur](#) vers l'IdP SAML correspondant. L'authentification auprès d'un fournisseur d'identité tiers ne nécessite aucune interaction de l'utilisateur avec l'interface utilisateur hébergée. Votre application peut ajouter un paramètre de demande à la [demande du serveur d'autorisation](#) d'un utilisateur et obliger celui-ci à être redirigé silencieusement vers sa page de connexion IdP.

Ressources connexes

- [Ajout de la connexion du groupe d'utilisateurs via un tiers](#)
- [Exemple de scénario : ajouter des applications Amazon Cognito à vos favoris dans un tableau de bord d'entreprise](#)
- [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#)
- [Référence de l'interface utilisateur hébergée et des points de terminaison de la fédération des groupes d'utilisateurs](#)

Authentification du pool d'identités

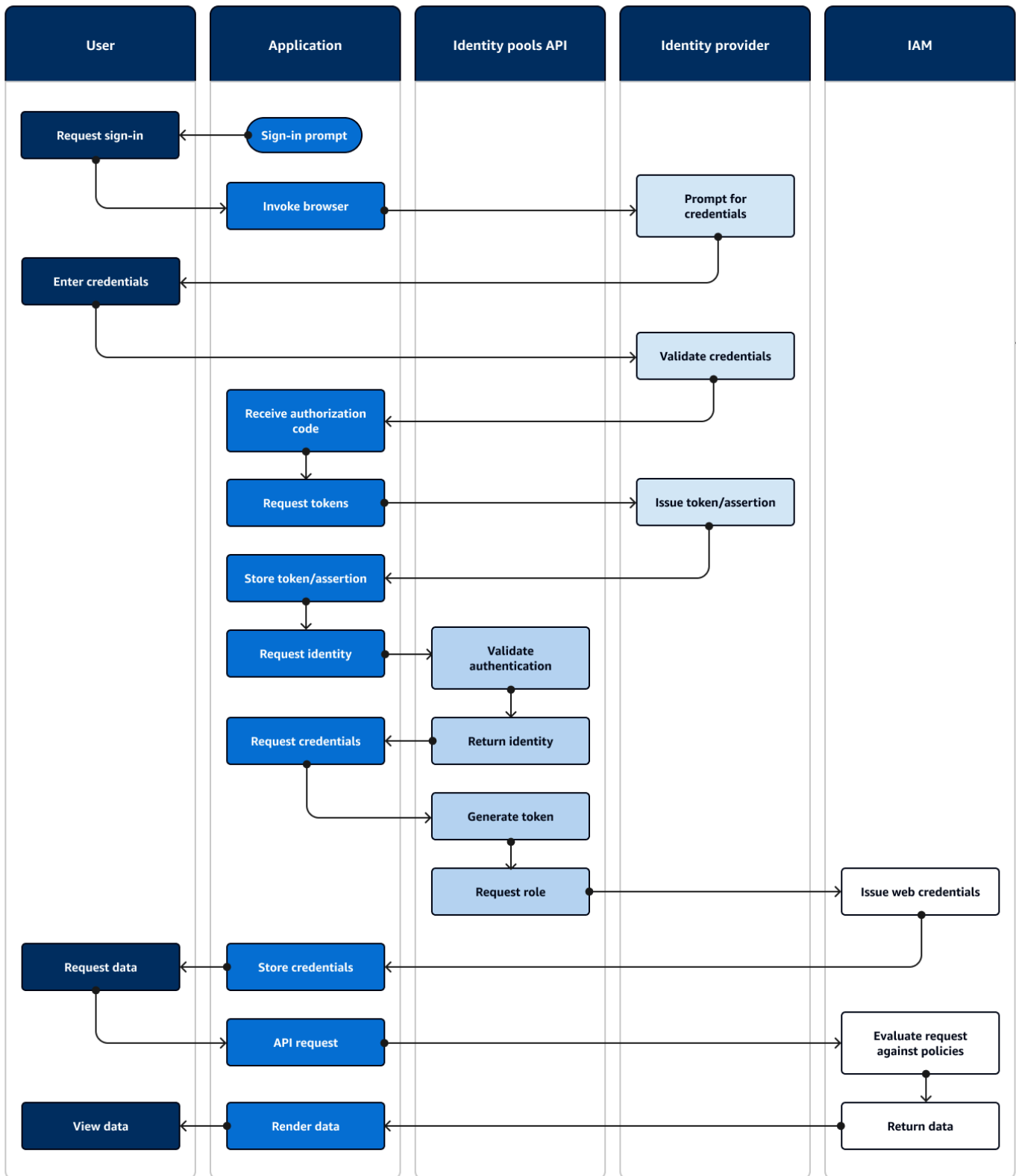
Un pool d'identités est un composant de votre application qui se distingue d'un groupe d'utilisateurs en termes de fonction, d'espace de noms d'API et de modèle de SDK. Lorsque les groupes d'utilisateurs proposent une authentification et une autorisation basées sur des jetons, les pools d'identités offrent une autorisation pour AWS Identity and Access Management (IAM).

Vous pouvez attribuer un ensemble de deux groupes IdPs d'identités et connecter des utilisateurs à leur aide. Les groupes d'utilisateurs sont étroitement intégrés en tant que pool d'identités IdPs et offrent aux groupes d'identités le plus grand nombre d'options en matière de contrôle d'accès. Dans le même temps, il existe un large choix d'options d'authentification pour les pools d'identités. Les groupes d'utilisateurs rejoignent les sources d'identité SAML, OIDC, sociales, de développement et d'invité pour accéder aux informations d'AWS identification temporaires à partir des pools d'identités.

L'authentification avec un pool d'identités est externe : elle suit l'un des flux du groupe d'utilisateurs illustrés précédemment, ou un flux que vous développez indépendamment avec un autre IdP. Une fois que votre application a effectué l'authentification initiale, elle transmet la preuve à un pool d'identités et reçoit une session temporaire en retour.

L'authentification avec un pool d'identités correspond à un modèle dans lequel vous appliquez le contrôle d'accès aux actifs et aux données des applications Services AWS avec l'autorisation IAM. Comme pour [l'authentification par API dans les groupes d'utilisateurs](#), une application performante inclut AWS des SDK pour chacun des services auxquels vous souhaitez accéder dans l'intérêt de vos utilisateurs. AWS Les SDK appliquent les informations d'identification issues de l'authentification du pool d'identités sous forme de signatures aux demandes d'API.

Le schéma suivant illustre une session de connexion typique pour l'authentification du pool d'identités avec un IdP.



Flux d'authentification fédéré

1. Un utilisateur accède à votre application.
2. Ils sélectionnent un lien « Se connecter ».
3. L'application dirige l'utilisateur vers une invite de connexion avec son IdP.
4. Ils saisissent leur nom d'utilisateur et leur mot de passe.
5. L'IdP valide les informations d'identification de l'utilisateur.
6. L'IdP redirige l'utilisateur vers l'application avec une réponse SAML ou un code d'autorisation.
7. Si l'utilisateur transmet un code d'autorisation, l'application échange le code contre des jetons IdP.
8. L'application décode, valide et stocke ou met en cache les JWT ou les assertions de l'utilisateur.
9. L'application invoque la méthode qui effectue une demande d'[GetId](#)API. Il transmet le jeton ou l'assertion de l'utilisateur et demande un identifiant d'identité.
- 10 Le pool d'identités valide le jeton ou l'assertion par rapport aux fournisseurs d'identité configurés.
- 11 Le pool d'identités renvoie un identifiant d'identité.
- 12 L'application invoque la méthode qui effectue une demande d'[GetCredentialsForIdentity](#)API. Il transmet le jeton ou les assertions de l'utilisateur et demande un rôle IAM.
- 13 Le pool d'identités génère un nouveau JWT. Le nouveau JWT contient des revendications qui demandent un rôle IAM. Le pool d'identités détermine le rôle en fonction de la demande de l'utilisateur et des critères de sélection des rôles dans la configuration du pool d'identités pour l'IdP.
- 14 AWS Security Token Service (AWS STS) répond à la [AssumeRoleWithWebIdentity](#) demande du pool d'identités. La réponse contient les informations d'identification de l'API pour une session temporaire avec un rôle IAM.
- 15 L'application enregistre les informations d'identification de session.
- 16 L'utilisateur effectue une action dans l'application qui nécessite l'accès à des ressources protégées. AWS
- 17 L'application applique les informations d'identification temporaires sous forme de [signatures](#) aux demandes d'API pour les informations requises Services AWS.
- 18 IAM évalue les politiques associées au rôle dans les informations d'identification. Il les compare à la demande.
- 19 Service AWS Renvoie les données demandées.
- 20 L'application affiche les données dans l'interface utilisateur.

21 L'utilisateur consulte les données.

Variantes et personnalisation

Pour visualiser l'authentification auprès d'un groupe d'utilisateurs, insérez l'un des aperçus précédents du groupe d'utilisateurs après l'étape Émettre un jeton/une assertion. L'authentification du développeur remplace toutes les étapes précédant la demande d'identité par une demande signée par les [informations d'identification du développeur](#). L'authentification des invités passe également directement à Request identity, ne valide pas l'authentification et renvoie les informations d'identification pour un rôle IAM [à accès limité](#).

Ressources connexes

- [Groupes d'identités Amazon Cognito](#)
- [Rôles IAM d'utilisateur](#)
- [Concepts de groupes d'identités](#)
- [Flux d'authentification de groupes d'identités \(identités fédérées\)](#)

Termes relatifs à Amazon Cognito

Amazon Cognito fournit des informations d'identification pour les applications Web et mobiles. Il s'inspire des termes courants en matière de gestion des identités et des accès et s'appuie sur ceux-ci. De nombreux guides sur l'identité universelle et les conditions d'accès sont disponibles. Voici quelques exemples :

- [La terminologie](#) dans le corpus de connaissances IDPro
- [AWS Services d'identité](#)
- [Glossaire du NIST CSRC](#)

Les listes suivantes décrivent des termes propres à Amazon Cognito ou ayant un contexte spécifique dans Amazon Cognito.

Rubriques

- [Général](#)
- [Groupes d'utilisateurs](#)

- [Réerves d'identités](#)

Général

Les termes de cette liste ne sont pas spécifiques à Amazon Cognito et sont largement reconnus par les professionnels de la gestion des identités et des accès. Ce qui suit n'est pas une liste exhaustive de termes, mais un guide de leur contexte spécifique à Amazon Cognito dans ce guide.

Appli

Généralement, une application mobile. Dans ce guide, application est souvent un raccourci pour désigner une application Web ou une application mobile qui se connecte à Amazon Cognito.

Utilisation du contrôle d'accès basé sur les attributs (ABAC)

Modèle dans lequel une application détermine l'accès aux ressources en fonction des propriétés d'un utilisateur, telles que le titre de son poste ou son département. Les outils Amazon Cognito destinés à appliquer l'ABAC incluent les jetons d'identification dans les groupes d'utilisateurs et les [balises principales](#) dans les groupes d'identités.

Serveur d'autorisation

Système Web qui génère des [jetons Web JSON](#). Les [points de terminaison de la fédération](#) des groupes d'utilisateurs Amazon Cognito sont le composant du serveur d'autorisation des deux méthodes d'authentification et d'autorisation des groupes d'utilisateurs. L'autre méthode est l'[API des groupes d'utilisateurs](#).

Application confidentielle, application côté serveur

Application à laquelle les utilisateurs se connectent à distance, avec du code sur un serveur d'applications et un accès à des secrets. Il s'agit généralement d'une application Web.

Identity provider (IdP) (Fournisseur d'identité)

Service qui enregistre et vérifie l'identité des utilisateurs. Amazon Cognito peut demander l'authentification à des [fournisseurs externes](#) et être un IdP pour les applications.

Jeton Web JSON (JWT)

Document au format JSON contenant des allégations concernant un utilisateur authentifié. Les jetons d'identification authentifient les utilisateurs, les jetons d'accès les autorisent et les jetons d'actualisation mettent à jour les informations d'identification. Amazon Cognito reçoit des jetons de [fournisseurs externes](#) et émet des jetons vers des applications ou. AWS STS

Authentification multifactorielle (MFA)

Obligation pour les utilisateurs de fournir une authentification supplémentaire après avoir fourni leur nom d'utilisateur et leur mot de passe. [Les groupes d'utilisateurs Amazon Cognito disposent de fonctionnalités MFA pour les utilisateurs locaux.](#)

fournisseur (social) OAuth 2.0

Un IdP vers un groupe d'utilisateurs ou un pool d'identités qui fournit un accès [JWT](#) et des jetons d'actualisation. Les groupes d'utilisateurs Amazon Cognito automatisent les interactions avec les fournisseurs sociaux une fois que les utilisateurs s'authentifient.

Fournisseur OpenID Connect (OIDC)

Un IdP vers un groupe d'utilisateurs ou un pool d'identités qui étend la spécification [OAuth](#) pour fournir des jetons d'identification. Les groupes d'utilisateurs Amazon Cognito automatisent les interactions avec les fournisseurs OIDC après l'authentification des utilisateurs.

Application publique

Application autonome sur un appareil, avec du code stocké localement et sans accès aux secrets. Il s'agit généralement d'une application mobile.

Serveur de ressources

Une API avec contrôle d'accès. Les groupes d'utilisateurs Amazon Cognito utilisent également le serveur de ressources pour décrire le composant qui définit la configuration pour interagir avec une API.

Contrôle d'accès basé sur les rôles (RBAC)

Modèle qui accorde l'accès en fonction de la désignation fonctionnelle de l'utilisateur. Les pools d'identités Amazon Cognito implémentent le RBAC en différenciant les rôles IAM.

Prestataire de services (SP), partie utilisatrice (RP)

Une application qui s'appuie sur un IdP pour affirmer que les utilisateurs sont dignes de confiance. Amazon Cognito agit en tant que SP pour les SP externes IdPs et en tant qu'IdP pour les SP basés sur des applications.

Fournisseur SAML

Un IdP vers un groupe d'utilisateurs ou un pool d'identités qui génère des documents d'assertion signés numériquement que votre utilisateur transmet à Amazon Cognito.

Identifiant unique universel (UUID)

Étiquette de 128 bits appliquée à un objet. Les UUID Amazon Cognito sont uniques par groupe d'utilisateurs ou par groupe d'identités.

Annuaire des utilisateurs

Ensemble d'utilisateurs et de leurs attributs qui transmet ces informations à d'autres systèmes. Les groupes d'utilisateurs Amazon Cognito sont des annuaires d'utilisateurs, ainsi que des outils de consolidation des utilisateurs provenant d'annuaires d'utilisateurs externes.

Groupes d'utilisateurs

Lorsque vous voyez les termes figurant dans la liste suivante de ce guide, ils font référence à une fonctionnalité ou à une configuration spécifique des groupes d'utilisateurs.

API des groupes d'utilisateurs Amazon Cognito

Ensemble d'opérations d'API d'authentification et d'autorisation que vous pouvez ajouter à votre application à l'aide d'un AWS SDK. L'API peut connecter les [utilisateurs locaux et les utilisateurs liés](#).

Authentification adaptative

Fonctionnalité de [sécurité avancée](#) qui détecte les activités malveillantes potentielles et applique une sécurité supplémentaire aux [profils utilisateur](#).

Fonctionnalités de sécurité avancées

Composant optionnel qui ajoute des outils pour la sécurité des utilisateurs.

Client d'application

Composant qui définit les paramètres d'un groupe d'utilisateurs en tant qu'IdP pour une application.

URL de rappel, URI de redirection

Un paramètre dans un [client d'application](#) et un paramètre dans les demandes adressées aux [points de terminaison de la fédération](#) des groupes d'utilisateurs. [L'URL de rappel est la destination initiale des utilisateurs authentifiés de votre application.](#)

Informations d'identification compromises

Fonctionnalité de [sécurité avancée](#) qui détecte les mots de passe utilisateur que les attaquants pourraient connaître et applique une sécurité supplémentaire aux [profils utilisateur](#).

Confirmation

Processus qui détermine que les conditions préalables sont remplies pour permettre à un nouvel utilisateur de se connecter. La confirmation se fait généralement par le biais de la [vérification de l'adresse e-mail ou du numéro de téléphone](#).

Authentification personnalisée

Une extension des processus d'authentification avec des [déclencheurs Lambda](#) qui définissent des défis et des réponses supplémentaires pour les utilisateurs.

Authentification des appareils

Processus d'authentification qui remplace le [MFA](#) par une connexion utilisant l'identifiant d'un appareil fiable.

Fournisseur externe, fournisseur tiers

Un IdP qui entretient une relation de confiance avec un groupe d'utilisateurs.

Utilisateur fédéré

Utilisateur d'un groupe d'utilisateurs authentifié par un [fournisseur externe](#).

Points de terminaison de fédération

Ensemble de pages Web sur le [domaine de votre groupe d'utilisateurs](#) hébergeant des services d'interaction avec IdPs et des applications.

Interface utilisateur hébergée

Ensemble de pages Web interactives sur le [domaine de votre groupe d'utilisateurs](#) hébergeant des services d'authentification utilisateur.

Déclencheur Lambda

Fonction AWS Lambda qu'un groupe d'utilisateurs peut invoquer automatiquement à des moments clés des processus d'authentification des utilisateurs. Vous pouvez utiliser des déclencheurs Lambda pour personnaliser les résultats de l'authentification.

Utilisateur local

Un [profil utilisateur](#) dans le [répertoire des utilisateurs du groupe d'utilisateurs](#) qui n'a pas été créé par authentification auprès d'un [fournisseur externe](#).

Utilisateur lié

Utilisateur d'un [fournisseur externe](#) dont l'identité est fusionnée avec celle d'un [utilisateur local](#).

Personnalisation des jetons

Résultat d'un [déclencheur Lambda](#) avant la génération du jeton qui modifie l'identifiant ou le jeton d'accès d'un utilisateur lors de l'exécution.

Groupe d'utilisateurs, fournisseur d'identité Amazon Cognito **cognito-idp**, groupes d'utilisateurs Amazon Cognito

Une AWS ressource avec des services d'authentification et d'autorisation pour les applications qui fonctionnent avec OIDC IdPs.

Domaine du pool d'utilisateurs

Nom de site Web que vous ajoutez à un groupe d'utilisateurs. Le domaine est l'URL de base pour l'[interface utilisateur hébergée](#) et les [points de terminaison de fédération](#).

Vérification

Processus permettant de confirmer qu'un utilisateur possède une adresse e-mail ou un numéro de téléphone. Un groupe d'utilisateurs envoie un code à un utilisateur qui a saisi une nouvelle adresse e-mail ou un nouveau numéro de téléphone. Lorsqu'ils soumettent le code à Amazon Cognito, ils vérifient qu'ils sont propriétaires de la destination du message et peuvent recevoir des messages supplémentaires de la part du groupe d'utilisateurs. Voir également la [confirmation](#).

Profil utilisateur, compte utilisateur

Entrée pour un utilisateur dans le [répertoire des utilisateurs](#). Tous les utilisateurs ont un profil dans leur groupe d'utilisateurs.

Réserves d'identités

Lorsque vous voyez les termes figurant dans la liste suivante de ce guide, ils font référence à une fonctionnalité ou à une configuration spécifique des pools d'identités.

Attributs pour le contrôle d'accès

Implémentation du [contrôle d'accès basé sur les attributs dans les pools](#) d'identités. Les pools d'identités appliquent les attributs utilisateur sous forme de balises aux informations d'identification des utilisateurs.

Authentification de base (classique)

Processus d'authentification dans le cadre duquel vous pouvez personnaliser la demande [d'informations d'identification utilisateur](#).

Identités authentifiées par le développeur

Processus d'authentification qui autorise les informations d'identification des [utilisateurs du pool d'identités avec les informations d'identification du développeur](#).

Informations d'identification du développeur

Les clés d'API IAM d'un administrateur de pool d'identités.

Authentification améliorée

Flux d'authentification qui sélectionne un rôle IAM et applique des balises principales conformément à la logique que vous définissez dans votre pool d'identités.

Identity

[UUID](#) qui lie un utilisateur de l'application et ses [informations d'identification](#) à son profil dans un [annuaire d'utilisateurs](#) externe qui entretient une relation de confiance avec un pool d'identités.

pool d'identités, identités fédérées Amazon Cognito, identité Amazon Cognito, **cognito-identity**

AWS Ressource proposant des services d'authentification et d'autorisation pour les applications utilisant des [AWS informations d'identification temporaires](#).

Identité non authentifiée

Utilisateur qui ne s'est pas connecté avec un IdP de pool d'identités. Vous pouvez autoriser les utilisateurs à générer des informations d'identification utilisateur limitées pour un seul rôle IAM avant de s'authentifier.

Informations d'identification utilisateur

Clés AWS d'API temporaires que les utilisateurs reçoivent après l'authentification du pool d'identités.

Utilisation de ce service avec un AWS SDK

AWS des kits de développement logiciel (SDK) sont disponibles pour de nombreux langages de programmation populaires. Chaque SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK for C++	AWS SDK for C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK for Go	AWS SDK for Go exemples de code
AWS SDK for Java	AWS SDK for Java exemples de code
AWS SDK for JavaScript	AWS SDK for JavaScript exemples de code
Kit AWS SDK pour Kotlin	Kit AWS SDK pour Kotlin exemples de code
AWS SDK for .NET	AWS SDK for .NET exemples de code
AWS SDK for PHP	AWS SDK for PHP exemples de code
AWS Tools for PowerShell	Outils pour des exemples PowerShell de code
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemples de code
AWS SDK for Ruby	AWS SDK for Ruby exemples de code
Kit AWS SDK pour Rust	Kit AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
Kit AWS SDK pour Swift	Kit AWS SDK pour Swift exemples de code

Exemple de disponibilité

Vous n'avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien [Faire un commentaire](#) en bas de cette page.

Commencer avec AWS

Avant de commencer à travailler avec Amazon Cognito, configurez certaines des ressources nécessaires. [AWS](#)

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

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

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

Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer [les tâches nécessitant un accès utilisateur root](#).

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

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez l'utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en [AWS Management Console](#) tant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez [Connexion en tant qu'utilisateur racine](#) dans le Guide de l'utilisateur Connexion à AWS .

2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

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

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

Pour obtenir des instructions, consultez [Activation d' AWS IAM Identity Center](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Dans IAM Identity Center, accordez un accès administratif à un utilisateur.

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

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

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

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

Attribuer l'accès à des utilisateurs supplémentaires

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

Pour obtenir des instructions, voir [Création d'un ensemble d'autorisations](#) dans le guide de AWS IAM Identity Center l'utilisateur.

2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

Pour obtenir des instructions, voir [Ajouter des groupes](#) dans le guide de AWS IAM Identity Center l'utilisateur.

Démarrage avec les groupes d'utilisateurs

Vous pouvez utiliser les guides de cette section pour créer les ressources de votre groupe d'utilisateurs initial. Pour une step-by-step présentation pas à pas, commencez par une [application Web de base dans l'environnement de JavaScript développement React](#). À partir de là, vous pouvez continuer à ajouter des fonctionnalités telles que l'[interface utilisateur hébergée \(interface utilisateur hébergée\)](#) et la connexion fédérée avec des fournisseurs d'[identité](#) externes ou [SAML 2.0](#) (). IdPs

Alors que vous vous efforcez d'étendre votre ensemble de fonctionnalités et d'intégrer davantage de composants d'Amazon Cognito, lisez le chapitre sur les groupes [d'utilisateurs Amazon Cognito](#) pour obtenir une description complète de tout ce que vous pouvez faire avec les groupes d'utilisateurs.

L'exemple de groupe d'utilisateurs et d'application présenté dans cette section illustre une intégration de base des ressources de l'application avec les groupes d'utilisateurs Amazon Cognito. Plus tard, vous pourrez ajuster votre groupe d'utilisateurs pour utiliser davantage d'options à votre disposition. Vous pouvez ensuite mettre à jour votre application pour adopter de nouvelles API et interagir avec l'interface utilisateur hébergée et IdPs.

Le didacticiel présenté dans cette section crée une application dotée d'une interface utilisateur personnalisée et d'une authentification basée sur une API à l'aide d'un AWS SDK. Les applications que vous créez de cette manière sont idéales pour authentifier les [utilisateurs locaux](#). Pour commencer avec une application dotée d'une interface utilisateur prédéfinie, de la gestion automatique de certaines fonctionnalités du groupe d'utilisateurs et de l'authentification des [utilisateurs fédérés](#), passez directement à [Ajouter un client d'application avec l'interface utilisateur hébergée](#)

Rubriques

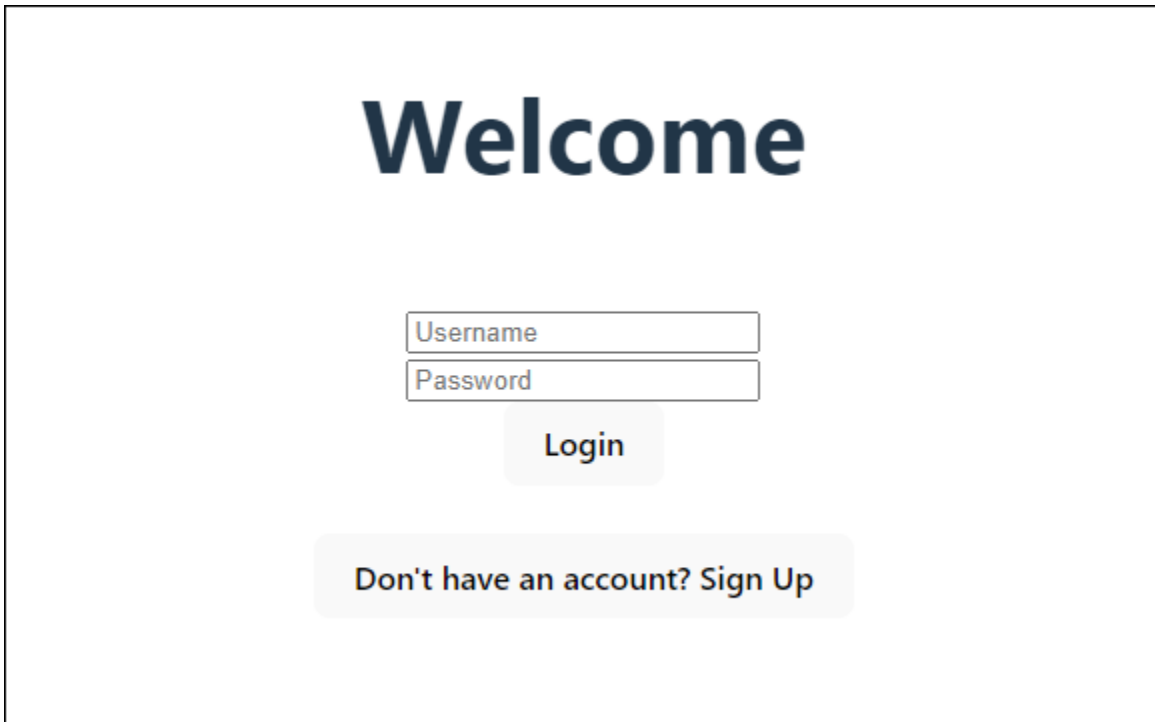
- [Configurer un exemple d'application d'une seule page React](#)
- [Configurer un exemple d'application Android avec Flutter](#)
- [Étapes suivantes](#)

Configurer un exemple d'application d'une seule page React

Dans ce didacticiel, vous allez créer une application React d'une seule page dans laquelle vous pourrez tester l'inscription, la confirmation et la connexion des utilisateurs. React est une bibliothèque JavaScript basée sur le Web et les applications mobiles, axée sur l'interface utilisateur (UI). Cet exemple d'application présente certaines fonctions de base des groupes d'utilisateurs Amazon

Cognito. Si vous avez déjà de l'expérience dans le développement d'applications Web avec React, [téléchargez l'exemple d'application sur GitHub](#).

La capture d'écran suivante montre la page d'authentification initiale de l'application que vous allez créer.



The screenshot shows a login interface. At the top, the word "Welcome" is displayed in a large, bold, dark blue font. Below it, there are two input fields: "Username" and "Password", both with light gray borders. Underneath the "Password" field is a light gray button with the text "Login" in a bold, dark gray font. At the bottom of the form area, there is a light gray button with the text "Don't have an account? Sign Up" in a bold, dark gray font.

La procédure de [création d'un groupe d'utilisateurs](#) vous permet de configurer un groupe d'utilisateurs qui fonctionne avec l'exemple d'application. Vous pouvez ignorer cette étape si votre groupe d'utilisateurs répond aux exigences suivantes :

- Les utilisateurs peuvent se connecter avec leur adresse e-mail. Options de connexion au groupe d'utilisateurs de Cognito : e-mail.
- Les noms d'utilisateur ne distinguent pas les majuscules et minuscules. Exigences relatives au nom d'utilisateur : l'option Mettre le nom d'utilisateur en majuscules et minuscules n'est pas sélectionnée.
- L'authentification multifactorielle (MFA) n'est pas requise. Application de la MFA : MFA optionnelle.
- Votre groupe d'utilisateurs vérifie les attributs pour la confirmation du profil utilisateur par e-mail. Attributs à vérifier : envoyer un message électronique, vérifier l'adresse e-mail.
- L'adresse e-mail est le seul attribut obligatoire. Attributs obligatoires : e-mail.
- Les utilisateurs peuvent s'inscrire eux-mêmes dans votre groupe d'utilisateurs. Auto-enregistrement : l'option Activer l'auto-enregistrement est sélectionnée.


- Le client d'application initial est un client public qui permet de se connecter avec un nom d'utilisateur et un mot de passe. Type d'application : client public, flux d'authentification :ALLOW_USER_PASSWORD_AUTH.

Créer un groupe d'utilisateurs

Création d'un nouveau groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Cliquez sur le bouton Créer un groupe d'utilisateurs. Vous devrez peut-être sélectionner les groupes d'utilisateurs dans le volet de navigation de gauche pour afficher cette option.
3. Dans le coin supérieur droit de la page, choisissez Créer un groupe d'utilisateurs pour lancer l'assistant de création de groupe d'utilisateurs.
4. Dans Configurer l'expérience de connexion, vous pouvez choisir les fournisseurs d'identité (IdPs) que vous utiliserez avec ce groupe d'utilisateurs. Pour plus d'informations, consultez [Ajout de la connexion du groupe d'utilisateurs via un tiers](#).
 - a. Sous Fournisseurs d'authentification, pour les types de fournisseurs, assurez-vous que seul le groupe d'utilisateurs Cognito est sélectionné.
 - b. Pour les options de connexion au groupe d'utilisateurs de Cognito, sélectionnez Nom d'utilisateur. Ne sélectionnez aucune exigence supplémentaire en matière de nom d'utilisateur.
 - c. Conservez toutes les autres options par défaut et choisissez Next.
5. Dans Configurer les exigences de sécurité, vous pouvez choisir votre politique de mot de passe, vos exigences en matière d'authentification multifactorielle (MFA) et les options de restauration du compte utilisateur. Pour plus d'informations, consultez [Utiliser les fonctions de sécurité des groupes d'utilisateurs Amazon Cognito](#).
 - a. Pour la politique de mot de passe, vérifiez que le mode de politique de mot de passe est défini sur les valeurs par défaut de Cognito.
 - b. Sous Authentification multifactorielle, pour l'application de l'authentification MFA, sélectionnez MFA optionnel.
 - c. Pour les méthodes MFA, choisissez les applications d'authentification et les messages SMS.

- d. Pour la restauration du compte utilisateur, vérifiez que l'option Activer la restauration du compte en libre-service est sélectionnée et que le mode de livraison des messages de récupération du compte utilisateur est défini sur E-mail uniquement.
 - e. Conservez toutes les autres options par défaut et choisissez Next.
6. Dans Configurer l'expérience d'inscription, vous pouvez déterminer comment les nouveaux utilisateurs vérifieront leur identité lors de leur inscription en tant que nouvel utilisateur, et quels attributs doivent être obligatoires ou facultatifs lors du processus d'inscription des utilisateurs. Pour plus d'informations, consultez [Gestion des utilisateurs dans votre groupe d'utilisateurs](#).
- a. Vérifiez que l'option Activer l'enregistrement automatique est sélectionnée. Ce paramètre ouvre votre groupe d'utilisateurs afin que n'importe qui puisse s'inscrire sur Internet. Ceci est destiné aux besoins de l'exemple d'application, mais appliquez ce paramètre avec prudence dans les environnements de production.
 - b. Sous Vérification et confirmation assistées par Cognito, vérifiez que la case Autoriser Cognito à envoyer automatiquement des messages pour vérifier et confirmer est cochée.
 - c. Vérifiez que les attributs à vérifier sont définis sur Envoyer un message électronique, vérifier l'adresse e-mail.
 - d. Sous Vérification des modifications d'attributs, vérifiez que les options par défaut sont sélectionnées : Conserver la valeur d'attribut d'origine lorsqu'une mise à jour est en attente est sélectionnée, et les valeurs d'attribut actives lorsqu'une mise à jour est en attente sont définies sur Adresse e-mail.
 - e. Sous Attributs obligatoires, vérifiez que les attributs obligatoires basés sur les sélections précédentes affichent le courrier électronique.

 Important

Pour cet exemple d'application, votre groupe d'utilisateurs ne doit pas définir phone_number comme attribut obligatoire. Si phone_number est affiché comme attribut obligatoire, passez en revue et mettez à jour vos choix précédents :

- MFA optionnel, e-mail uniquement pour le mode de livraison des messages de récupération du compte utilisateur
- Envoyer un message électronique, vérifier l'adresse e-mail pour les attributs à vérifier

- f. Conservez toutes les autres options par défaut et choisissez Next.

7. Dans Configurer la livraison des messages, vous pouvez configurer l'intégration avec Amazon Simple Email Service et Amazon Simple Notification Service pour envoyer des e-mails et des SMS à vos utilisateurs pour l'inscription, la confirmation du compte, le MFA et le rétablissement du compte. Pour plus d'informations, consultez [Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito](#) et [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#).
 - a. Pour le fournisseur de messagerie, choisissez Envoyer un e-mail avec Cognito et utilisez l'expéditeur d'e-mail par défaut fourni par Amazon Cognito. Ce paramètre pour un faible volume d'e-mails est suffisant pour tester les applications. Vous pouvez revenir après avoir vérifié une adresse e-mail auprès d'Amazon Simple Email Service (Amazon SES) et choisi Envoyer un e-mail avec Amazon SES.
 - b. Pour les SMS, sélectionnez Créer un nouveau rôle IAM et entrez un nom de rôle IAM. Cela crée un rôle qui autorise Amazon Cognito à envoyer des SMS.
 - c. Conservez toutes les autres options par défaut et choisissez Next.
8. Dans Intégrer votre application, vous pouvez nommer votre groupe d'utilisateurs, configurer l'interface utilisateur hébergée et créer un client d'application. Pour plus d'informations, consultez [Ajouter un client d'application avec l'interface utilisateur hébergée](#). Les exemples d'applications n'utilisent pas l'interface utilisateur hébergée.
 - a. Sous Nom du groupe d'utilisateurs, entrez un nom de groupe d'utilisateurs.
 - b. Ne sélectionnez pas Utiliser l'interface utilisateur hébergée par Cognito.
 - c. Sous Client d'application initial, vérifiez que le type d'application est défini sur Client public.
 - d. Sous Secret client, vérifiez que l'option Ne pas générer de secret client est sélectionnée.
 - e. Saisissez un nom de client d'application.
 - f. Développez les paramètres avancés du client de l'application. Ajoutez ALLOW_USER_PASSWORD_AUTH à la liste des flux d'authentification.
 - g. Conservez toutes les autres options par défaut et choisissez Next.
9. Passez en revue vos choix dans l'écran Révision et création et modifiez les sélections selon vos besoins. Lorsque vous êtes satisfait de la configuration de votre groupe d'utilisateurs, choisissez Create user pool pour continuer.
10. Sur la page Groupes d'utilisateurs, choisissez votre nouveau groupe d'utilisateurs.
11. Sous Vue d'ensemble du groupe d'utilisateurs, notez l'ID de votre groupe d'utilisateurs. Vous fournirez cette chaîne lorsque vous créerez votre exemple d'application.

12. Choisissez l'onglet Intégration des applications et localisez la section Clients et analyses des applications. Sélectionnez votre nouveau client d'application. Notez votre numéro de client.

Ressources connexes

- [Groupes d'utilisateurs Amazon Cognito](#)
- [Flux d'authentification de groupe d'utilisateurs](#)
- [Utilisation des jetons avec des groupes d'utilisateurs](#)

Création d'une application

Pour créer cette application, vous devez configurer un environnement de développement. Les exigences relatives à l'environnement du développeur sont les suivantes :

1. Node.js est installé et mis à jour.
2. Le gestionnaire de packages de nœuds (npm) est installé et mis à jour vers au moins la version 10.2.3.
3. L'environnement est accessible sur le port TCP 5173 dans un navigateur Web.

Pour créer un exemple d'application Web React

1. Connectez-vous à votre environnement de développement et accédez au répertoire parent de votre application.

```
cd ~/path/to/project/folder/
```

2. Créez un nouveau service React.

```
npm create vite@latest frontend-client -- --template react-ts
```

3. Clonez le [dossier du cognito-developer-guide-react-example projet](#) à partir du référentiel d'exemples de AWS code sur GitHub.

```
cd ~/some/other/path
```

```
git clone https://github.com/awsdocs/aws-doc-sdk-examples.git
```

```
cp -r ./aws-doc-sdk-examples/javascriptv3/example_code/cognito-identity-provider/  
scenarios/cognito-developer-guide-react-example/frontend-client ~/path/to/project/  
folder/frontend-client
```

4. Accédez au `src` répertoire de votre projet.

```
cd ~/path/to/project/folder/frontend-client/src
```

5. Modifiez `config.ts` et remplacez les valeurs suivantes :
 - a. Remplacez `YOUR_AWS_REGION` par un Région AWS code. Par exemple : `us-east-1`.
 - b. `YOUR_COGNITO_USER_POOL_ID` Remplacez-le par l'ID du groupe d'utilisateurs que vous avez désigné pour le test. Par exemple : `us-east-1_EXAMPLE`. Le groupe d'utilisateurs doit être dans celui Région AWS que vous avez saisi à l'étape précédente.
 - c. `YOUR_COGNITO_APP_CLIENT_ID` Remplacez-le par l'ID du client d'application que vous avez désigné pour le test. Par exemple : `1example23456789`. Le client de l'application doit figurer dans le groupe d'utilisateurs de l'étape précédente.
6. Si vous souhaitez accéder à votre exemple d'application à partir d'une adresse IP autre que `localhost`, modifiez `package.json` et remplacez la ligne `"dev": "vite"`, par `"dev": "vite --host 0.0.0.0",`.
7. Installez votre application.

```
npm install
```

8. Lancez l'application.

```
npm run dev
```

9. Accédez à l'application dans un navigateur Web à l'adresse `http://localhost:5173` ou `http://[IP address]:5173`.
10. Inscrivez un nouvel utilisateur avec une adresse e-mail valide.
11. Récupérez le code de confirmation contenu dans votre e-mail. Entrez le code de confirmation dans l'application.
12. Connectez-vous à l'aide de votre nom d'utilisateur et de votre mot de passe.

Création d'un environnement de développement React avec Amazon Lightsail

Pour démarrer rapidement avec cette application, vous pouvez créer un serveur cloud virtuel avec Amazon Lightsail.

Avec Lightsail, vous pouvez créer rapidement une petite instance de serveur préconfigurée avec les prérequis pour cet exemple d'application. Vous pouvez accéder à votre instance par SSH à l'aide d'un client basé sur un navigateur et vous connecter au serveur Web via une adresse IP publique ou privée.

Pour créer une instance Lightsail pour cet exemple d'application

1. Accédez à la console [Lightsail](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Créer une instance.
3. Pour Sélectionner une plate-forme, choisissez Linux/Unix.
4. Pour Sélectionner un plan, choisissez Node.js.
5. Sous Identifiez votre instance, attribuez un nom convivial à votre environnement de développement.
6. Choisissez Créer une instance.
7. Une fois que Lightsail a créé votre instance, sélectionnez-la puis, dans l'onglet Connect, sélectionnez Connect using SSH.
8. Une session SSH s'ouvre dans une fenêtre de navigateur. Exécutez `node -v` et `npm -v` pour confirmer que votre instance a été provisionnée avec Node.js et la version minimale de npm de 10.2.3.
9. Procédez à la [configuration de votre application React](#).

Configurer un exemple d'application Android avec Flutter

Dans ce didacticiel, vous allez créer une application mobile dans Android Studio dans laquelle vous pouvez émuler un appareil et tester l'inscription, la confirmation et la connexion des utilisateurs. Cet exemple d'application crée un client mobile de base pour groupes d'utilisateurs Amazon Cognito pour Android dans Flutter. Si vous avez déjà de l'expérience dans le développement d'applications mobiles avec Flutter, [téléchargez l'exemple d' GitHubapplication sur](#).

La capture d'écran suivante montre l'application s'exécutant sur un appareil Android virtuel.

10:06



DEBUG

Sample Cognito App

Sign-Up

Confirm Sign-Up

Sign-In

Sign Up

Email

Password

Sign Up

La procédure de [création d'un groupe d'utilisateurs](#) vous permet de configurer un groupe d'utilisateurs qui fonctionne avec l'exemple d'application. Vous pouvez ignorer cette étape si votre groupe d'utilisateurs répond aux exigences suivantes :

- Les utilisateurs peuvent se connecter avec leur adresse e-mail. Options de connexion au groupe d'utilisateurs de Cognito : e-mail.
- Les noms d'utilisateur ne distinguent pas les majuscules et minuscules. Exigences relatives au nom d'utilisateur : l'option Mettre le nom d'utilisateur en majuscules et minuscules n'est pas sélectionnée.
- L'authentification multifactorielle (MFA) n'est pas requise. Application de la MFA : MFA optionnelle.
- Votre groupe d'utilisateurs vérifie les attributs pour la confirmation du profil utilisateur par e-mail. Attributs à vérifier : envoyer un message électronique, vérifier l'adresse e-mail.
- L'adresse e-mail est le seul attribut obligatoire. Attributs obligatoires : e-mail.
- Les utilisateurs peuvent s'inscrire eux-mêmes dans votre groupe d'utilisateurs. Auto-enregistrement : l'option Activer l'auto-enregistrement est sélectionnée.
- Le client d'application initial est un client public qui permet de se connecter avec un nom d'utilisateur et un mot de passe. Type d'application : client public, flux d'authentification : ALLOW_USER_PASSWORD_AUTH.

Créer un groupe d'utilisateurs

Création d'un nouveau groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Cliquez sur le bouton Créer un groupe d'utilisateurs. Vous devrez peut-être sélectionner les groupes d'utilisateurs dans le volet de navigation de gauche pour afficher cette option.
3. Dans le coin supérieur droit de la page, choisissez Créer un groupe d'utilisateurs pour lancer l'assistant de création de groupe d'utilisateurs.
4. Dans Configurer l'expérience de connexion, vous pouvez choisir les fournisseurs d'identité (IdPs) que vous utiliserez avec ce groupe d'utilisateurs. Pour plus d'informations, consultez [Ajout de la connexion du groupe d'utilisateurs via un tiers](#).
 - a. Sous Fournisseurs d'authentification, pour les types de fournisseurs, assurez-vous que seul le groupe d'utilisateurs Cognito est sélectionné.

- b. Pour les options de connexion au groupe d'utilisateurs de Cognito, sélectionnez Nom d'utilisateur. Ne sélectionnez aucune exigence supplémentaire en matière de nom d'utilisateur.
 - c. Conservez toutes les autres options par défaut et choisissez Next.
5. Dans Configurer les exigences de sécurité, vous pouvez choisir votre politique de mot de passe, vos exigences en matière d'authentification multifactorielle (MFA) et les options de restauration du compte utilisateur. Pour plus d'informations, consultez [Utiliser les fonctions de sécurité des groupes d'utilisateurs Amazon Cognito](#).
 - a. Pour la politique de mot de passe, vérifiez que le mode de politique de mot de passe est défini sur les valeurs par défaut de Cognito.
 - b. Sous Authentification multifactorielle, pour l'application de l'authentification MFA, sélectionnez MFA optionnel.
 - c. Pour les méthodes MFA, choisissez les applications d'authentification et les messages SMS.
 - d. Pour la restauration du compte utilisateur, vérifiez que l'option Activer la restauration du compte en libre-service est sélectionnée et que le mode de livraison des messages de récupération du compte utilisateur est défini sur E-mail uniquement.
 - e. Conservez toutes les autres options par défaut et choisissez Next.
6. Dans Configurer l'expérience d'inscription, vous pouvez déterminer comment les nouveaux utilisateurs vérifieront leur identité lors de leur inscription en tant que nouvel utilisateur, et quels attributs doivent être obligatoires ou facultatifs lors du processus d'inscription des utilisateurs. Pour plus d'informations, consultez [Gestion des utilisateurs dans votre groupe d'utilisateurs](#).
 - a. Vérifiez que l'option Activer l'enregistrement automatique est sélectionnée. Ce paramètre ouvre votre groupe d'utilisateurs afin que n'importe qui puisse s'inscrire sur Internet. Ceci est destiné aux besoins de l'exemple d'application, mais appliquez ce paramètre avec prudence dans les environnements de production.
 - b. Sous Vérification et confirmation assistées par Cognito, vérifiez que la case Autoriser Cognito à envoyer automatiquement des messages pour vérifier et confirmer est cochée.
 - c. Vérifiez que les attributs à vérifier sont définis sur Envoyer un message électronique, vérifier l'adresse e-mail.
 - d. Sous Vérification des modifications d'attributs, vérifiez que les options par défaut sont sélectionnées : Conserver la valeur d'attribut d'origine lorsqu'une mise à jour est en attente est sélectionnée, et les valeurs d'attribut actives lorsqu'une mise à jour est en attente sont définies sur Adresse e-mail.

- e. Sous Attributs obligatoires, vérifiez que les attributs obligatoires basés sur les sélections précédentes affichent le courrier électronique.

⚠ Important

Pour cet exemple d'application, votre groupe d'utilisateurs ne doit pas définir `phone_number` comme attribut obligatoire. Si `phone_number` est affiché comme attribut obligatoire, passez en revue et mettez à jour vos choix précédents :

- MFA optionnel, e-mail uniquement pour le mode de livraison des messages de récupération du compte utilisateur
- Envoyer un message électronique, vérifier l'adresse e-mail pour les attributs à vérifier

- f. Conservez toutes les autres options par défaut et choisissez Next.
7. Dans Configurer la livraison des messages, vous pouvez configurer l'intégration avec Amazon Simple Email Service et Amazon Simple Notification Service pour envoyer des e-mails et des SMS à vos utilisateurs pour l'inscription, la confirmation du compte, le MFA et le rétablissement du compte. Pour plus d'informations, consultez [Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito](#) et [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#).
 - a. Pour le fournisseur de messagerie, choisissez Envoyer un e-mail avec Cognito et utilisez l'expéditeur d'e-mail par défaut fourni par Amazon Cognito. Ce paramètre pour un faible volume d'e-mails est suffisant pour tester les applications. Vous pouvez revenir après avoir vérifié une adresse e-mail auprès d'Amazon Simple Email Service (Amazon SES) et choisi Envoyer un e-mail avec Amazon SES.
 - b. Pour les SMS, sélectionnez Créer un nouveau rôle IAM et entrez un nom de rôle IAM. Cela crée un rôle qui autorise Amazon Cognito à envoyer des SMS.
 - c. Conservez toutes les autres options par défaut et choisissez Next.
 8. Dans Intégrer votre application, vous pouvez nommer votre groupe d'utilisateurs, configurer l'interface utilisateur hébergée et créer un client d'application. Pour plus d'informations, consultez [Ajouter un client d'application avec l'interface utilisateur hébergée](#). Les exemples d'applications n'utilisent pas l'interface utilisateur hébergée.
 - a. Sous Nom du groupe d'utilisateurs, entrez un nom de groupe d'utilisateurs.
 - b. Ne sélectionnez pas Utiliser l'interface utilisateur hébergée par Cognito.

- c. Sous Client d'application initial, vérifiez que le type d'application est défini sur Client public.
 - d. Sous Secret client, vérifiez que l'option Ne pas générer de secret client est sélectionnée.
 - e. Saisissez un nom de client d'application.
 - f. Développez les paramètres avancés du client de l'application. Ajoutez ALLOW_USER_PASSWORD_AUTH à la liste des flux d'authentification.
 - g. Conservez toutes les autres options par défaut et choisissez Next.
9. Passez en revue vos choix dans l'écran Révision et création et modifiez les sélections selon vos besoins. Lorsque vous êtes satisfait de la configuration de votre groupe d'utilisateurs, choisissez Create user pool pour continuer.
 10. Sur la page Groupes d'utilisateurs, choisissez votre nouveau groupe d'utilisateurs.
 11. Sous Vue d'ensemble du groupe d'utilisateurs, notez l'ID de votre groupe d'utilisateurs. Vous fournirez cette chaîne lorsque vous créerez votre exemple d'application.
 12. Choisissez l'onglet Intégration des applications et localisez la section Clients et analyses des applications. Sélectionnez votre nouveau client d'application. Notez votre numéro de client.

Ressources connexes

- [Groupes d'utilisateurs Amazon Cognito](#)
- [Flux d'authentification de groupe d'utilisateurs](#)
- [Utilisation des jetons avec des groupes d'utilisateurs](#)

Création d'une application

Pour créer un exemple d'application Android

1. Installez le [studio Android](#) et les outils de [ligne de commande](#).
2. Dans Android Studio, installez le [plugin Flutter](#).
3. Créez un nouveau projet Android Studio à partir du contenu du cognito_flutter_mobile_app répertoire de [cet exemple d'application](#).
 - Modifiez assets/config.json et remplacez <<YOUR_USER_POOL_ID>> et <<YOUR_CLIENT_ID>> par les identifiants [du groupe d'utilisateurs et du client d'application que vous avez créés précédemment](#).
4. Installez [Flutter](#).

- a. Ajoutez Flutter à votre variable PATH.
- b. Acceptez les licences à l'aide de la commande suivante.

```
flutter doctor --android-licenses
```

- c. Vérifiez votre environnement Flutter et installez les composants manquants.

```
flutter doctor
```

- Si des composants sont manquants, lancez-vous `flutter doctor -v` pour savoir comment résoudre le problème.

- d. Accédez au répertoire de votre nouveau projet Flutter et installez les dépendances.

- Exécutez `flutter pub add amazon_cognito_identity_dart_2`.

- e. Exécutez `flutter pub add flutter_secure_storage`.

5. Créez un appareil Android virtuel.

1. Dans l'interface graphique d'Android Studio, créez un nouvel appareil à l'aide du [gestionnaire de périphériques](#).

2. Dans la CLI, exécutez `flutter emulators --create --name android-device`.

6. Lancez votre appareil Android virtuel.

1. Dans l'interface graphique d'Android Studio, sélectionnez



l'icône de démarrage à côté de votre appareil virtuel.

icône

2. Dans la CLI, exécutez `flutter emulators --launch android-device`.

7. Lancez votre application sur votre appareil virtuel.

1. Dans l'interface graphique d'Android Studio, sélectionnez



l'icône de déploiement.

icône

2. Dans la CLI, exécutez `flutter run`.

8. Accédez à votre appareil virtuel en cours d'exécution dans Android Studio.

9. Inscrivez un nouvel utilisateur avec une adresse e-mail valide.

10. Récupérez le code de confirmation contenu dans votre e-mail. Entrez le code de confirmation dans l'application.
11. Connectez-vous à l'aide de votre nom d'utilisateur et de votre mot de passe.

Étapes suivantes

Après avoir suivi les didacticiels pour réaliser des exemples d'applications, vous pouvez élargir le champ de mise en œuvre de votre groupe d'utilisateurs. Vous pouvez [créer des groupes d'utilisateurs supplémentaires](#), [personnaliser les fonctionnalités des groupes d'utilisateurs pour d'autres applications](#) ou [ajouter des fournisseurs d'identité externes](#). Lorsque vous planifiez votre migration vers l'intégration des groupes d'utilisateurs Amazon Cognito dans les applications de production, vous pouvez évaluer des [exemples et des didacticiels supplémentaires](#).

Voici quelques fonctionnalités supplémentaires relatives aux groupes d'utilisateurs Amazon Cognito :

- [Personnalisation des pages web intégrées d'inscription et de connexion](#)
- [Ajout de l'authentification MFA à un groupe d'utilisateurs](#)
- [Ajout de la sécurité avancée à un groupe d'utilisateurs](#)
- [Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda](#)
- [Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito](#)

Pour une présentation des modèles d'authentification et d'autorisation Amazon Cognito, consultez [Comment fonctionne l'authentification avec les groupes d'utilisateurs et les groupes d'identités Amazon Cognito](#)

Pour accéder aux autres utilisateurs Services AWS après une authentification réussie du groupe d'utilisateurs, consultez [Accès à Services AWS l'aide d'un pool d'identités après la connexion](#).

Outre les SDK AWS Management Console et les kits de développement du groupe d'utilisateurs, vous pouvez également gérer vos groupes d'utilisateurs à l'aide du [AWS Command Line Interface](#).

Rubriques

- [Création d'un nouveau groupe d'utilisateurs](#)
- [Ajouter un client d'application avec l'interface utilisateur hébergée](#)
- [Ajouter la connexion par réseaux sociaux à un groupe d'utilisateurs \(Facultatif\)](#)
- [Ajouter la connexion à un groupe d'utilisateurs avec un fournisseur d'identité SAML \(Facultatif\)](#)


Création d'un nouveau groupe d'utilisateurs

Avec un groupe d'utilisateurs, vos utilisateurs peuvent se connecter à votre application web ou mobile via Amazon Cognito.

Création d'un nouveau groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Cliquez sur le bouton Créer un groupe d'utilisateurs. Vous devrez peut-être sélectionner les groupes d'utilisateurs dans le volet de navigation de gauche pour afficher cette option.
3. Dans le coin supérieur droit de la page, choisissez Créer un groupe d'utilisateurs pour lancer l'assistant de création de groupe d'utilisateurs.
4. Dans Configurer l'expérience de connexion, vous pouvez choisir les fournisseurs d'identité (IdPs) que vous utiliserez avec ce groupe d'utilisateurs. Pour plus d'informations, consultez [Ajout de la connexion du groupe d'utilisateurs via un tiers](#).
 - a. Sous Fournisseurs d'authentification, pour les types de fournisseurs, assurez-vous que seul le groupe d'utilisateurs Cognito est sélectionné.
 - b. Pour les options de connexion au groupe d'utilisateurs de Cognito, sélectionnez Nom d'utilisateur. Ne sélectionnez aucune exigence supplémentaire en matière de nom d'utilisateur.
 - c. Conservez toutes les autres options par défaut et choisissez Next.
5. Dans Configurer les exigences de sécurité, vous pouvez choisir votre politique de mot de passe, vos exigences en matière d'authentification multifactorielle (MFA) et les options de restauration du compte utilisateur. Pour plus d'informations, consultez [Utiliser les fonctions de sécurité des groupes d'utilisateurs Amazon Cognito](#).
 - a. Pour la politique de mot de passe, vérifiez que le mode de politique de mot de passe est défini sur les valeurs par défaut de Cognito.
 - b. Sous Authentification multifactorielle, pour l'application de l'authentification MFA, sélectionnez MFA optionnel.
 - c. Pour les méthodes MFA, choisissez les applications d'authentification et les messages SMS.
 - d. Pour la restauration du compte utilisateur, vérifiez que l'option Activer la restauration du compte en libre-service est sélectionnée et que le mode de livraison des messages de récupération du compte utilisateur est défini sur E-mail uniquement.

- e. Conservez toutes les autres options par défaut et choisissez Next.
6. Dans Configurer l'expérience d'inscription, vous pouvez déterminer comment les nouveaux utilisateurs vérifieront leur identité lors de leur inscription en tant que nouvel utilisateur, et quels attributs doivent être obligatoires ou facultatifs lors du processus d'inscription des utilisateurs. Pour plus d'informations, consultez [Gestion des utilisateurs dans votre groupe d'utilisateurs](#).
 - a. Vérifiez que l'option Activer l'enregistrement automatique est sélectionnée. Ce paramètre ouvre votre groupe d'utilisateurs afin que n'importe qui puisse s'inscrire sur Internet. Ceci est destiné aux besoins de l'exemple d'application, mais appliquez ce paramètre avec prudence dans les environnements de production.
 - b. Sous Vérification et confirmation assistées par Cognito, vérifiez que la case Autoriser Cognito à envoyer automatiquement des messages pour vérifier et confirmer est cochée.
 - c. Vérifiez que les attributs à vérifier sont définis sur Envoyer un message électronique, vérifier l'adresse e-mail.
 - d. Sous Vérification des modifications d'attributs, vérifiez que les options par défaut sont sélectionnées : Conserver la valeur d'attribut d'origine lorsqu'une mise à jour est en attente est sélectionnée, et les valeurs d'attribut actives lorsqu'une mise à jour est en attente sont définies sur Adresse e-mail.
 - e. Sous Attributs obligatoires, vérifiez que les attributs obligatoires basés sur les sélections précédentes affichent le courrier électronique.

 Important

Pour cet exemple d'application, votre groupe d'utilisateurs ne doit pas définir phone_number comme attribut obligatoire. Si phone_number est affiché comme attribut obligatoire, passez en revue et mettez à jour vos choix précédents :

- MFA optionnel, e-mail uniquement pour le mode de livraison des messages de récupération du compte utilisateur
- Envoyer un message électronique, vérifier l'adresse e-mail pour les attributs à vérifier

- f. Conservez toutes les autres options par défaut et choisissez Next.
7. Dans Configurer la livraison des messages, vous pouvez configurer l'intégration avec Amazon Simple Email Service et Amazon Simple Notification Service pour envoyer des e-mails et des SMS à vos utilisateurs pour l'inscription, la confirmation du compte, le MFA et le rétablissement

du compte. Pour plus d'informations, consultez [Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito](#) et [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#).

- a. Pour le fournisseur de messagerie, choisissez Envoyer un e-mail avec Cognito et utilisez l'expéditeur d'e-mail par défaut fourni par Amazon Cognito. Ce paramètre pour un faible volume d'e-mails est suffisant pour tester les applications. Vous pouvez revenir après avoir vérifié une adresse e-mail auprès d'Amazon Simple Email Service (Amazon SES) et choisi Envoyer un e-mail avec Amazon SES.
 - b. Pour les SMS, sélectionnez Créer un nouveau rôle IAM et entrez un nom de rôle IAM. Cela crée un rôle qui autorise Amazon Cognito à envoyer des SMS.
 - c. Conservez toutes les autres options par défaut et choisissez Next.
8. Dans Intégrer votre application, vous pouvez nommer votre groupe d'utilisateurs, configurer l'interface utilisateur hébergée et créer un client d'application. Pour plus d'informations, consultez [Ajouter un client d'application avec l'interface utilisateur hébergée](#). Les exemples d'applications n'utilisent pas l'interface utilisateur hébergée.
- a. Sous Nom du groupe d'utilisateurs, entrez un nom de groupe d'utilisateurs.
 - b. Ne sélectionnez pas Utiliser l'interface utilisateur hébergée par Cognito.
 - c. Sous Client d'application initial, vérifiez que le type d'application est défini sur Client public.
 - d. Sous Secret client, vérifiez que l'option Ne pas générer de secret client est sélectionnée.
 - e. Saisissez un nom de client d'application.
 - f. Développez les paramètres avancés du client de l'application. Ajoutez ALLOW_USER_PASSWORD_AUTH à la liste des flux d'authentification.
 - g. Conservez toutes les autres options par défaut et choisissez Next.
9. Passez en revue vos choix dans l'écran Révision et création et modifiez les sélections selon vos besoins. Lorsque vous êtes satisfait de la configuration de votre groupe d'utilisateurs, choisissez Create user pool pour continuer.
10. Sur la page Groupes d'utilisateurs, choisissez votre nouveau groupe d'utilisateurs.
11. Sous Vue d'ensemble du groupe d'utilisateurs, notez l'ID de votre groupe d'utilisateurs. Vous fournirez cette chaîne lorsque vous créerez votre exemple d'application.
12. Choisissez l'onglet Intégration des applications et localisez la section Clients et analyses des applications. Sélectionnez votre nouveau client d'application. Notez votre numéro de client.

Pour créer un groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Dans l'angle supérieur droit de la page, choisissez Créer un groupe d'utilisateurs pour lancer l'assistant de création de groupe d'utilisateurs.
4. Dans la section Configurer l'expérience de connexion, choisissez les fournisseurs fédérés que vous utiliserez avec ce groupe d'utilisateurs. Pour plus d'informations, consultez [Ajout de la connexion du groupe d'utilisateurs via un tiers](#).
5. Dans Configurer les exigences de sécurité, choisissez votre politique de mot de passe, vos exigences d'authentification multi-facteur (MFA) et vos options de récupération de compte d'utilisateur. Pour plus d'informations, consultez [Utiliser les fonctions de sécurité des groupes d'utilisateurs Amazon Cognito](#).
6. Dans la section Configurer l'expérience d'inscription, déterminez comment les nouveaux utilisateurs vérifieront leur identité lors de leur inscription et quels attributs seront requis ou facultatifs pendant le flux d'inscription de l'utilisateur. Pour plus d'informations, consultez [Gestion des utilisateurs dans votre groupe d'utilisateurs](#).

Important

Si vous activez l'inscription des utilisateurs dans votre groupe d'utilisateurs, n'importe qui sur Internet peut créer un compte et se connecter à vos applications. N'activez pas l'auto-inscription dans votre groupe d'utilisateurs, sauf si vous souhaitez ouvrir votre application à des inscriptions publiques. Pour modifier ce paramètre, mettez à jour l'inscription en libre-service dans l'onglet Expérience d'inscription de la console du pool d'utilisateurs, ou mettez à jour la valeur de [AllowAdminCreateUserOnly](#) dans une demande d'API [CreateUserPool](#). [UpdateUserPool](#)

Pour plus d'informations sur les fonctionnalités de sécurité que vous pouvez configurer dans vos groupes d'utilisateurs, consultez [Utiliser les fonctions de sécurité des groupes d'utilisateurs Amazon Cognito](#).

7. Dans Configurer la diffusion des messages, configurez l'intégration avec Amazon Simple Email Service et Amazon Simple Notification Service pour envoyer des e-mails et des SMS à vos utilisateurs pour inscription, confirmation de compte, MFA et récupération de compte. Pour plus

d'informations, consultez [Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito](#) et [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#).

8. Dans Intégrez votre application, nommez votre groupe d'utilisateurs, configurez l'interface utilisateur hébergée et créez un client d'application. Pour plus d'informations, consultez [Ajouter un client d'application avec l'interface utilisateur hébergée](#).
9. Passez en revue vos choix dans l'écran Révision et création et modifiez les sélections selon vos besoins. Lorsque vous êtes satisfait de la configuration de votre groupe d'utilisateurs, sélectionnez Créer un groupe d'utilisateurs pour continuer.

Ressources connexes

Pour plus d'informations sur les groupes d'utilisateurs, consultez [Groupes d'utilisateurs Amazon Cognito](#).

Voir également : [Flux d'authentification de groupe d'utilisateurs](#) et [Utilisation des jetons avec des groupes d'utilisateurs](#).

Ajouter un client d'application avec l'interface utilisateur hébergée

Après avoir créé un groupe d'utilisateurs, vous pouvez créer un [client d'application](#) pour une application qui affiche les pages Web intégrées de l'interface utilisateur hébergée. Dans l'interface utilisateur hébergée, les utilisateurs peuvent :

- Inscrivez-vous pour obtenir un profil d'utilisateur.
- Connectez-vous auprès d'un fournisseur d'identité tiers.
- Connectez-vous avec ou sans authentification multifactorielle.
- Réinitialisez leur mot de passe.

Pour créer un client d'application pour la connexion à l'interface utilisateur hébergée

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#). Si vous créez un nouveau groupe d'utilisateurs, vous serez invité à configurer un client d'application et à configurer l'interface utilisateur hébergée pendant l'assistant.

4. Choisissez l'onglet App integration (Intégration d'applications) pour votre groupe d'utilisateurs.
5. À côté de Domaine, choisissez Actions, puis sélectionnez l'une de ces options : Créer un nom de domaine personnalisé ou Créer un nom de domaine Amazon Cognito. Si vous avez déjà configuré un domaine de groupe d'utilisateurs, choisissez Supprimer le domaine Amazon Cognito ou Supprimer le domaine personnalisé avant de créer votre nouveau domaine personnalisé.
6. Saisissez un préfixe de domaine disponible à utiliser avec un domaine Amazon Cognito. Pour plus d'informations sur la configuration d'un Custom domain (Domaine personnalisé), consultez [Utilisation de votre propre domaine pour l'interface utilisateur hébergée](#).
7. Choisissez Créer.
8. Naviguez jusqu'à Intégration d'applications pour le même groupe d'utilisateurs et localisez Clients d'application. Choisissez Créer une application client.
9. Choisissez un type d'application. Certains paramètres recommandés seront fournis en fonction de votre sélection. Une application qui utilise l'interface utilisateur hébergée est une Client public.
10. Saisissez un Nom de client d'application.
11. Pour cet exercice, choisissez Ne générez pas de secret client. Le secret client est utilisé par les applications confidentielles qui authentifient les utilisateurs à partir d'une application centralisée. Dans cet exercice, vous présenterez une page de connexion à l'interface utilisateur hébergée à vos utilisateurs et vous n'aurez pas besoin de secret client.
12. Choisissez les flux d'authentification que vous allez autoriser avec votre application. Assurez-vous que USER_SRP_AUTH a été sélectionné.
13. Personnalisez l'expiration du jeton, Configuration de sécurité avancée, et Permissions de lecture et d'écriture d'attributs si nécessaire. Pour plus d'informations, consultez [Configuration des paramètres de l'Application Client](#).
14. Ajouter une URL de rappel pour votre client d'application. C'est là que vous serez dirigé après l'authentification de l'interface utilisateur hébergée. Il n'est pas nécessaire d'ajouter une URL de déconnexion autorisée avant de pouvoir implémenter la déconnexion dans votre application.

Pour une application iOS ou Android, vous pouvez utiliser une URL de rappel telle que `myapp://`.
15. Sélectionnez les Fournisseurs d'identité pour le client d'application. Au minimum, activez Groupe d'utilisateurs Amazon Cognito en tant que fournisseur.

Note

Pour vous connecter auprès de fournisseurs d'identité externes (IdPs) tels que Facebook, Amazon, Google et Apple, ainsi que via OpenID Connect (OIDC) ou SAML IdPs, configurez-les d'abord comme indiqué dans [Ajouter une connexion à un groupe d'utilisateurs](#) via un tiers. Retournez ensuite à la page des paramètres du client de l'application pour les activer.

16. Choisissez Types de subventions OAuth 2.0. Sélectionnez Authorization code grant (Octroi de code d'autorisation) pour renvoyer un code d'autorisation qui est ensuite remplacé par des jetons de groupe d'utilisateurs. Étant donné que les jetons ne sont jamais exposés directement à un utilisateur final, ils sont moins susceptibles d'être compromis. Toutefois, une application personnalisée est requise au niveau du backend pour échanger le code d'autorisation contre des jetons de groupe d'utilisateurs. Pour des raisons de sécurité, nous vous recommandons d'utiliser ce flux d'octroi de code d'autorisation avec une [PKCE \(Proof Key for Code Exchange\)](#) pour les applications mobiles.

Sélectionnez implicit grant (octroi implicite) pour que les jetons web JSON (JSON web tokens, JWT) vous soient renvoyés à partir d'Amazon Cognito. Vous pouvez utiliser ce flux lorsqu'il n'y a pas de backend disponible pour échanger un code d'autorisation contre des jetons. C'est également utile pour déboguer les jetons.

Note

Vous pouvez activer Authorization code grant (Octroi de code d'autorisation) et Implicit code grant (Octroi de code implicite), puis utiliser chaque octroi selon vos besoins. Sélectionnez Informations d'identification du client uniquement si votre application doit demander des jetons d'accès pour elle-même et non pas pour un utilisateur.

17. À moins que vous ne souhaitiez spécifiquement en exclure un, sélectionnez tous les Périmètres OpenID Connect.
18. Sélectionnez les étendues personnalisées que vous avez configurées. Les périmètres personnalisés sont généralement utilisés avec des clients confidentiels.
19. Choisissez Créer.

Pour afficher votre page de connexion

Sur la page client de votre application, sélectionnez Afficher l'interface utilisateur hébergée pour ouvrir un nouvel onglet de navigateur vers une page de connexion préremplie avec l'ID du client de l'application, le champ d'application, l'autorisation et les paramètres de l'URL de rappel.

Vous pouvez afficher manuellement la page Web de connexion de l'interface utilisateur hébergée en utilisant l'URL suivante. Notez le `response_type`. Dans ce cas, `response_type=code` pour l'octroi de code d'autorisation.

```
https://your_domain/login?  
response_type=code&client_id=your_app_client_id&redirect_uri=your_callback_url
```

Vous pouvez afficher la page Web d'interface utilisateur de connexion hébergée à l'URL suivante pour l'octroi de code implicite où `response_type = token`. Après une connexion réussie, Amazon Cognito renvoie des jetons de groupe d'utilisateurs à la barre d'adresse de votre navigateur web.

```
https://your_domain/login?  
response_type=token&client_id=your_app_client_id&redirect_uri=your_callback_url
```

Vous pouvez trouver les jetons d'identité JWT (JSON Web Token) juste après le paramètre `#idtoken=` dans la réponse.

L'URL suivante est un exemple de réponse à une demande de subvention implicite. Votre chaîne de jetons d'identité sera beaucoup plus longue.

```
https://www.example.com/  
#id_token=123456789tokens123456789&expires_in=3600&token_type=Bearer
```

Les jetons de groupe d'utilisateurs Amazon Cognito sont signés à l'aide d'un algorithme RS256. Vous pouvez décoder et vérifier les jetons du groupe d'utilisateurs à l'aide AWS Lambda de. Pour plus d'informations, consultez [Décoder et vérifier les jetons Amazon Cognito JWT](#) sur le site Web. AWS GitHub

Votre domaine s'affiche dans la page Domain name (Nom de domaine). Votre ID de client d'application et votre URL de rappel s'affichent dans la page Paramètres généraux. Si les modifications que vous avez apportées dans la console n'apparaissent pas immédiatement, attendez quelques minutes, puis actualisez votre navigateur.

Ajouter la connexion par réseaux sociaux à un groupe d'utilisateurs (Facultatif)

Vos utilisateurs d'application peuvent se connecter via un fournisseur d'identité sociale comme Facebook, Google, Amazon et Apple. Que vos utilisateurs se connectent directement ou via un tiers, ils ont tous un profil dans le groupe d'utilisateurs. Ignorez cette étape si vous ne souhaitez pas ajouter la connexion via un fournisseur d'identité de connexion par les réseaux sociaux.

Inscription avec un fournisseur d'identité social

Avant de créer un fournisseur d'identité social avec Amazon Cognito, vous devez enregistrer votre application auprès du fournisseur d'identité social pour recevoir un ID client et une clé secrète de client.

Pour enregistrer une application avec Facebook

1. Créez un [compte développeur avec Facebook](#).
2. [Connectez-vous](#) avec vos informations d'identification Facebook.
3. Dans le menu Mes applications, choisissez Créer une nouvelle application.

Si vous n'avez pas d'application Facebook existante, vous verrez une autre option. Sélectionnez Create App (Créer une application).

4. Sur la page Créer une application, sélectionnez un cas d'utilisation pour votre application, puis Suivant.
5. Nommez votre application Facebook, puis sélectionnez Créer une application.
6. Dans la barre de navigation de gauche, sélectionnez Paramètres d'application, puis Basique.
7. Notez l'ID d'app et la Clé secrète d'application. Vous les utiliserez dans la section suivante.
8. Au bas de la page, choisissez + Ajouter une plateforme.
9. Sur l'écran Select Platform, sélectionnez vos plateformes, puis choisissez Next.
10. Sélectionnez Enregistrer les modifications.
11. Pour App Domains (Domaines d'applications), saisissez le domaine de votre groupe d'utilisateurs.

`https://your_user_pool_domain`

12. Sélectionnez Enregistrer les modifications.

13. Dans la barre de navigation, choisissez Produits, puis sélectionnez Configurer depuis Facebook Login.
14. Dans le menu Connexion avec Facebook Configurer, sélectionnez Paramètres.

Saisissez votre URL de redirection dans le champ URL de redirection OAuth valide. L'URL de redirection comprend le domaine de votre groupe d'utilisateurs avec le /oauth2/idpresponse point de terminaison.

```
https://your_user_pool_domain/oauth2/idpresponse
```

15. Sélectionnez Enregistrer les modifications.

Pour enregistrer une application avec Amazon

1. Créez un [compte développeur avec Amazon](#).
2. [Connectez-vous](#) avec vos informations d'identification Amazon.
3. Vous devez créer un profil de sécurité Amazon pour recevoir l'ID client et de la clé secrète de client Amazon.

Choisissez Apps and Services dans la barre de navigation en haut de la page, puis choisissez Login with Amazon.

4. Sélectionnez Create a Security Profile (Créer un profil de sécurité).
5. Saisissez un Nom du profil de sécurité, une Description du profil de sécurité et une URL de consentement à l'avis de confidentialité.
6. Choisissez Save (Enregistrer).
7. Choisissez Client ID (ID client) et Client Secret (Secret client) pour afficher l'ID et le secret client. Vous les utiliserez dans la section suivante.
8. Passez le curseur sur l'engrenage et choisissez l'icône Web Settings (Paramètres web), puis Modifier.
9. Saisissez le domaine de votre groupe d'utilisateurs dans le champ Allowed Origins (Origines autorisées).

```
https://<your-user-pool-domain>
```

10. Saisissez le domaine de votre groupe d'utilisateurs avec le point de terminaison /oauth2/idpresponse dans Allowed Return URLs (URL de retour autorisées).

```
https://<your-user-pool-domain>/oauth2/idpresponse
```

11. Choisissez Enregistrer.

Pour enregistrer une application avec Google

Pour plus d'informations concernant OAuth 2.0 sur la plateforme Google Cloud, consultez [En savoir plus sur l'authentification et l'autorisation](#) (langue Français non garanti) dans la documentation Google Workspace for Developers.

1. Créez un [compte développeur avec Google](#).
2. Connectez-vous à la [console Google Cloud Platform](#).
3. Dans la barre de navigation supérieure, choisissez Select a project (Sélectionner un projet). Si vous avez déjà un projet sur la plateforme Google, ce menu affiche votre projet par défaut.
4. Sélectionnez NEW PROJECT (NOUVEAU PROJET).
5. Saisissez le nom de votre produit, puis choisissez CREATE (CRÉER).
6. Dans la barre de navigation de gauche, choisissez APIs and Services, puis choisissez l'écran de consentement OAuth.
7. Entrez les informations de l'application, le domaine de l'application, les domaines autorisés et les coordonnées du développeur. Vos domaines autorisés doivent inclure `amazoncognito.com` la racine de votre domaine personnalisé. Par exemple : `example.com`. Choisissez SAVE AND CONTINUE (ENREGISTRER ET CONTINUER).
8. 1. Sous Étendue, choisissez Ajouter ou supprimer des étendues, puis choisissez, au minimum, les étendues OAuth suivantes.
 1. `.../auth/userinfo.email`
 2. `.../auth/userinfo.profile`
 3. `openid`
9. Sous Test users (Utilisateurs test), choisissez Add users (Ajouter des utilisateurs). Entrez votre adresse e-mail et tout autre utilisateur de test autorisé, puis choisissez ENREGISTRER ET CONTINUER.
10. Développez à nouveau la barre de navigation de gauche, choisissez APIs and Services, puis Credentials.
11. Choisissez CREATE CREDENTIALS, puis choisissez OAuth client ID.

12. Choisissez un Application type (Type d'application) et donnez à votre client un Name (Nom).
13. Sous JavaScript Origines autorisées, choisissez AJOUTER UN URI. Saisissez le domaine de votre groupe d'utilisateurs.

```
https://<your-user-pool-domain>
```

14. Sous Authorized redirect URIs (URI de redirection autorisés), choisissez ADD URI (AJOUTER UN URI). Entrez le chemin d'accès au point de terminaison /oauth2/idpresponse de votre domaine de groupe d'utilisateurs.

```
https://<your-user-pool-domain>/oauth2/idpresponse
```

15. Choisissez CREATE.
16. Stockez en toute sécurité les valeurs que Google affiche sous Your client ID (ID de votre client) et Your client secret (Secret de votre client). Fournissez ces valeurs à Amazon Cognito lorsque vous ajoutez un fournisseur d'identité Google.

Pour enregistrer une application avec Apple

Pour plus d'informations sur la configuration de la fonctionnalité Se connecter avec Apple, consultez [Configuring Your Environment for Sign in with Apple](#) dans la documentation Apple Developer.

1. Créez un [compte développeur Apple](#).
2. [Connectez-vous](#) avec vos informations d'identification Apple.
3. Dans la barre de navigation de gauche, choisissez Certificates, Identifiers & Profiles (Certificats, identifiants et profils).
4. Dans la barre de navigation de gauche, choisissez Identifiers (Identifiants).
5. Dans la page Identifiers (Identifiants), choisissez l'icône +.
6. Dans la page Register a New Identifier (Enregistrer un nouvel identifiant), choisissez App IDs (ID application), puis Continue (Continuer).
7. Sur la page Sélectionnez un type, choisissez App, puis choisissez Continuer.
8. Dans la page Register an App ID (Enregistrer un ID d'application), procédez comme suit :
 1. Sous Description, entrez une description.
 2. Sous App ID Prefix (Préfixe d'ID d'application), saisissez un Bundle ID (ID de bundle). Notez la valeur sous Make a note of the value under (Préfixe d'ID d'application). Vous utiliserez

cette valeur après avoir choisi Apple comme fournisseur d'identité dans [Étape 2 : Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs](#).

3. Sous Capabilities (Capacités), choisissez Sign In with Apple (Connexion avec Apple), puis Edit (Modifier).
4. Sur la page Sign in with Apple: App ID Configuration (Se connecter avec Apple : Configuration d'ID d'application), choisissez de configurer l'application comme principale ou groupée avec d'autres ID d'application, puis choisissez Save (Enregistrer).
5. Choisissez Continuer.
9. Dans la page Confirm your App ID (Confirmer votre ID d'application), choisissez Register (Inscrire).
10. Dans la page Identifiers (Identifiants), choisissez l'icône +.
11. Dans la page Register a New Identifier (Enregistrer un nouvel identifiant), choisissez Services IDs (ID de service), puis Continue (Continuer).
12. Dans la page Register an App ID (Enregistrer un ID d'application), procédez comme suit :
 1. Sous Description, entrez une description.
 2. Sous Identifiant, saisissez un identifiant. Prenez note de cet identifiant de services, car vous aurez besoin de cette valeur une fois que vous aurez choisi Apple comme fournisseur d'identité dans [Étape 2 : Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs](#).
 3. Choisissez Continue (Continuer), puis Register (S'inscrire).
13. Choisissez l'ID de services que vous venez de créer sur la page Identifiants.
 1. Sélectionnez Sign In with Apple (Connexion avec Apple), puis choisissez Configure (Configurer).
 2. Sur la page Web Authentication Configuration (Configuration de l'authentification web), sélectionnez l'ID d'application que vous avez créé précédemment comme Primary App ID (ID d'application principale).
 3. Choisissez l'icône + en regard de Website URLs (URL de sites web).
 4. Sous Domains and subdomains (Domaines et sous-domaines), entrez le domaine de votre groupe d'utilisateurs sans utiliser de préfixe `https://`.

<your-user-pool-domain>
 5. Sous Return URLs (URL de retour), saisissez le chemin d'accès au point de terminaison `/oauth2/idpresponse` du domaine de votre groupe d'utilisateurs.

```
https://<your-user-pool-domain>/oauth2/idpresponse
```

6. Choisissez Suivant, puis cliquez sur Terminé. Vous n'avez pas besoin de vérifier le domaine.
7. Choisissez Continue (Continuer), puis Save (Enregistrer).
14. Dans la barre de navigation de gauche, choisissez Keys (Clés).
15. Dans la page Keys (Clés), choisissez l'icône +.
16. Dans la page Register a New Key (Enregistrer une nouvelle clé), procédez comme suit :
 1. Sous Key Name (Nom de clé), saisissez un nom de clé.
 2. Sélectionnez Sign In with Apple (Connexion avec Apple), puis choisissez Configure (Configurer).
 3. Sur la page Configurer la clé, sélectionnez l'ID d'application que vous avez créé précédemment comme ID d'application principal. Choisissez Enregistrer.
 4. Choisissez Continue (Continuer), puis Register (Enregistrer).
17. Sur la page Télécharger votre clé, choisissez Télécharger pour télécharger la clé privée, notez l'ID de clé affiché, puis choisissez OK. Vous aurez besoin de cette clé privée et de la valeur Key ID (ID de clé) affichées sur cette page après avoir choisi Apple comme fournisseur d'identité dans [Étape 2 : Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs](#).

Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs

Dans cette section, vous configurez un fournisseur d'identité social dans votre groupe d'utilisateurs à l'aide de l'ID client et de la clé secrète du client de la section précédente.

Pour configurer le fournisseur d'identité sociale d'un groupe d'utilisateurs avec AWS Management Console

1. Accédez à la [console Amazon Cognito](#). Il se peut que vous soyez invité à saisir vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet Expérience de connexion. Localisez la Session fédérée et sélectionnez Ajouter un fournisseur d'identité.
5. Choisissez un fournisseur d'identité social : Facebook, Google, Login with Amazon ou Se connecter sur Apple.

6. Choisissez l'une des étapes suivantes, en fonction du fournisseur d'identité sociale de votre choix :
 - Google et Login with Amazon — Entrez l'identifiant du client de l'application et le code secret du client généré dans la section précédente.
 - Facebook : entrez l'ID client de l'application et le code secret du client de l'application générés dans la section précédente, puis choisissez une version de l'API (par exemple, la version 2.12). Nous vous recommandons de choisir la dernière version possible : chaque API Facebook possède un cycle de vie et une date d'obsolescence. Les périmètres et attributs Facebook peuvent varier d'une version d'API à l'autre. Nous vous recommandons de tester votre connexion d'identité sociale avec Facebook pour vous assurer que la fédération fonctionne comme prévu.
 - Connectez-vous avec Apple : entrez l'identifiant de service, l'identifiant d'équipe, l'identifiant de clé et la clé privée générés dans la section précédente.
7. Entrez les noms des étendues autorisées que vous souhaitez utiliser. Les périmètres définissent les attributs d'utilisateur (tels que `name` et `email`) auxquels vous souhaitez accéder avec votre application. Pour Facebook, ils doivent être séparés par des virgules. Pour Google et Login with Amazon, ils doivent être séparés par des espaces. Pour Sign in with Apple (Connexion avec Apple), activez les cases des périmètres auxquelles vous souhaitez accéder.

Fournisseur d'identité social	Exemple de règles
Facebook	<code>public_profile, email</code>
Google	<code>profile email openid</code>
Login with Amazon	<code>profile postal_code</code>
Se connecter avec Apple	<code>email name</code>

L'utilisateur de l'application est invité à accepter de fournir ces attributs à votre application. Pour plus d'informations sur les périmètres d'application des fournisseurs sociaux, consultez la documentation de Google, Facebook, Login with Amazon et Login with Apple.

Dans le cas de Sign in with Apple, les scénarios d'utilisation où les périmètres ne peuvent pas être renvoyés sont les suivants :

- Un utilisateur final rencontre des échecs après avoir quitté la page de connexion d'Apple (cela peut être dû à des défaillances internes d'Amazon Cognito ou à un document écrit par le développeur).
 - L'identifiant de service est utilisé dans les groupes d'utilisateurs et/ou dans d'autres services d'authentification.
 - Un développeur ajoute des étendues après la connexion de l'utilisateur. Les utilisateurs ne récupèrent de nouvelles informations que lorsqu'ils s'authentifient et lorsqu'ils actualisent leurs jetons.
 - Un développeur supprime l'utilisateur, puis celui-ci se reconnecte sans supprimer l'application de son profil Apple ID.
8. Mappez les attributs de votre fournisseur d'identité à votre groupe d'utilisateurs. Pour plus d'informations, consultez [Choses à savoir sur les mappages](#).
 9. Sélectionnez Créer.
 10. De l'onglet Intégration du client d'application, choisissez l'un des Clients d'application dans la liste et Modifier les paramètres d'interface utilisateur hébergée. Ajoutez le nouveau fournisseur d'identité sociale au client d' l'application sous Fournisseurs d'identité.
 11. Sélectionnez Enregistrer les modifications.

Test de la configuration de votre fournisseur d'identité social

Vous pouvez créer une URL de connexion en utilisant les éléments des deux sections précédentes. Utilisez-les pour tester votre configuration de fournisseur d'identité social.

```
https://mydomain.us-east-1.amazoncognito.com/login?  
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Votre domaine se trouve sur la page de la console répertoriant le nom de domaine du groupe d'utilisateurs. L'ID client se trouve sur la page Paramètres du client d'application. Utilisez votre URL de rappel pour le paramètre `redirect_uri`. Il s'agit de l'URL de la page vers laquelle l'utilisateur est redirigé après une authentification réussie.

Note

Amazon Cognito annule les demandes d'authentification qui ne se terminent pas dans les 5 minutes et redirige l'utilisateur vers l'interface utilisateur hébergée. La page affiche un message d'erreur `Something went wrong`.

Ajouter la connexion à un groupe d'utilisateurs avec un fournisseur d'identité SAML (Facultatif)

Vous pouvez autoriser la connexion des utilisateurs de l'application via un fournisseur d'identité SAML. Que vos utilisateurs se connectent directement ou via un tiers, ils ont tous un profil dans le groupe d'utilisateurs. Ignorez cette étape si vous ne souhaitez pas ajouter la connexion via un fournisseur d'identité SAML.

Pour plus d'informations, consultez [Utilisation de fournisseurs d'identité SAML avec un groupe d'utilisateurs](#).

Vous devez mettre à jour votre fournisseur d'identité SAML et configurer votre groupe d'utilisateurs. Pour plus d'informations sur la façon d'ajouter votre groupe d'utilisateurs en tant que partie de confiance ou application pour votre fournisseur d'identité SAML 2.0, consultez la documentation de votre fournisseur d'identité SAML.

Vous devez également fournir un point de terminaison ACS (Assertion Consumer Service) à votre fournisseur d'identité SAML. Configurez le point de terminaison suivant dans le domaine de votre groupe d'utilisateurs pour la liaison POST SAML 2.0 dans votre fournisseur d'identité SAML. Pour plus d'informations sur les domaines du groupe d'utilisateurs, consultez [Configuration d'un domaine de groupe d'utilisateurs](#).

```
https://Your user pool domain/saml2/idpresponse
```

With an Amazon Cognito domain:

```
https://<yourDomainPrefix>.auth.<region>.amazoncognito.com/saml2/idpresponse
```

With a custom domain:

```
https://Your custom domain/saml2/idpresponse
```

Vous trouverez le préfixe de votre domaine et la valeur de région pour votre groupe d'utilisateurs dans l'onglet Nom de domaine de la console [Amazon Cognito](#).

Pour certains fournisseurs d'identité SAML, vous devez également fournir le fournisseur de services (SP)urn, également appelé URI d'audience ou ID d'entité SP, au format suivant :

```
urn:amazon:cognito:sp:<yourUserPoolID>
```

L'ID de votre groupe d'utilisateurs se trouve dans l'onglet Paramètres généraux de la [console Amazon Cognito](#).

Vous devez également configurer votre fournisseur d'identité SAML afin qu'il fournisse des valeurs d'attributs pour tous les attributs requis dans votre groupe d'utilisateurs. Généralement, email est un attribut requis pour les groupes d'utilisateurs. Dans ce cas, le fournisseur d'identité SAML doit fournir une valeur email (revendication) dans l'assertion SAML.

Les groupes d'utilisateurs Amazon Cognito prennent en charge la fédération SAML 2.0 avec points de terminaison de liaison postérieure. Votre application n'a donc plus besoin de récupérer ou d'analyser les réponses aux assertions SAML, car le groupe d'utilisateurs reçoit directement la réponse SAML de votre fournisseur d'identité via un agent utilisateur.

Pour configurer un fournisseur d'identité SAML 2.0 dans votre groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet Expérience de connexion. Localisez la Session fédérée et sélectionnez Ajouter un fournisseur d'identité.
5. Choisissez un SAML fournisseur d'identité social.
6. Saisissez Identifiants séparés par des virgules. Un identifiant indique à Amazon Cognito qu'il doit vérifier l'adresse e-mail saisie par un utilisateur lorsqu'il se connecte. Il les dirige ensuite vers le fournisseur correspondant à leur domaine.
7. Choisissez Ajouter un flux de déconnexion si vous souhaitez qu'Amazon Cognito envoie des demandes de déconnexion signées à votre fournisseur lorsqu'un utilisateur se déconnecte. Vous devez configurer votre fournisseur d'identité SAML 2.0 afin qu'il envoie les réponses de déconnexion au point de terminaison `https://<your Amazon Cognito domain>/saml2/logout` créé lorsque vous configurez l'interface utilisateur hébergée. Le `saml2/logout` point de terminaison utilise la liaison POST.

Note

Si cette option est sélectionnée et que votre fournisseur d'identité SAML attend une demande de déconnexion signée, vous devrez également configurer le certificat de signature fourni par Amazon Cognito avec votre IdP SAML.

L'IdP SAML traitera la demande de déconnexion signée et déconnectera votre utilisateur de la session Amazon Cognito.

8. Choisissez une Source du document de métadonnées. Si votre fournisseur d'identité propose des métadonnées SAML à une URL publique, vous pouvez choisir Metadata document URL (URL du document de métadonnées) et saisir cette URL publique. Sinon, choisissez Upload metadata document (Charger un document de métadonnées) et sélectionnez un fichier de métadonnées que vous avez téléchargé depuis votre fournisseur précédemment.

Note

Nous vous recommandons de saisir l'URL d'un document de métadonnées si votre fournisseur dispose d'un point de terminaison public, plutôt que de télécharger un fichier. Cela permet à Amazon Cognito d'actualiser automatiquement les métadonnées. En règle générale, l'actualisation des métadonnées a lieu toutes les 6 heures ou avant l'expiration des métadonnées, selon la première éventualité.

9. Sélectionnez Mappage des attributs entre votre fournisseur SAML et votre application pour mapper les attributs du fournisseur SAML au profil utilisateur de votre groupe d'utilisateurs. Incluez les attributs requis de votre groupe d'utilisateurs dans votre carte attributaire.

Par exemple, lorsque vous choisissez le champ groupe d'utilisateurs email, saisissez le nom de l'attribut SAML tel qu'il apparaît dans l'assertion SAML de votre fournisseur d'identité. Votre fournisseur d'identité peut proposer des exemples d'assertions SAML à titre de référence. Certains fournisseurs d'identité utilisent des noms simples, comme email, tandis que d'autres utilisent des noms d'attributs au format URL, tels que l'exemple suivant :

```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

10. Sélectionnez Create (Créer).

Commencer à utiliser les pools d'identités Amazon Cognito

Les groupes d'identités Amazon Cognito vous permettent de créer des identités uniques et d'attribuer des autorisations aux utilisateurs. Votre groupe d'identités peut inclure :

- Utilisateurs dans un groupes d'utilisateurs Amazon Cognito
- Utilisateurs qui s'authentifient avec des fournisseurs d'identité externes, tels que Facebook, Google, Apple ou un fournisseur d'identité OIDC ou SAML.
- Utilisateurs authentifiés via votre propre processus d'authentification existant.

Avec un pool d'identités, vous pouvez obtenir des AWS informations d'identification temporaires avec des autorisations que vous définissez pour accéder directement à d'autres ressources Services AWS ou pour accéder à des ressources via Amazon API Gateway.

Rubriques

- [Créer un groupe d'identités dans Amazon Cognito](#)
- [Configuration d'un kit SDK](#)
- [Intégrer les fournisseurs d'identité](#)
- [Obtenir des informations d'identification](#)

Créer un groupe d'identités dans Amazon Cognito

Vous pouvez créer un groupe d'identités via la console Amazon Cognito, l' AWS Command Line Interface (CLI) ou des API Amazon Cognito.

Pour créer un groupe d'identités dans la console

1. Connectez-vous à la [console Amazon Cognito](#) et sélectionnez Groupes d'identités.
2. Choisissez Créer un groupe d'identités.
3. Dans Configurer l'approbation du groupe d'identités, choisissez de configurer votre réserve d'identités en sélectionnant Accès authentifié, Accès invité ou les deux.
 - Si vous avez choisi Accès authentifié, sélectionnez un ou plusieurs types d'identité que vous souhaitez définir comme source des identités authentifiées dans votre réserve d'identités. Si

vous configurez un fournisseur du développeur personnalisé, vous ne pouvez ni le modifier ni le supprimer après avoir créé votre réserve d'identités.

4. Dans Configurer les autorisations, choisissez un rôle IAM par défaut pour les utilisateurs authentifiés ou invités dans votre réserve d'identités.
 - a. Choisissez Créer un nouveau rôle IAM si vous souhaitez qu'Amazon Cognito crée un nouveau rôle pour vous avec des autorisations de base et une relation d'approbation avec votre réserve d'identités. Saisissez le nom du rôle IAM pour identifier votre nouveau rôle, par exemple `myidentitypool_authenticatedrole`. Sélectionnez Afficher le document de stratégie pour passer en revue les autorisations qu'Amazon Cognito attribuera à votre nouveau rôle IAM.
 - b. Vous pouvez choisir d'utiliser un rôle IAM existant si vous avez déjà un rôle Compte AWS que vous souhaitez utiliser dans le vôtre. Vous devez configurer votre politique d'approbation de rôle IAM de manière à inclure `cognito-identity.amazonaws.com`. Configurez votre politique d'approbation de rôle pour autoriser Amazon Cognito à endosser le rôle uniquement quand il présente une preuve que la demande provient d'un utilisateur authentifié dans votre réserve d'identités spécifique. Pour plus d'informations, consultez [Autorisations et approbation de rôle](#).
5. Dans Connect identity providers, entrez les détails des fournisseurs d'identité (IdPs) que vous avez choisis dans Configurer la confiance du pool d'identités. Vous pouvez être invité à fournir des informations sur le client d'application OAuth, à choisir un groupe d'utilisateurs Amazon Cognito, à choisir un fournisseur d'identité IAM ou à saisir un identifiant personnalisé pour un fournisseur de développement.
 - a. Choisissez les paramètres de rôle pour chaque fournisseur d'identité. Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles. Avec un fournisseur d'identité de groupe d'utilisateurs Amazon Cognito, vous pouvez également sélectionner Choisir le rôle avec `preferred_role` dans les jetons. Pour plus d'informations sur le champ standard `cognito:preferred_role`, consultez [Affectation de valeurs de priorité à des groupes](#).
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond.

- Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
- ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
- b. Configurez Attributs de contrôle d'accès pour chaque fournisseur d'identité. L'option Attributs de contrôle d'accès mappe les champs standard utilisateur sur les [balises de principal](#) qu'Amazon Cognito applique à la session temporaire. Vous pouvez générer des politiques IAM pour filtrer l'accès des utilisateurs en fonction des balises que vous appliquez à leur session.
- i. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - ii. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - iii. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
6. Dans Configurer les propriétés, saisissez un nom sous Nom du groupe d'identités.
7. Sous Authentification de base (classique), choisissez si vous souhaitez activer le flux de base. Lorsque le flux de base est actif, vous pouvez ignorer les sélections de rôles que vous avez effectuées pour vos IdPs et appeler [AssumeRoleWithWebIdentity](#) directement. Pour plus d'informations, consultez [Flux d'authentification de groupes d'identités \(identités fédérées\)](#).
8. Sous Balises, choisissez Ajouter une balise si vous souhaitez appliquer des [balises](#) à votre réserve d'identités.
9. Dans Vérifier et créer, confirmez les sélections que vous avez effectuées pour votre nouvelle réserve d'identités. Sélectionnez Modifier pour revenir dans l'assistant et modifier des paramètres. Lorsque vous avez terminé, sélectionnez Créer un groupe d'identités.

Configuration d'un kit SDK

Pour utiliser les pools d'identités Amazon Cognito, configurez le AWS Amplify AWS SDK for Java, ou le AWS SDK for .NET Pour plus d'informations, consultez les rubriques suivantes.

- [Configuration du SDK JavaScript dans le](#) guide du AWS SDK for Java développeur

- [Documentation Amplify](#) (langue française non garantie) sur le site Amplify Dev Center
- [Fournisseur d'informations d'identification Amazon Cognito](#) dans le Guide du développeur AWS SDK for .NET

Intégrer les fournisseurs d'identité

Les réserves d'identités Amazon Cognito (identités fédérées) prennent en charge l'authentification utilisateur via les groupes d'utilisateurs Amazon Cognito, les fournisseurs d'identité fédérée (y compris les fournisseurs d'identité Amazon, Facebook, Google, Apple et SAML) et des identités non authentifiées. Cette fonctionnalité prend également en charge [Identités authentifiées par le développeur \(réserves d'identités\)](#), qui vous permet d'enregistrer et d'authentifier les utilisateurs via votre propre processus d'authentification backend.

Pour en savoir plus sur l'utilisation d'un groupe d'utilisateurs Amazon Cognito afin de créer votre propre répertoire d'utilisateurs, consultez [Groupes d'utilisateurs Amazon Cognito](#) et [Accès à Services AWS l'aide d'un pool d'identités après la connexion](#).

Pour en savoir plus sur l'utilisation des fournisseurs d'identité externes, consultez [Fournisseurs d'identité externes aux groupes d'identités](#).

Pour en savoir plus sur l'intégration de votre propre processus d'authentification backend, consultez [Identités authentifiées par le développeur \(réserves d'identités\)](#).

Obtenir des informations d'identification

Les pools d'identités Amazon Cognito fournissent des AWS informations d'identification temporaires aux utilisateurs invités (non authentifiés) et aux utilisateurs qui se sont authentifiés et ont reçu un jeton. Grâce à ces AWS informations d'identification, votre application peut accéder en toute sécurité à un backend AWS interne ou externe AWS via Amazon API Gateway. veuillez consulter [Obtention des informations d'identification](#).

Options de configuration guidée pour Amazon Cognito

Vous souhaitez peut-être évaluer les fonctionnalités d'Amazon Cognito dans le cadre d'une expérience guidée structurée. Voici quelques ressources externes qui proposent des expériences personnalisées avec des groupes d'utilisateurs et des groupes d'identités.

Terminez un atelier

AWS workshop studio [organise un atelier](#) qui vous explique comment configurer la plupart des fonctionnalités d'Amazon Cognito. Ces fonctionnalités incluent l'API des groupes d'utilisateurs, l'interface utilisateur hébergée des groupes d'utilisateurs, les pools d'identités et la configuration de sécurité.

Ajouter du code d'application à partir d'exemples

Le chapitre sur les [exemples de code](#) de ce guide contient du code d'application que vous pouvez utiliser avec les groupes d'utilisateurs et les groupes d'identités. La section sur les groupes d'utilisateurs du chapitre sur les exemples de code contient de courts extraits qui couvrent des opérations individuelles, ainsi que des exemples plus longs, end-to-end par exemple des applications dans divers langages de programmation.

Créez une application Fullstack avec AWS Amplify

[AWS Amplify](#) est Service AWS destiné aux développeurs qui souhaitent développer et héberger une application et une interface utilisateur. Amazon Cognito est le composant d'authentification d'Amplify. Lorsque vous ajoutez l'authentification à votre application, Amplify peut automatiser le déploiement du pool d'utilisateurs et des ressources du pool d'identités Amazon Cognito. Consultez aussi [Intégration de l'authentification et de l'autorisation Amazon Cognito avec des applications Web et mobiles](#).

Plus de ressources sur les applications Amazon Cognito sur GitHub

- [Exemples de flux d'authentification avec .NET pour Amazon Cognito](#)
- [Authentification sans mot de passe Amazon Cognito](#)
- [PetStore exemple avec Amazon Verified Permissions](#)
- [Exemple d'application React utilisant des pools d'identités ABAC + pour accéder aux ressources AWS](#)
- [Autorisation machine à machine basée sur Amazon Cognito et API Gateway à l'aide du CDK AWS](#)

- [Création d'autorisations précises à l'aide d'Amazon Cognito, d'API Gateway et d'IAM](#)
- [CloudFrontauthorization @edge](#)

Plus d'ateliers

- [Mettez en œuvre l'authentification sans mot de passe avec Amazon Cognito et WebAuthn](#)
- [Identité SaaS multi-locataires avec groupes d'utilisateurs Amazon Cognito](#)
- [Présentation approfondie d'Amazon Cognito JWT](#)

Intégration de l'authentification et de l'autorisation Amazon Cognito avec des applications Web et mobiles

Lorsque vous intégrez votre application à un client d'application Amazon Cognito, vous pouvez invoquer des opérations d'API pour l'authentification et l'autorisation de vos utilisateurs. Nous vous recommandons d'[AWS Amplify](#) intégrer Amazon Cognito à vos applications Web et mobiles. AWS Amplify est une solution complète qui permet aux développeurs Web et mobiles frontaux de créer, de connecter et d'héberger facilement des applications Fullstack AWS, avec la flexibilité nécessaire pour tirer parti de l'ampleur de l'évolution de vos Services AWS cas d'utilisation. Amplify Auth utilise principalement Amazon Cognito pour créer des fonctionnalités d'authentification.

Rubriques

- [Authentification avec AWS Amplify](#)
- [Authentification avec les kits AWS SDK](#)
- [Autorisation avec Amazon Verified Permissions](#)

Une implémentation classique d'Amazon Cognito utilise un mélange d'outils visuels et d'API. La console Amazon Cognito est l'interface visuelle permettant de configurer et de gérer vos groupes d'utilisateurs et réserves d'identités Amazon Cognito. L'interface utilisateur hébergée est une application de connexion ready-to-use Web permettant de tester et de déployer rapidement des groupes d'utilisateurs Amazon Cognito. En outre, dans la plupart des déploiements Amazon Cognito, vous devez ajouter du code dans vos applications pour interagir avec vos groupes d'utilisateurs et vos réserves d'identités. Par exemple, votre application peut invoquer l'interface utilisateur hébergée pour la connexion des utilisateurs, puis appeler le point de terminaison du jeton à partir du code de votre application pour échanger le code d'autorisation de votre utilisateur contre des jetons. Votre application doit ensuite interpréter et stocker les jetons de vos utilisateurs, puis les présenter dans le contexte approprié pour l'authentification et l'autorisation. Amplify ajoute des outils d'intégration guidée avec des fonctions intégrées pour ces processus.

Vous pouvez également créer vos ressources Amazon Cognito entièrement en code. Pour commencer à utiliser votre propre code d'application personnalisé, consultez les [exemples de code](#) Amazon Cognito pour les [SDK AWS](#). Pour l'intégration à Amazon Cognito en tant que fournisseur d'identité OpenID Connect, utilisez les [Outils de développement OpenID Connect](#).

Avant d'utiliser l'authentification et l'autorisation Amazon Cognito, choisissez une plateforme d'application et préparez votre code pour l'intégrer au service. Pour connaître les plateformes disponibles, consultez [Authentification avec les kits AWS SDK](#). AWS CLI II s'agit d'un SDK en ligne de commande pour Amazon Cognito et d'autres Services AWS applications. Il constitue un excellent point de départ pour vous familiariser avec l'API Amazon Cognito.

Note

Certains composants d'Amazon Cognito ne peuvent être configurés qu'avec cette API. Par exemple, vous ne pouvez définir un déclencheur Lambda d'[expéditeur de SMS ou d'e-mail personnalisé](#) pour un groupe d'utilisateurs qu'avec une demande qui met à jour la LambdaConfig propriété de la [UserPool](#) classe dans une demande d'[UpdateUserPool](#) API [CreateUserPool](#) ou d'API.

L'API des groupes d'utilisateurs Amazon Cognito partage son espace de nommage avec plusieurs classes d'opérations d'API. Une classe configure les groupes d'utilisateurs et leurs processus, les fournisseurs d'identité et les utilisateurs. Une autre inclut des opérations non authentifiées permettant à vos utilisateurs d'un client public de se connecter, de se déconnecter et de gérer leurs profils. La dernière classe d'opérations d'API effectue des opérations utilisateur que vous autorisez avec vos propres AWS informations d'identification dans un client confidentiel côté serveur. Vous devez connaître l'architecture de votre application prévue avant de commencer à implémenter le code de l'application. Pour plus d'informations, consultez [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#).

Authentification avec AWS Amplify

AWS Amplify est une solution complète pour créer des applications Web et mobiles. Avec Amplify, vous pouvez vous connecter à des ressources existantes à l'aide des bibliothèques Amplify, ou vous pouvez créer et configurer de nouvelles ressources à l'aide de l'interface de ligne de commande (CLI) d'Amplify. Amplify possède également des composants d'interface utilisateur connectés tels que [Authenticator](#) pour configurer et personnaliser l'expérience de connexion et d'inscription dans votre application.

Pour utiliser les fonctionnalités d'authentification Amplify dans votre application front-end, consultez la documentation suivante par plateforme.

- [Amplify l'authentification pour JavaScript](#)

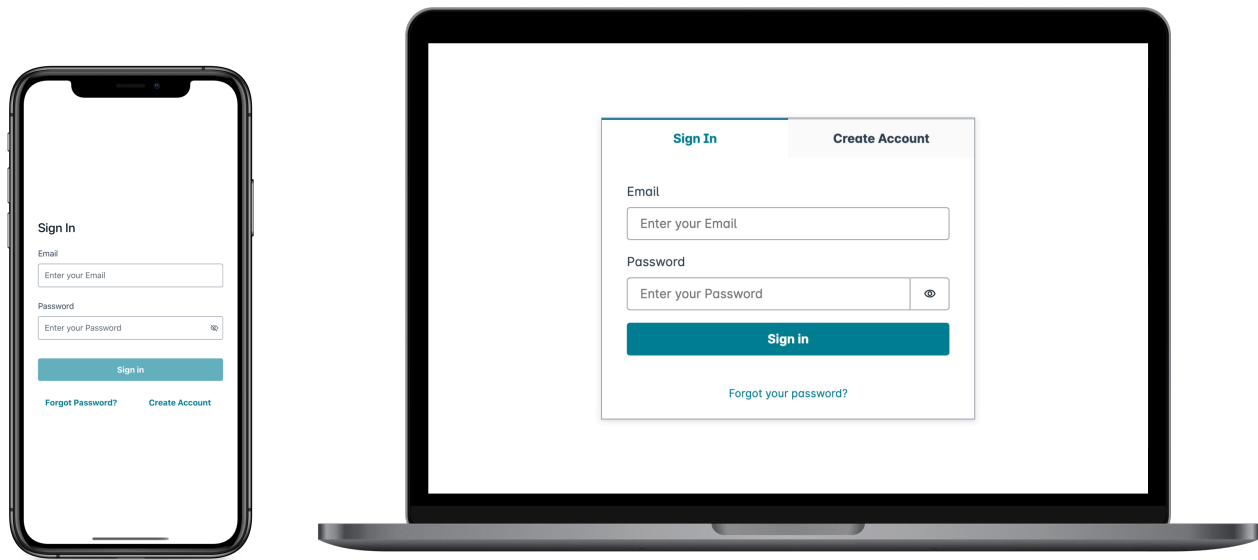
- [Authentification Amplify pour iOS](#)
- [Authentification Amplify pour Android](#)
- [Authentification Amplify pour Flutter](#) (langue française non garantie)

Les bibliothèques Amplify sont open source et sont disponibles sur [GitHub](#). Pour en savoir plus sur la façon dont Amplify Auth implémente l'authentification Amazon Cognito, consultez les bibliothèques suivantes.

- [amplify-js](#)
- [amplify-swift](#)
- [amplify-flutter](#)
- [amplify-android](#)

Créer une interface utilisateur (UI) avec Amplify

L'[interface utilisateur hébergée pour les groupes d'utilisateurs Amazon Cognito](#) peut répondre aux besoins essentiels d'une interface d'authentification front-end pour une application Web ou mobile. Pour personnaliser votre interface utilisateur au-delà des paramètres pris en charge par l'interface utilisateur hébergée, créez une application personnalisée. L'[interface utilisateur Amplify](#) est une collection personnalisable de composants front-end dans différentes langues.



Pour commencer à utiliser votre composant d'authentification personnalisé, consultez la documentation suivante relative au composant Authenticator.

- [Authenticator for Android](#)
- [Authenticator for Angular](#)
- [Authenticator for Flutter](#)
- [Authenticator for React](#)
- [Authenticator for React Native](#)
- [Authenticator for Swift](#)
- [Authenticator for Vue](#)

Authentification avec les kits AWS SDK

Pour utiliser un backend sécurisé afin de créer votre propre microservice d'identité qui interagit avec AWS Amazon Cognito, connectez-vous aux groupes d'utilisateurs Amazon Cognito et à l'API des groupes d'identités Amazon Cognito à l'aide d'un SDK dans la langue de votre choix.

Pour plus d'informations sur chaque opération d'API, consultez les documents [Référence d'API des groupes d'utilisateurs Amazon Cognito](#) et [Référence d'API Amazon Cognito](#). Ces documents

contiennent des sections [Voir aussi](#) qui répertorient les ressources permettant d'utiliser divers kits SDK sur les plateformes prises en charge.

- [Interface de ligne de commande AWS](#)
- [AWS SDK pour .NET](#)
- [AWS SDK pour C++](#)
- [AWS SDK pour Go](#)
- [AWS SDK pour Java V2](#)
- [AWS SDK pour JavaScript](#)
- [AWS SDK pour PHP V3](#)
- [AWS SDK pour Python](#)
- [AWS SDK pour Ruby V3](#)

Autorisation avec Amazon Verified Permissions

[Amazon Verified Permissions](#) est un service d'autorisation pour les applications que vous créez. Lorsque vous ajoutez un groupe d'utilisateurs Amazon Cognito comme source d'identité, votre application peut transmettre des jetons d'accès au groupe d'utilisateurs ou d'identité (ID) à Verified Permissions pour une décision d'autorisation ou de refus. Verified Permissions prend en compte les propriétés de votre utilisateur et le contexte de la demande en fonction des politiques que vous rédigez en [langage de politique Cedar](#). Le contexte de la demande peut inclure un identifiant pour le document, l'image ou toute autre ressource demandée, ainsi que l'action que l'utilisateur souhaite effectuer sur la ressource.

Votre application peut fournir l'identité de votre utilisateur ou des jetons d'accès aux autorisations vérifiées [IsAuthorizedWithToken](#) ou aux demandes [BatchIsAuthorizedWithToken](#) d'API. Ces opérations d'API acceptent vos utilisateurs en tant que tels `Principal` et prennent des décisions d'autorisation pour `Resource` celui auquel ils souhaitent accéder. `Action` Une personnalisation supplémentaire `Context` peut contribuer à une décision d'accès détaillée.

Lorsque votre application présente un jeton dans une demande d'API `IsAuthorizedWithToken`, Verified Permissions effectue les validations suivantes.

1. Votre groupe d'utilisateurs est une [source d'identité](#) Verified Permissions configurée pour le magasin de politiques demandé.

2. Le champ standard `client_id` ou `aud`, dans votre jeton d'accès ou d'identité respectivement, correspond à l'ID client d'application de groupe d'utilisateurs que vous avez fourni à Verified Permissions. Pour vérifier ce champ standard, vous devez [configurer la validation de l'ID client](#) dans votre source d'identité Verified Permissions.
3. Votre jeton n'a pas expiré.
4. La valeur de la `token_use` réclamation contenue dans votre jeton correspond aux paramètres que vous avez transmis `IsAuthorizedWithToken`. La `token_use` réclamation doit être `access` si vous l'avez transmise au `accessToken` paramètre et `id` si vous l'avez transmise au `identityToken` paramètre.
5. La signature figurant dans votre jeton provient des clés Web JSON (JWK) publiées de votre groupe d'utilisateurs. Vous pouvez consulter vos clés JWK à l'adresse `https://cognito-idp.Region.amazonaws.com/your user pool ID/.well-known/jwks.json`.

Jetons révoqués et utilisateurs supprimés

Verified Permissions ne valide que les informations qu'il obtient de votre source d'identité et depuis l'expiration du jeton de votre utilisateur. Verified Permissions ne recherche pas la révocation du jeton ni l'existence de l'utilisateur. Si vous avez révoqué le jeton de votre utilisateur ou supprimé le profil de votre utilisateur de votre groupe d'utilisateurs, Verified Permissions considère toujours le jeton comme valide jusqu'à son expiration.

Évaluation des politiques

Configurez votre groupe d'utilisateurs en tant que [source d'identité](#) pour votre [magasin de politiques](#). Configurez votre application pour envoyer les jetons de vos utilisateurs dans les demandes à Verified Permissions. Pour chaque demande, Verified Permissions compare les champs standard du jeton à une politique. Une politique Verified Permissions est similaire à une politique IAM dans AWS. Elle déclare un principal, une ressource et une action. Verified Permissions répond à votre demande `Allow` si elle correspond à une action autorisée et non à une `Deny` action explicite ; sinon, elle répond par `Deny`. Pour plus d'informations, consultez [Politiques Amazon Verified Permissions](#) dans le Guide de l'utilisateur Amazon Verified Permissions.

Personnalisation des jetons

Pour modifier, ajouter et supprimer les demandes d'utilisateur que vous souhaitez présenter à Verified Permissions, personnalisez le contenu de vos jetons d'accès et d'identité avec un [Déclencheur Lambda avant génération de jeton](#). Un déclencheur avant génération de jeton vous

permet d'ajouter et de modifier des champs standard dans vos jetons. Par exemple, vous pouvez interroger une base de données pour obtenir des attributs utilisateur supplémentaires et les encoder dans votre jeton d'identité.

Note

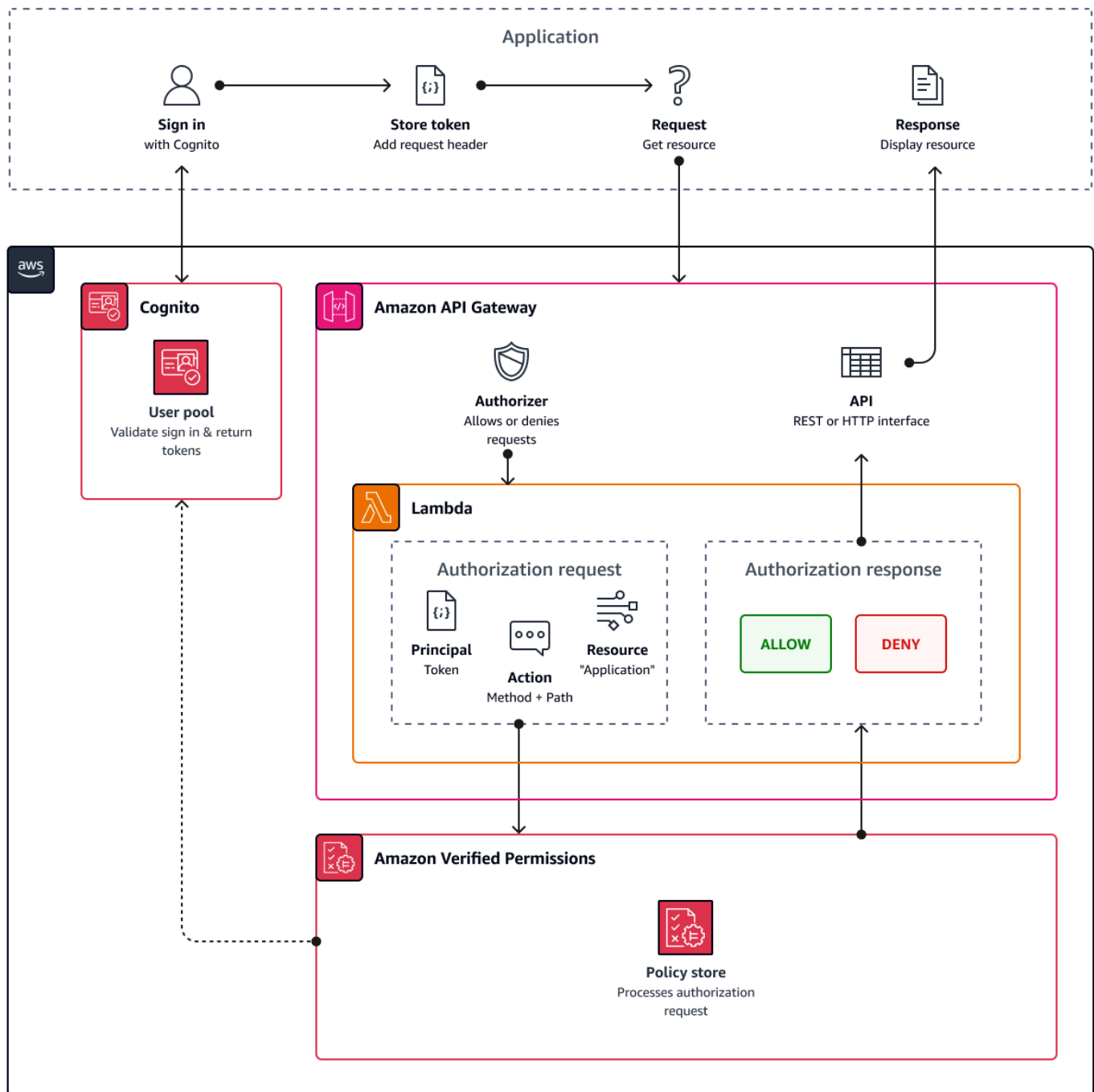
En raison de la façon dont Verified Permissions traite les champs standard, n'ajoutez pas de champs standard nommés `cognito`, `dev` ou `custom` dans votre fonction avant génération de jeton. Lorsque vous présentez ces préfixes de champ standard réservés non pas dans un format délimité par des deux-points, tel que `cognito:username`, mais sous forme de noms de champs standard complets, vos demandes d'autorisation échouent.

Pour plus d'informations sur la façon dont les autorisations vérifiées font correspondre les demandes contenues dans les jetons Amazon Cognito aux politiques d'autorisation, consultez [Associer les jetons Amazon Cognito au schéma des autorisations vérifiées](#).

Autorisation d'API avec autorisations vérifiées

Votre identifiant ou vos jetons d'accès peuvent autoriser les demandes destinées au back-end des API REST Amazon API Gateway avec des autorisations vérifiées. Vous pouvez créer un [magasin de politiques](#) avec des liens immédiats vers votre groupe d'utilisateurs et votre API. Avec l'option de démarrage [Configurer avec Cognito et API Gateway](#), Verified Permissions ajoute une source d'identité de groupe d'utilisateurs au magasin de politiques et un autorisateur Lambda à l'API. Lorsque votre application transmet un jeton porteur d'un pool d'utilisateurs à l'API, l'autorisateur Lambda invoque les autorisations vérifiées. L'autorisateur transmet le jeton en tant que principal et le chemin et la méthode de la demande en tant qu'action.

Le schéma suivant illustre le flux d'autorisation pour une API API Gateway avec des autorisations vérifiées. Pour une analyse détaillée, consultez les [boutiques de politiques liées aux API](#) dans le guide de l'utilisateur Amazon Verified Permissions.



Les autorisations vérifiées structurent l'autorisation de l'API en fonction des [groupes de groupes d'utilisateurs](#). Étant donné que les jetons d'identification et d'accès incluent une `cognito:groups` réclamation, votre magasin de politiques peut gérer le contrôle d'accès basé sur les rôles (RBAC) pour vos API dans divers contextes d'application.

Choix des paramètres du Policy Store

Lorsque vous configurez une source d'identité sur un magasin de politiques, vous devez choisir si vous souhaitez traiter l'accès ou les jetons d'identification. Cette décision est importante pour le fonctionnement de votre moteur de politiques. Les jetons d'identification contiennent des attributs utilisateur. Les jetons d'accès contiennent des informations de contrôle d'accès utilisateur : étendues [OAuth](#). Bien que les deux types de jetons contiennent des informations sur l'appartenance au groupe, nous recommandons généralement le jeton d'accès pour le RBAC avec un magasin de politiques d'autorisations vérifiées. Le jeton d'accès ajoute à l'appartenance au groupe des étendues qui peuvent contribuer à la décision d'autorisation. Les revendications contenues dans un jeton d'accès deviennent [contextuelles](#) dans la demande d'autorisation.

Vous devez également configurer les types d'entités d'utilisateur et de groupe lorsque vous configurez un groupe d'utilisateurs comme source d'identité. Les types d'entités sont des identifiants principaux, d'action et de ressources auxquels vous pouvez faire référence dans les politiques d'autorisations vérifiées. Les entités des magasins de politiques peuvent avoir une relation d'adhésion, dans laquelle une entité peut être membre d'une entité parent. Avec l'adhésion, vous pouvez référencer des groupes principaux, des groupes d'action et des groupes de ressources. Dans le cas de groupes de groupes d'utilisateurs, le type d'entité utilisateur que vous spécifiez doit être membre du type d'entité de groupe. Lorsque vous configurez un [magasin de politiques lié à une API](#) ou que vous suivez la procédure de configuration guidée dans la console des autorisations vérifiées, votre magasin de politiques possède automatiquement cette relation parent-membre.

Le jeton d'identification peut combiner le RBAC avec le contrôle d'accès basé sur les attributs (ABAC). Après avoir créé un [magasin de règles lié à une API](#), vous pouvez améliorer vos politiques avec des [attributs utilisateur](#) et l'appartenance à un groupe. Les revendications d'attributs contenues dans un jeton d'identification deviennent [les attributs principaux](#) de la demande d'autorisation. Vos politiques peuvent prendre des décisions d'autorisation en fonction des principaux attributs.

Vous pouvez également configurer un magasin de politiques pour accepter les jetons avec une `client_id` réclamation aud ou une réclamation correspondant à une liste de clients d'applications acceptables que vous fournissez.

Exemple de politique d'autorisation d'API basée sur les rôles

L'exemple de politique suivant a été créé par la configuration d'un magasin de politiques d'autorisations vérifiées pour un [PetStore](#) exemple d'API REST.

```
permit(
```

```
principal in PetStore::UserGroup::"us-east-1_EXAMPLE|MyGroup",
action in [ PetStore::Action::"get /pets", PetStore::Action::"get /pets/{petId}" ],
resource
);
```

Verified Permissions renvoie une Allow décision à la demande d'autorisation de votre application lorsque :

1. Votre application a transmis un identifiant ou un jeton d'accès dans un Authorization en-tête en tant que jeton porteur.
2. Votre application a transmis un jeton avec une `cognito:groups` réclamation contenant la chaîne `MyGroup`.
3. Votre application a fait une HTTP GET demande à, par exemple, `https://myapi.example.com/pets` ou `https://myapi.example.com/pets/scrappy`.

Exemple de politique pour un utilisateur Amazon Cognito

Votre groupe d'utilisateurs peut également générer des demandes d'autorisation à Verified Permissions dans des conditions autres que les demandes d'API. Vous pouvez soumettre toutes les décisions de contrôle d'accès de votre application à votre magasin de politiques. Par exemple, vous pouvez renforcer la sécurité d'Amazon DynamoDB ou d'Amazon S3 par un contrôle d'accès basé sur les attributs avant que les demandes ne transitent par le réseau, réduisant ainsi l'utilisation des quotas.

L'exemple suivant utilise le [langage de politique Cedar](#) pour permettre aux utilisateurs du service financier qui s'authentifient à l'aide d'un client d'application de groupe d'utilisateurs de lire et d'écrire `example_image.png`. John, un utilisateur de votre application, reçoit un jeton d'identité de la part de votre client d'application et le transmet dans une demande GET à une URL nécessitant une autorisation, `https://example.com/images/example_image.png`. Le jeton d'identité de John a un champ standard `aud` de votre ID client d'application de groupe d'utilisateurs `1234567890example`. Votre fonction Lambda avant génération de jeton a également inséré un nouveau champ standard `costCenter` avec une valeur, pour John, de `Finance1234`.

```
permit (
  principal,
  actions in [ExampleCorp::Action::"readFile", "writeFile"],
  resource == ExampleCorp::Photo::"example_image.png"
)
```



```
when {
  principal.aud == "1234567890example" &&
  principal.custom.costCenter like "Finance*"
};
```

Le corps de demande suivant entraîne la réponse Allow.

```
{
  "accesstoken": "[John's ID token]",
  "action": {
    "actionId": "readFile",
    "actionType": "Action"
  },
  "resource": {
    "entityId": "example_image.png",
    "entityType": "Photo"
  }
}
```

Lorsque vous souhaitez spécifier un principal dans une politique Verified Permissions, utilisez le format suivant :

```
permit (
  principal == [Namespace]::[Entity]::"[user pool ID]"|"[user sub]",
  action,
  resource
);
```

Voici un exemple de principal pour un utilisateur d'un groupe d'utilisateurs dont l'ID us-east-1_Example est accompagné d'un sub, ou d'un ID utilisateur,973db890-092c-49e4-a9d0-912a4c0a20c7.

```
principal == ExampleCorp::User::"us-east-1_Example|973db890-092c-49e4-a9d0-912a4c0a20c7",
```

Lorsque vous souhaitez spécifier un groupe d'utilisateurs dans une politique d'autorisations vérifiées, utilisez le format suivant :

```
permit (
  principal in [Namespace]::[Group Entity]::"[Group name]",
```

```

    action,
    resource
);

```

Ce qui suit est un exemple

Contrôle d'accès basé sur les attributs

L'autorisation avec autorisations vérifiées pour vos applications et les [attributs pour la fonctionnalité de contrôle d'accès](#) des groupes d'identités Amazon Cognito pour les AWS informations d'identification sont deux formes de contrôle d'accès basé sur les attributs (ABAC). Vous trouverez ci-dessous une comparaison des fonctionnalités de Verified Permissions et d'Amazon Cognito ABAC. Dans ABAC, un système examine les attributs d'une entité et prend une décision d'autorisation à partir des conditions que vous définissez.

Service	Processus	Résultat
Amazon Verified Permissions	Renvoie une Deny décision Allow ou issue de l'analyse d'un groupe d'utilisateurs JWT.	L'accès aux ressources des applications réussit ou échoue en fonction de l'évaluation des politiques de Cedar.
Groupes d'identités Amazon Cognito (attributs pour le contrôle d'accès)	Attribue des balises de session à votre utilisateur en fonction de ses attributs. Les conditions de la politique IAM peuvent vérifier les balises Allow ou Deny l'accès des utilisateurs à Services AWS.	Une session balisée avec des AWS informations d'identification temporaires pour un rôle IAM.

Exemples de code pour Amazon Cognito utilisant les kits AWS SDK.

Les exemples de code suivants montrent comment utiliser Amazon Cognito avec un kit de développement logiciel AWS (SDK).

Pour obtenir la liste complète des guides de développement AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- [Exemples de code pour Amazon Cognito Identity à l'aide de kits SDK AWS](#)
 - [Actions pour Amazon Cognito Identity à l'aide des SDK AWS](#)
 - [Utilisation CreateIdentityPool avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteIdentityPool avec un AWS SDK ou une CLI](#)
 - [Utilisation DescribeIdentityPool avec un AWS SDK ou une CLI](#)
 - [Utilisation GetCredentialsForIdentity avec un AWS SDK ou une CLI](#)
 - [Utilisation GetIdentityPoolRoles avec un AWS SDK ou une CLI](#)
 - [Utilisation ListIdentityPools avec un AWS SDK ou une CLI](#)
 - [Utilisation SetIdentityPoolRoles avec un AWS SDK ou une CLI](#)
 - [Utilisation UpdateIdentityPool avec un AWS SDK ou une CLI](#)
 - [Exemples multiservices pour Amazon Cognito Identity à l'aide de kits SDK AWS](#)
 - [Créer une application Amazon Transcribe](#)
 - [Créer une application Amazon Textract Explorer](#)
- [Exemples de code pour le fournisseur d'identité Amazon Cognito utilisant des SDK AWS](#)
 - [Actions pour le fournisseur d'identité Amazon Cognito à l'aide de kits de développement logiciel AWS](#)
 - [Utilisation AdminCreateUser avec un AWS SDK ou une CLI](#)
 - [Utilisation AdminGetUser avec un AWS SDK ou une CLI](#)
 - [Utilisation AdminInitiateAuth avec un AWS SDK ou une CLI](#)
 - [Utilisation AdminRespondToAuthChallenge avec un AWS SDK ou une CLI](#)

- [Utilisation AdminSetUserPassword avec un AWS SDK ou une CLI](#)
- [Utilisation AssociateSoftwareToken avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmDevice avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmForgotPassword avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmSignUp avec un AWS SDK ou une CLI](#)
- [Utilisation CreateUserPool avec un AWS SDK ou une CLI](#)
- [Utilisation CreateUserPoolClient avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteUser avec un AWS SDK ou une CLI](#)
- [Utilisation ForgotPassword avec un AWS SDK ou une CLI](#)
- [Utilisation InitiateAuth avec un AWS SDK ou une CLI](#)
- [Utilisation ListUserPools avec un AWS SDK ou une CLI](#)
- [Utilisation ListUsers avec un AWS SDK ou une CLI](#)
- [Utilisation ResendConfirmationCode avec un AWS SDK ou une CLI](#)
- [Utilisation RespondToAuthChallenge avec un AWS SDK ou une CLI](#)
- [Utilisation SignUp avec un AWS SDK ou une CLI](#)
- [Utilisation UpdateUserPool avec un AWS SDK ou une CLI](#)
- [Utilisation VerifySoftwareToken avec un AWS SDK ou une CLI](#)
- [Scénarios pour le fournisseur d'identité Amazon Cognito utilisant des kits de développement logiciel AWS](#)
 - [Confirmez automatiquement les utilisateurs Amazon Cognito connus à l'aide d'une fonction Lambda à l'aide d'un SDK AWS](#)
 - [Migrez automatiquement les utilisateurs connus d'Amazon Cognito à l'aide d'une fonction Lambda à l'aide d'un SDK AWS](#)
 - [Inscrire un utilisateur auprès d'un groupe d'utilisateurs Amazon Cognito qui nécessite l'authentification multifacteur à l'aide d'un SDK AWS](#)
 - [Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito à l'aide d'un SDK AWS](#)
- [Exemples de code pour Amazon Cognito Sync à l'aide de kits SDK AWS](#)
 - [Actions pour Amazon Cognito Sync à l'aide des SDK AWS](#)
 - [Utilisation ListIdentityPoolUsage avec un AWS SDK ou une CLI](#)

Exemples de code pour Amazon Cognito Identity à l'aide de kits SDK AWS

Les exemples de code suivants montrent comment utiliser Amazon Cognito Identity avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés et dans des exemples interservices.

Les Exemples de services croisés sont des exemples d'applications fonctionnant sur plusieurs Services AWS.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- [Actions pour Amazon Cognito Identity à l'aide des SDK AWS](#)
 - [Utilisation CreateIdentityPool avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteIdentityPool avec un AWS SDK ou une CLI](#)
 - [Utilisation DescribeIdentityPool avec un AWS SDK ou une CLI](#)
 - [Utilisation GetCredentialsForIdentity avec un AWS SDK ou une CLI](#)
 - [Utilisation GetIdentityPoolRoles avec un AWS SDK ou une CLI](#)
 - [Utilisation ListIdentityPools avec un AWS SDK ou une CLI](#)
 - [Utilisation SetIdentityPoolRoles avec un AWS SDK ou une CLI](#)
 - [Utilisation UpdateIdentityPool avec un AWS SDK ou une CLI](#)
- [Exemples multiservices pour Amazon Cognito Identity à l'aide de kits SDK AWS](#)
 - [Créer une application Amazon Transcribe](#)
 - [Créer une application Amazon Textract Explorer](#)

Actions pour Amazon Cognito Identity à l'aide des SDK AWS

Les exemples de code suivants montrent comment effectuer des actions Amazon Cognito Identity individuelles à l'aide AWS des SDK. Ces extraits appellent l'API Identité Amazon Cognito et sont des extraits de code de programmes de plus grande envergure qui doivent être exécutés en contexte. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez [Amazon Cognito Identity API Reference](#) (Référence de l'API Identité Amazon Cognito).

Exemples

- [Utilisation CreateIdentityPool avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteIdentityPool avec un AWS SDK ou une CLI](#)
- [Utilisation DescribeIdentityPool avec un AWS SDK ou une CLI](#)
- [Utilisation GetCredentialsForIdentity avec un AWS SDK ou une CLI](#)
- [Utilisation GetIdentityPoolRoles avec un AWS SDK ou une CLI](#)
- [Utilisation ListIdentityPools avec un AWS SDK ou une CLI](#)
- [Utilisation SetIdentityPoolRoles avec un AWS SDK ou une CLI](#)
- [Utilisation UpdateIdentityPool avec un AWS SDK ou une CLI](#)

Utilisation **CreateIdentityPool** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `CreateIdentityPool`.

CLI

AWS CLI

Pour créer une réserve d'identités avec le fournisseur de réserves d'identités Cognito

Cet exemple crée un pool d'identités nommé `MyIdentityPool`. Il dispose d'un fournisseur de réserve d'identités Cognito. Les identités non authentifiées ne sont pas autorisées.

Commande :

```
aws cognito-identity create-identity-pool --identity-pool-name
MyIdentityPool --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_aaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Sortie :

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
    {
      "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-
west-2_11111111",
      "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
      "ServerSideTokenCheck": false
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous [CreateIdentityPool](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet [GitHub](#). Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
import
software.amazon.awssdk.services.cognitoidentity.model.CreateIdentityPoolRequest;
import
software.amazon.awssdk.services.cognitoidentity.model.CreateIdentityPoolResponse;
```

```
import
software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExco

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateIdentityPool {
    public static void main(String[] args) {
        final String usage = ""
            Usage:
                <identityPoolName>\s

            Where:
                identityPoolName - The name to give your identity pool.
            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String identityPoolName = args[0];
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
            .region(Region.US_EAST_1)
            .build();

        String identityPoolId = createIdPool(cognitoClient, identityPoolName);
        System.out.println("Unity pool ID " + identityPoolId);
        cognitoClient.close();
    }

    public static String createIdPool(CognitoIdentityClient cognitoClient, String
identityPoolName) {
        try {
            CreateIdentityPoolRequest poolRequest =
CreateIdentityPoolRequest.builder()
                .allowUnauthenticatedIdentities(false)
                .identityPoolName(identityPoolName)

```



```
        .build();

        CreateIdentityPoolResponse response =
cognitoClient.createIdentityPool(poolRequest);
        return response.identityPoolId();

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
}
```

- Pour plus de détails sur l'API, reportez-vous [CreateIdentityPool](#) à la section Référence des AWS SDK for Java 2.x API.

PowerShell

Outils pour PowerShell

Exemple 1 : crée un nouveau pool d'identités qui autorise les identités non authentifiées.

```
New-CGIIIdentityPool -AllowUnauthenticatedIdentities $true -IdentityPoolName
CommonTests13
```

Sortie :

```
LoggedAt                : 8/12/2015 4:56:07 PM
AllowUnauthenticatedIdentities : True
DeveloperProviderName   :
IdentityPoolId         : us-east-1:15d49393-ab16-431a-b26e-EXAMPLEGUID3
IdentityPoolName       : CommonTests13
OpenIdConnectProviderARNs : {}
SupportedLoginProviders : {}
ResponseMetadata       : Amazon.Runtime.ResponseMetadata
ContentLength          : 136
HttpStatusCode         : OK
```

- Pour plus de détails sur l'API, reportez-vous [CreateIdentityPool](#) à la section Référence des AWS Tools for PowerShell applets de commande.

Swift

Kit SDK pour Swift

Note

Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Créez un nouveau groupe d'identités.

```
/// Create a new identity pool and return its ID.
///
/// - Parameters:
///   - name: The name to give the new identity pool.
///
/// - Returns: A string containing the newly created pool's ID, or `nil`
///   if an error occurred.
///
func createIdentityPool(name: String) async throws -> String? {
    let cognitoInputCall = CreateIdentityPoolInput(developerProviderName:
"com.exampleco.CognitoIdentityDemo",
                                                    identityPoolName: name)

    let result = try await cognitoIdentityClient.createIdentityPool(input:
cognitoInputCall)
    guard let poolId = result.identityPoolId else {
        return nil
    }

    return poolId
}
```

- Pour plus d'informations, consultez [Guide du développeur du kit AWS SDK pour Swift](#).
- Pour plus de détails sur l'API, reportez-vous [CreateIdentityPool](#) à la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeleteIdentityPool** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `DeleteIdentityPool`.

CLI

AWS CLI

Pour supprimer une réserve d'identités

L'exemple `delete-identity-pool` suivant supprime la réserve d'identités spécifiée.

Commande :

```
aws cognito-identity delete-identity-pool \  
  --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous [DeleteIdentityPool](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.awscore.exception.AwsServiceException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
import
  software.amazon.awssdk.services.cognitoidentity.model.DeleteIdentityPoolRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteIdentityPool {

    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <identityPoolId>\s

            Where:
                identityPoolId - The Id value of your identity pool.
            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String identityPoolId = args[0];
        CognitoIdentityClient cognitoIdClient = CognitoIdentityClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

        deleteIdPool(cognitoIdClient, identityPoolId);
        cognitoIdClient.close();
    }
}
```

```
public static void deleteIdPool(CognitoIdentityClient cognitoIdClient, String
identityPoolId) {
    try {

        DeleteIdentityPoolRequest identityPoolRequest =
DeleteIdentityPoolRequest.builder()
        .identityPoolId(identityPoolId)
        .build();

        cognitoIdClient.deleteIdentityPool(identityPoolRequest);
        System.out.println("Done");

    } catch (AwsServiceException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Pour plus de détails sur l'API, reportez-vous [DeleteIdentityPool](#) à la section Référence des AWS SDK for Java 2.x API.

PowerShell

Outils pour PowerShell

Exemple 1 : Supprime un pool d'identités spécifique.

```
Remove-CGIIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

- Pour plus de détails sur l'API, reportez-vous [DeleteIdentityPool](#) à la section Référence des AWS Tools for PowerShell applets de commande.

Swift

Kit SDK pour Swift

Note

Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Supprimez le groupe d'identités spécifié.

```
/// Delete the specified identity pool.
///
/// - Parameters:
///   - id: The ID of the identity pool to delete.
///
func deleteIdentityPool(id: String) async throws {
    let input = DeleteIdentityPoolInput(
        identityPoolId: id
    )

    _ = try await cognitoIdentityClient.deleteIdentityPool(input: input)
}
```

- Pour plus d'informations, consultez [Guide du développeur du kit AWS SDK pour Swift](#).
- Pour plus de détails sur l'API, reportez-vous [DeleteIdentityPool](#) à la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DescribeIdentityPool** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `DescribeIdentityPool`.

CLI

AWS CLI

Pour décrire un pool d'identités

Cet exemple décrit un pool d'identités.

Commande :

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Sortie :

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
    {
      "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_1111111111",
      "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
      "ServerSideTokenCheck": false
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous [DescribeIdentityPool](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : récupère les informations relatives à un pool d'identités spécifique par son identifiant.

```
Get-CGIIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

Sortie :

```
LoggedAt                : 8/12/2015 4:29:40 PM
AllowUnauthenticatedIdentities : True
DeveloperProviderName   :
IdentityPoolId          : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
IdentityPoolName        : CommonTests1
OpenIdConnectProviderARNs : {}
SupportedLoginProviders  : {}
ResponseMetadata        : Amazon.Runtime.ResponseMetadata
ContentLength            : 142
HttpStatusCode           : OK
```

- Pour plus de détails sur l'API, reportez-vous [DescribeIdentityPool](#) à la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetCredentialsForIdentity** avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser `GetCredentialsForIdentity`.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
```



```
import
    software.amazon.awssdk.services.cognitoidentity.model.GetCredentialsForIdentityRequest;
import
    software.amazon.awssdk.services.cognitoidentity.model.GetCredentialsForIdentityResponse;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class GetIdentityCredentials {
    public static void main(String[] args) {

        final String usage = ""

            Usage:
                <identityId>\s

            Where:
                identityId - The Id of an existing identity in the format
                REGION:GUID.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String identityId = args[0];
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
            .region(Region.US_EAST_1)
            .build();

        getCredsForIdentity(cognitoClient, identityId);
        cognitoClient.close();
    }
}
```

```
public static void getCredsForIdentity(CognitoIdentityClient cognitoClient,
String identityId) {
    try {
        GetCredentialsForIdentityRequest getCredentialsForIdentityRequest =
GetCredentialsForIdentityRequest
            .builder()
            .identityId(identityId)
            .build();

        GetCredentialsForIdentityResponse response = cognitoClient
            .getCredentialsForIdentity(getCredentialsForIdentityRequest);
        System.out.println(
            "Identity ID " + response.identityId() + ", Access key ID " +
response.credentials().accessKeyId());

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Pour plus de détails sur l'API, reportez-vous [GetCredentialsForIdentity](#) à la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetIdentityPoolRoles** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `GetIdentityPoolRoles`.

CLI

AWS CLI

Pour obtenir des rôles dans le pool d'identités

Cet exemple permet d'obtenir les rôles du pool d'identités.

Commande :

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Sortie :

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "Roles": {
    "authenticated": "arn:aws:iam::111111111111:role/Cognito_MyIdentityPoolAuth_Role",
    "unauthenticated": "arn:aws:iam::111111111111:role/Cognito_MyIdentityPoolUnauth_Role"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous [GetIdentityPoolRoles](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : obtient les informations sur les rôles pour un pool d'identités spécifique.

```
Get-CGIIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
```

Sortie :

```
LoggedAt      : 8/12/2015 4:33:51 PM
IdentityPoolId : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
Roles         : {[unauthenticated, arn:aws:iam::123456789012:role/CommonTests1Role]}
ResponseMetadata : Amazon.Runtime.ResponseMetadata
ContentLength  : 165
HttpStatusCode : OK
```

- Pour plus de détails sur l'API, reportez-vous [GetIdentityPoolRoles](#) à la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ListIdentityPools** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `ListIdentityPools`.

CLI

AWS CLI

Pour afficher les réserves d'identités

Cet exemple répertorie les réserves d'identités. Un maximum de 20 identités sont répertoriées.

Commande :

```
aws cognito-identity list-identity-pools --max-results 20
```

Sortie :

```
{
  "IdentityPools": [
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "MyIdentityPool"
    },
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "AnotherIdentityPool"
    },
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "IdentityPoolRegionA"
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous [ListIdentityPools](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
import
    software.amazon.awssdk.services.cognitoidentity.model.ListIdentityPoolsRequest;
import
    software.amazon.awssdk.services.cognitoidentity.model.ListIdentityPoolsResponse;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ListIdentityPools {
    public static void main(String[] args) {
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
            .region(Region.US_EAST_1)
            .build();

        listIdPools(cognitoClient);
        cognitoClient.close();
    }

    public static void listIdPools(CognitoIdentityClient cognitoClient) {
        try {
            ListIdentityPoolsRequest poolsRequest =
                ListIdentityPoolsRequest.builder()

```

```

        .maxResults(15)
        .build();

        ListIdentityPoolsResponse response =
cognitoClient.listIdentityPools(poolsRequest);
        response.identityPools().forEach(pool -> {
            System.out.println("Pool ID: " + pool.identityPoolId());
            System.out.println("Pool name: " + pool.identityPoolName());
        });

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
}

```

- Pour plus de détails sur l'API, reportez-vous [ListIdentityPools](#) à la section Référence des AWS SDK for Java 2.x API.

PowerShell

Outils pour PowerShell

Exemple 1 : récupère une liste de pools d'identités existants.

```
Get-CGIIIdentityPoolList
```

Sortie :

```

IdentityPoolId
IdentityPoolName
-----
-----
us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1      CommonTests1
us-east-1:118d242d-204e-4b88-b803-EXAMPLEGUID2      Tests2
us-east-1:15d49393-ab16-431a-b26e-EXAMPLEGUID3      CommonTests13

```

- Pour plus de détails sur l'API, reportez-vous [ListIdentityPools](#) à la section Référence des AWS Tools for PowerShell applets de commande.

Swift

Kit SDK pour Swift

Note

Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Recherchez l'ID d'un groupe d'identités en fonction de son nom.

```
/// Return the ID of the identity pool with the specified name.
///
/// - Parameters:
///   - name: The name of the identity pool whose ID should be returned.
///
/// - Returns: A string containing the ID of the specified identity pool
///   or `nil` on error or if not found.
///
func getIdentityPoolID(name: String) async throws -> String? {
    var token: String? = nil

    // Iterate over the identity pools until a match is found.

    repeat {
        /// `token` is a value returned by `ListIdentityPools()` if the
        /// returned list of identity pools is only a partial list. You
        /// use the `token` to tell Amazon Cognito that you want to
        /// continue where you left off previously. If you specify `nil`
        /// or you don't provide the token, Amazon Cognito will start at
        /// the beginning.

        let listPoolsInput = ListIdentityPoolsInput(maxResults: 25,
nextToken: token)
```

```

    /// Read pages of identity pools from Cognito until one is found
    /// whose name matches the one specified in the `name` parameter.
    /// Return the matching pool's ID. Each time we ask for the next
    /// page of identity pools, we pass in the token given by the
    /// previous page.

    let output = try await cognitoIdentityClient.listIdentityPools(input:
listPoolsInput)

    if let identityPools = output.identityPools {
        for pool in identityPools {
            if pool.identityPoolName == name {
                return pool.identityPoolId!
            }
        }
    }

    token = output.nextToken
} while token != nil

return nil
}

```

Obtenez l'ID d'un groupe d'identités existant ou créez-le s'il n'existe pas encore.

```

/// Return the ID of the identity pool with the specified name.
///
/// - Parameters:
///   - name: The name of the identity pool whose ID should be returned
///
/// - Returns: A string containing the ID of the specified identity pool.
///   Returns `nil` if there's an error or if the pool isn't found.
///
public func getOrCreateIdentityPoolID(name: String) async throws -> String? {
    // See if the pool already exists. If it doesn't, create it.

    guard let poolId = try await self.getIdentityPoolID(name: name) else {
        return try await self.createIdentityPool(name: name)
    }

    return poolId
}

```


- Pour plus d'informations, consultez [Guide du développeur du kit AWS SDK pour Swift](#).
- Pour plus de détails sur l'API, reportez-vous [ListIdentityPools](#) à la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **SetIdentityPoolRoles** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `SetIdentityPoolRoles`.

CLI

AWS CLI

Pour définir les rôles du pool d'identités

L'`set-identity-pool-role` exemple suivant définit un rôle de pool d'identités.

```
aws cognito-identity set-identity-pool-roles \  
  --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" \  
  --roles authenticated="arn:aws:iam::111111111111:role/  
Cognito_MyIdentityPoolAuth_Role"
```

- Pour plus de détails sur l'API, reportez-vous [SetIdentityPoolRoles](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : configure le pool d'identités spécifique pour qu'il ait un rôle IAM non authentifié.

```
Set-CGIIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-  
EXAMLEGUID1 -Role @{ "unauthenticated" = "arn:aws:iam::123456789012:role/  
CommonTests1Role" }
```

- Pour plus de détails sur l'API, reportez-vous [SetIdentityPoolRoles](#) à la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **UpdateIdentityPool** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `UpdateIdentityPool`.

CLI

AWS CLI

Pour mettre à jour un pool d'identités

Cet exemple met à jour un pool d'identités. Il définit le nom sur `MyIdentityPool`. Il ajoute Cognito en tant que fournisseur d'identité. Il interdit les identités non authentifiées.

Commande :

```
aws cognito-identity update-identity-pool --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" --identity-pool-name "MyIdentityPool" --no-allow-unauthenticated-identities --cognito-identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-west-2_1111111111",ClientId="3n4b5urk1ft4f13mg5e62d9ado",ServerSideTokenCheck=false
```

Sortie :

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
    {
      "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_1111111111",
      "ClientId": "3n4b5urk1ft4f13mg5e62d9ado",
      "ServerSideTokenCheck": false
    }
  ]
}
```

```
]
}
```

- Pour plus de détails sur l'API, reportez-vous [UpdateIdentityPool](#) à la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : met à jour certaines propriétés du pool d'identités, en l'occurrence le nom du pool d'identités.

```
Update-CGIIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1 -IdentityPoolName NewPoolName
```

Sortie :

```
LoggedAt                : 8/12/2015 4:53:33 PM
AllowUnauthenticatedIdentities : False
DeveloperProviderName   :
IdentityPoolId          : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
IdentityPoolName        : NewPoolName
OpenIdConnectProviderARNs : {}
SupportedLoginProviders  : {}
ResponseMetadata        : Amazon.Runtime.ResponseMetadata
ContentLength            : 135
HttpStatusCode           : OK
```

- Pour plus de détails sur l'API, reportez-vous [UpdateIdentityPool](#) à la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples multiservices pour Amazon Cognito Identity à l'aide de kits SDK AWS

Les exemples d'applications suivants utilisent des AWS SDK pour associer Amazon Cognito Identity à d'autres applications. Services AWS Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions sur la façon de configurer et d'exécuter l'application.

Exemples

- [Créer une application Amazon Transcribe](#)
- [Créer une application Amazon Textract Explorer](#)

Créer une application Amazon Transcribe

L'exemple de code suivant montre comment utiliser Amazon Transcribe pour transcrire et afficher des enregistrements vocaux dans le navigateur.

JavaScript

SDK pour JavaScript (v3)

Créez une application qui utilise Amazon Transcribe pour transcrire et afficher des enregistrements vocaux dans le navigateur. L'application utilise deux compartiments Amazon Simple Storage Service (Amazon S3), l'un pour héberger le code de l'application, l'autre pour stocker les transcriptions. L'application utilise un groupe d'utilisateurs Amazon Cognito pour authentifier vos utilisateurs. Les utilisateurs authentifiés disposent des autorisations AWS Identity and Access Management (IAM) nécessaires pour accéder aux services requis. AWS

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur [GitHub](#).

Cet exemple est également disponible dans le [AWS SDK for JavaScript guide du développeur v3](#).

Les services utilisés dans cet exemple

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Créer une application Amazon Textract Explorer

Les exemples de code suivants expliquent comment explorer la sortie Amazon Textract via une application interactive.

JavaScript

SDK pour JavaScript (v3)

Montre comment utiliser le AWS SDK for JavaScript pour créer une application React qui utilise Amazon Textract pour extraire des données d'une image de document et les afficher sur une page Web interactive. Cet exemple s'exécute dans un navigateur Web et nécessite une identité Amazon Cognito authentifiée pour les informations d'identification. Il utilise Amazon Simple Storage Service (Amazon S3) pour le stockage et, pour les notifications, il interroge une file d'attente Amazon Simple Queue Service (Amazon SQS) abonnée à une rubrique Amazon Simple Notification Service (Amazon SNS).

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur [GitHub](#).

Les services utilisés dans cet exemple

- Amazon Cognito Identity
- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples de code pour le fournisseur d'identité Amazon Cognito utilisant des SDK AWS

Les exemples de code suivants montrent comment utiliser le fournisseur d'identité Amazon Cognito avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés et dans des exemples interservices.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir une tâche spécifique en appelant plusieurs fonctions au sein d'un même service.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Mise en route

Bonjour Amazon Cognito

Les exemples de code suivants montrent comment bien démarrer avec Amazon Cognito.

C++

SDK pour C++

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Code pour le MakeLists fichier CMake C.txt.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
```

```
set(SERVICE_COMPONENTS cognito-idp)

# Set this project's name.
project("hello_cognito")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
  libraries for the AWS SDK.
  string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
    "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
  list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
  # Copy relevant AWS SDK for C++ libraries into the current binary directory
  for running and debugging.

  # set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
  may need to uncomment this
  # and set the proper subdirectory to the
  executables' location.

  AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
    ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
  hello_cognito.cpp)

target_link_libraries(${PROJECT_NAME}
  ${AWSSDK_LINK_LIBRARIES})
```

Code pour le fichier source hello_cognito.cpp.

```
#include <aws/core/Aws.h>
#include <aws/cognito-idp/CognitoIdentityProviderClient.h>
#include <aws/cognito-idp/model/ListUserPoolsRequest.h>
#include <iostream>

/*
 * A "Hello Cognito" starter application which initializes an Amazon Cognito
 client and lists the Amazon Cognito
 * user pools.
 *
 * main function
 *
 * Usage: 'hello_cognito'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
    // options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
cognitoClient(clientConfig);

        Aws::String nextToken; // Used for pagination.
        std::vector<Aws::String> userPools;

        do {
            Aws::CognitoIdentityProvider::Model::ListUserPoolsRequest
listUserPoolsRequest;
            if (!nextToken.empty()) {
                listUserPoolsRequest.SetNextToken(nextToken);
            }

            Aws::CognitoIdentityProvider::Model::ListUserPoolsOutcome
listUserPoolsOutcome =
                cognitoClient.ListUserPools(listUserPoolsRequest);
```



```
        if (listUserPoolsOutcome.IsSuccess()) {
            for (auto &userPool:
listUserPoolsOutcome.GetResult().GetUserPools()) {

                userPools.push_back(userPool.GetName());
            }

            nextToken = listUserPoolsOutcome.GetResult().GetNextToken();
        } else {
            std::cerr << "ListUserPools error: " <<
listUserPoolsOutcome.GetError().GetMessage() << std::endl;
            result = 1;
            break;
        }


    } while (!nextToken.empty());
    std::cout << userPools.size() << " user pools found." << std::endl;
    for (auto &userPool: userPools) {
        std::cout << "    user pool: " << userPool << std::endl;
    }
}

Aws::ShutdownAPI(options); // Should only be called once.
return result;
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUserPools](#) à la section Référence des AWS SDK for C++ API.

Go

Kit SDK for Go V2

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
package main

import (
    "context"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
    "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Notification
// Service
// (Amazon SNS) client and list the topics in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        fmt.Println("Couldn't load default configuration. Have you set up your AWS
account?")
        fmt.Println(err)
        return
    }
    cognitoClient := cognitoidentityprovider.NewFromConfig(sdkConfig)
    fmt.Println("Let's list the user pools for your account.")
    var pools []types.UserPoolDescriptionType
    paginator := cognitoidentityprovider.NewListUserPoolsPaginator(
        cognitoClient, &cognitoidentityprovider.ListUserPoolsInput{MaxResults:
aws.Int32(10)})
    for paginator.HasMorePages() {
        output, err := paginator.NextPage(context.TODO())
        if err != nil {
            log.Printf("Couldn't get user pools. Here's why: %v\n", err)
        } else {
            pools = append(pools, output.UserPools...)
        }
    }
    if len(pools) == 0 {
        fmt.Println("You don't have any user pools!")
    }
}
```

```
} else {
  for _, pool := range pools {
    fmt.Printf("\t\t%v: %v\n", *pool.Name, *pool.Id)
  }
}
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUserPools](#) à la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import
  software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
  software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderException;
import
  software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsResponse;
import
  software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
```

```
public class ListUserPools {
    public static void main(String[] args) {
        CognitoIdentityProviderClient cognitoClient =
        CognitoIdentityProviderClient.builder()
            .region(Region.US_EAST_1)
            .build();

        listAllUserPools(cognitoClient);
        cognitoClient.close();
    }

    public static void listAllUserPools(CognitoIdentityProviderClient
cognitoClient) {
        try {
            ListUserPoolsRequest request = ListUserPoolsRequest.builder()
                .maxResults(10)
                .build();

            ListUserPoolsResponse response =
cognitoClient.listUserPools(request);
            response.userPools().forEach(userpool -> {
                System.out.println("User pool " + userpool.name() + ", User ID "
+ userpool.id());
            });

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUserPools](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
  paginateListUserPools,
  CognitoIdentityProviderClient,
} from "@aws-sdk/client-cognito-identity-provider";

const client = new CognitoIdentityProviderClient({});

export const helloCognito = async () => {
  const paginator = paginateListUserPools({ client }, {});

  const userPoolNames = [];

  for await (const page of paginator) {
    const names = page.UserPools.map((pool) => pool.Name);
    userPoolNames.push(...names);
  }

  console.log("User pool names: ");
  console.log(userPoolNames.join("\n"));
  return userPoolNames;
};
```

- Pour plus de détails sur l'API, reportez-vous [ListUserPools](#) à la section Référence des AWS SDK for JavaScript API.

Exemples de code

- [Actions pour le fournisseur d'identité Amazon Cognito à l'aide de kits de développement logiciel AWS](#)
- [Utilisation AdminCreateUser avec un AWS SDK ou une CLI](#)

- [Utilisation AdminGetUser avec un AWS SDK ou une CLI](#)
- [Utilisation AdminInitiateAuth avec un AWS SDK ou une CLI](#)
- [Utilisation AdminRespondToAuthChallenge avec un AWS SDK ou une CLI](#)
- [Utilisation AdminSetUserPassword avec un AWS SDK ou une CLI](#)
- [Utilisation AssociateSoftwareToken avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmDevice avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmForgotPassword avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmSignUp avec un AWS SDK ou une CLI](#)
- [Utilisation CreateUserPool avec un AWS SDK ou une CLI](#)
- [Utilisation CreateUserPoolClient avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteUser avec un AWS SDK ou une CLI](#)
- [Utilisation ForgotPassword avec un AWS SDK ou une CLI](#)
- [Utilisation InitiateAuth avec un AWS SDK ou une CLI](#)
- [Utilisation ListUserPools avec un AWS SDK ou une CLI](#)
- [Utilisation ListUsers avec un AWS SDK ou une CLI](#)
- [Utilisation ResendConfirmationCode avec un AWS SDK ou une CLI](#)
- [Utilisation RespondToAuthChallenge avec un AWS SDK ou une CLI](#)
- [Utilisation SignUp avec un AWS SDK ou une CLI](#)
- [Utilisation UpdateUserPool avec un AWS SDK ou une CLI](#)
- [Utilisation VerifySoftwareToken avec un AWS SDK ou une CLI](#)
- [Scénarios pour le fournisseur d'identité Amazon Cognito utilisant des kits de développement logiciel AWS](#)
 - [Confirmez automatiquement les utilisateurs Amazon Cognito connus à l'aide d'une fonction Lambda à l'aide d'un SDK AWS](#)
 - [Migrez automatiquement les utilisateurs connus d'Amazon Cognito à l'aide d'une fonction Lambda à l'aide d'un SDK AWS](#)
 - [Inscrire un utilisateur auprès d'un groupe d'utilisateurs Amazon Cognito qui nécessite l'authentification multifacteur à l'aide d'un SDK AWS](#)
 - [Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito à l'aide d'un SDK AWS](#)

Actions pour le fournisseur d'identité Amazon Cognito à l'aide de kits de développement logiciel AWS

Les exemples de code suivants montrent comment effectuer des actions individuelles du fournisseur d'identité Amazon Cognito à l'aide AWS des SDK. Ces extraits appellent l'API Fournisseur d'identité Amazon Cognito et sont des extraits de code de programmes de plus grande envergure qui doivent être exécutés en contexte. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez [Amazon Cognito Identity Provider API Reference](#) (Référence de l'API Fournisseur d'identité Amazon Cognito).

Exemples

- [Utilisation AdminCreateUser avec un AWS SDK ou une CLI](#)
- [Utilisation AdminGetUser avec un AWS SDK ou une CLI](#)
- [Utilisation AdminInitiateAuth avec un AWS SDK ou une CLI](#)
- [Utilisation AdminRespondToAuthChallenge avec un AWS SDK ou une CLI](#)
- [Utilisation AdminSetUserPassword avec un AWS SDK ou une CLI](#)
- [Utilisation AssociateSoftwareToken avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmDevice avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmForgotPassword avec un AWS SDK ou une CLI](#)
- [Utilisation ConfirmSignUp avec un AWS SDK ou une CLI](#)
- [Utilisation CreateUserPool avec un AWS SDK ou une CLI](#)
- [Utilisation CreateUserPoolClient avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteUser avec un AWS SDK ou une CLI](#)
- [Utilisation ForgotPassword avec un AWS SDK ou une CLI](#)
- [Utilisation InitiateAuth avec un AWS SDK ou une CLI](#)
- [Utilisation ListUserPools avec un AWS SDK ou une CLI](#)
- [Utilisation ListUsers avec un AWS SDK ou une CLI](#)
- [Utilisation ResendConfirmationCode avec un AWS SDK ou une CLI](#)
- [Utilisation RespondToAuthChallenge avec un AWS SDK ou une CLI](#)
- [Utilisation SignUp avec un AWS SDK ou une CLI](#)

- [Utilisation UpdateUserPool avec un AWS SDK ou une CLI](#)
- [Utilisation VerifySoftwareToken avec un AWS SDK ou une CLI](#)

Utilisation **AdminCreateUser** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `AdminCreateUser`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito](#)

CLI

AWS CLI

Pour créer un utilisateur

L'`admin-create-user`exemple suivant crée un utilisateur avec l'adresse e-mail et le numéro de téléphone spécifiés dans les paramètres.

```
aws cognito-idp admin-create-user \  
  --user-pool-id us-west-2_aaaaaaaaa \  
  --username diego \  
  --user-attributes Name=email,Value=diego@example.com \  
  Name=phone_number,Value="+15555551212" \  
  --message-action SUPPRESS
```

Sortie :

```
{  
  "User": {  
    "Username": "diego",  
    "Attributes": [  
      {  
        "Name": "sub",  
        "Value": "7325c1de-b05b-4f84-b321-9adc6e61f4a2"  
      },  
    ],  
  },  
}
```




```
    {
      "Name": "phone_number",
      "Value": "+15555551212"
    },
    {
      "Name": "email",
      "Value": "diego@example.com"
    }
  ],
  "UserCreateDate": 1548099495.428,
  "UserLastModifiedDate": 1548099495.428,
  "Enabled": true,
  "UserStatus": "FORCE_CHANGE_PASSWORD"
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminCreateUser](#) à la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
type CognitoActions struct {
  CognitoClient *cognitoidentityprovider.Client
}

// AdminCreateUser uses administrator credentials to add a user to a user pool.
// This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(userPoolId string, userName string,
  userEmail string) error {
```

```
_, err := actor.CognitoClient.AdminCreateUser(context.TODO(),
&cognitoidentityprovider.AdminCreateUserInput{
    UserPoolId:      aws.String(userPoolId),
    Username:        aws.String(userName),
    MessageAction:   types.MessageActionTypeSuppress,
    UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
aws.String(userEmail)}}},
)})
if err != nil {
    var userExists *types.UsernameExistsException
    if errors.As(err, &userExists) {
        log.Printf("User %v already exists in the user pool.", userName)
        err = nil
    } else {
        log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
    }
}
return err
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminCreateUser](#) à la section Référence des AWS SDK for Go API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **AdminGetUser** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `AdminGetUser`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// Get the specified user from an Amazon Cognito user pool with
administrator access.
/// </summary>
/// <param name="userName">The name of the user.</param>
/// <param name="poolId">The Id of the Amazon Cognito user pool.</param>
/// <returns>Async task.</returns>
public async Task<UserStatusType> GetAdminUserAsync(string userName, string
poolId)
{
    AdminGetUserRequest userRequest = new AdminGetUserRequest
    {
        Username = userName,
        UserPoolId = poolId,
    };


    var response = await _cognitoService.AdminGetUserAsync(userRequest);

    Console.WriteLine($"User status {response.UserStatus}");
    return response.UserStatus;
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminGetUser](#) à la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

 Note

Il y en a plus à ce sujet [GitHub](#). Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

Aws::CognitoIdentityProvider::Model::AdminGetUserRequest request;
request.SetUsername(userName);
request.SetUserPoolId(userPoolID);

Aws::CognitoIdentityProvider::Model::AdminGetUserOutcome outcome =
    client.AdminGetUser(request);

if (outcome.IsSuccess()) {
    std::cout << "The status for " << userName << " is " <<

    Aws::CognitoIdentityProvider::Model::UserStatusTypeMapper::GetNameForUserStatusType(
        outcome.GetResult().GetUserStatus()) << std::endl;
    std::cout << "Enabled is " << outcome.GetResult().GetEnabled() <<
std::endl;
}
else {
    std::cerr << "Error with CognitoIdentityProvider::AdminGetUser. "
        << outcome.GetError().GetMessage()
        << std::endl;
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminGetUser](#) à la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour obtenir un utilisateur

Cet exemple permet d'obtenir des informations sur le nom d'utilisateur jane@example.com.

Commande :

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --username jane@example.com
```

Sortie :

```
{
  "Username": "4320de44-2322-4620-999b-5e2e1c8df013",
  "Enabled": true,
  "UserStatus": "FORCE_CHANGE_PASSWORD",
  "UserCreateDate": 1548108509.537,
  "UserAttributes": [
    {
      "Name": "sub",
      "Value": "4320de44-2322-4620-999b-5e2e1c8df013"
    },
    {
      "Name": "email_verified",
      "Value": "true"
    },
    {
      "Name": "phone_number_verified",
      "Value": "true"
    },
    {
      "Name": "phone_number",
      "Value": "+01115551212"
    },
    {
      "Name": "email",
      "Value": "jane@example.com"
    }
  ],
  "UserLastModifiedDate": 1548108509.537
}
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminGetUser](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
public static void getAdminUser(CognitoIdentityProviderClient
identityProviderClient, String userName,
    String poolId) {
    try {
        AdminGetUserRequest userRequest = AdminGetUserRequest.builder()
            .username(userName)
            .userPoolId(poolId)
            .build();

        AdminGetUserResponse response =
identityProviderClient.adminGetUser(userRequest);
        System.out.println("User status " + response.userStatusAsString());

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminGetUser](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const adminGetUser = ({ userPoolId, username }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new AdminGetUserCommand({
    UserPoolId: userPoolId,
    Username: username,
  });

  return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [AdminGetUser](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
suspend fun getAdminUser(userNameVal: String?, poolIdVal: String?) {
  val userRequest = AdminGetUserRequest {
    username = userNameVal
    userPoolId = poolIdVal
  }
}
```

```

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    val response = identityProviderClient.adminGetUser(userRequest)
    println("User status ${response.userStatus}")
}
}

```

- Pour plus de détails sur l'API, consultez [AdminGetUser](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret

```



```
def sign_up_user(self, user_name, password, user_email):
    """
    Signs up a new user with Amazon Cognito. This action prompts Amazon
    Cognito
    to send an email to the specified email address. The email contains a
    code that
    can be used to confirm the user.

    When the user already exists, the user status is checked to determine
    whether
    the user has been confirmed.

    :param user_name: The user name that identifies the new user.
    :param password: The password for the new user.
    :param user_email: The email address for the new user.
    :return: True when the user is already confirmed with Amazon Cognito.
            Otherwise, false.
    """
    try:
        kwargs = {
            "ClientId": self.client_id,
            "Username": user_name,
            "Password": password,
            "UserAttributes": [{"Name": "email", "Value": user_email}],
        }
        if self.client_secret is not None:
            kwargs["SecretHash"] = self._secret_hash(user_name)
        response = self.cognito_idp_client.sign_up(**kwargs)
        confirmed = response["UserConfirmed"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "UsernameExistsException":
            response = self.cognito_idp_client.admin_get_user(
                UserPoolId=self.user_pool_id, Username=user_name
            )
            logger.warning(
                "User %s exists and is %s.", user_name,
                response["UserStatus"]
            )
            confirmed = response["UserStatus"] == "CONFIRMED"
        else:
            logger.error(
                "Couldn't sign up %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
```

```
        err.response["Error"]["Message"],
    )
    raise
return confirmed
```

- Pour plus de détails sur l'API, consultez [AdminGetUser](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **AdminInitiateAuth** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `AdminInitiateAuth`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// Initiate an admin auth request.
/// </summary>
/// <param name="clientId">The client ID to use.</param>
/// <param name="userPoolId">The ID of the user pool.</param>
/// <param name="userName">The username to authenticate.</param>
```

```
/// <param name="password">The user's password.</param>
/// <returns>The session to use in challenge-response.</returns>
public async Task<string> AdminInitiateAuthAsync(string clientId, string
userPoolId, string userName, string password)
{
    var authParameters = new Dictionary<string, string>();
    authParameters.Add("USERNAME", userName);
    authParameters.Add("PASSWORD", password);

    var request = new AdminInitiateAuthRequest
    {
        ClientId = clientId,
        UserPoolId = userPoolId,
        AuthParameters = authParameters,
        AuthFlow = AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
    };

    var response = await _cognitoService.AdminInitiateAuthAsync(request);
    return response.Session;
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminInitiateAuth](#) à la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

Note

Il y en a plus à ce sujet [GitHub](#). Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);
```

```
Aws::CognitoIdentityProvider::Model::AdminInitiateAuthRequest request;
request.SetClientId(clientID);
request.SetUserPoolId(userPoolID);
request.AddAuthParameters("USERNAME", userName);
request.AddAuthParameters("PASSWORD", password);
request.SetAuthFlow(

Aws::CognitoIdentityProvider::Model::AuthFlowType::ADMIN_USER_PASSWORD_AUTH);

Aws::CognitoIdentityProvider::Model::AdminInitiateAuthOutcome outcome =
    client.AdminInitiateAuth(request);

if (outcome.IsSuccess()) {
    std::cout << "Call to AdminInitiateAuth was successful." << std::endl;
    sessionResult = outcome.GetResult().GetSession();
}
else {
    std::cerr << "Error with CognitoIdentityProvider::AdminInitiateAuth. "
              << outcome.GetError().GetMessage()
              << std::endl;
}
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminInitiateAuth](#) à la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour initier l'autorisation

Cet exemple initie l'autorisation à l'aide du flux ADMIN_NO_SRP_AUTH pour le nom d'utilisateur jane@example.com

Le client doit avoir activé l'API de connexion pour l'authentification basée sur le serveur (ADMIN_NO_SRP_AUTH).

Utilisez les informations de session contenues dans la valeur de retour pour appeler admin-respond-to-auth -challenge.

Commande :

```
aws cognito-idp admin-initiate-auth --user-pool-id us-west-2_aaaaaaaaa --client-id 3n4b5urk1ft4f13mg5e62d9ado --auth-flow ADMIN_NO_SRP_AUTH --auth-parameters USERNAME=jane@example.com,PASSWORD=password
```

Sortie :

```
{
  "ChallengeName": "NEW_PASSWORD_REQUIRED",
  "Session": "SESSION",
  "ChallengeParameters": {
    "USER_ID_FOR_SRP": "84514837-dcbc-4af1-abff-f3c109334894",
    "requiredAttributes": "[]",
    "userAttributes": "{\"email_verified\": \"true\", \"phone_number_verified\": \"true\", \"phone_number\": \"+01xxx5550100\", \"email\": \"jane@example.com\"}"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminInitiateAuth](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
public static AdminInitiateAuthResponse
initiateAuth(CognitoIdentityProviderClient identityProviderClient,
             String clientId, String userName, String password, String userPoolId)
{
    try {
        Map<String, String> authParameters = new HashMap<>();
        authParameters.put("USERNAME", userName);
        authParameters.put("PASSWORD", password);
    }
}
```

```
        AdminInitiateAuthRequest authRequest =
AdminInitiateAuthRequest.builder()
    .clientId(clientId)
    .userPoolId(userPoolId)
    .authParameters(authParameters)
    .authFlow(AuthFlowType.ADMIN_USER_PASSWORD_AUTH)
    .build();

        AdminInitiateAuthResponse response =
identityProviderClient.adminInitiateAuth(authRequest);
        System.out.println("Result Challenge is : " +
response.challengeName());
        return response;

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return null;
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminInitiateAuth](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const adminInitiateAuth = ({ clientId, userPoolId, username, password }) => {
    const client = new CognitoIdentityProviderClient({});

    const command = new AdminInitiateAuthCommand({
        ClientId: clientId,
```

```
UserPoolId: userPoolId,  
AuthFlow: AuthFlowType.ADMIN_USER_PASSWORD_AUTH,  
AuthParameters: { USERNAME: username, PASSWORD: password },  
});  
  
return client.send(command);  
};
```

- Pour plus de détails sur l'API, reportez-vous [AdminInitiateAuth](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
suspend fun checkAuthMethod(clientIdVal: String, userNameVal: String,  
passwordVal: String, userPoolIdVal: String): AdminInitiateAuthResponse {  
    val authParas = mutableMapOf<String, String>()  
    authParas["USERNAME"] = userNameVal  
    authParas["PASSWORD"] = passwordVal  
  
    val authRequest = AdminInitiateAuthRequest {  
        clientId = clientIdVal  
        userPoolId = userPoolIdVal  
        authParameters = authParas  
        authFlow = AuthFlowType.AdminUserPasswordAuth  
    }  
  
    CognitoIdentityProviderClient { region = "us-east-1" }.use  
{ identityProviderClient ->  
    val response = identityProviderClient.adminInitiateAuth(authRequest)  
    println("Result Challenge is ${response.challengeName}")  
    return response  
}
```

```
}
```

- Pour plus de détails sur l'API, consultez [AdminInitiateAuth](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
                 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
        client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
        pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret

    def start_sign_in(self, user_name, password):
        """
        Starts the sign-in process for a user by using administrator credentials.
        This method of signing in is appropriate for code running on a secure
        server.
```



```

in
    If the user pool is configured to require MFA and this is the first sign-
    for the user, Amazon Cognito returns a challenge response to set up an
    MFA application. When this occurs, this function gets an MFA secret from
    Amazon Cognito and returns it to the caller.

    :param user_name: The name of the user to sign in.
    :param password: The user's password.
    :return: The result of the sign-in attempt. When sign-in is successful,
this
        returns an access token that can be used to get AWS credentials.
Otherwise,
        Amazon Cognito returns a challenge to set up an MFA application,
        or a challenge to enter an MFA code from a registered MFA
application.
    """
    try:
        kwargs = {
            "UserPoolId": self.user_pool_id,
            "ClientId": self.client_id,
            "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
            "AuthParameters": {"USERNAME": user_name, "PASSWORD": password},
        }
        if self.client_secret is not None:
            kwargs["AuthParameters"]["SECRET_HASH"] =
self._secret_hash(user_name)
        response = self.cognito_idp_client.admin_initiate_auth(**kwargs)
        challenge_name = response.get("ChallengeName", None)
        if challenge_name == "MFA_SETUP":
            if (
                "SOFTWARE_TOKEN_MFA"
                in response["ChallengeParameters"]["MFAS_CAN_SETUP"]
            ):
                response.update(self.get_mfa_secret(response["Session"]))
            else:
                raise RuntimeError(
                    "The user pool requires MFA setup, but the user pool is
not "
                    "configured for TOTP MFA. This example requires TOTP
MFA."
                )
        except ClientError as err:
            logger.error(
                "Couldn't start sign in for %s. Here's why: %s: %s",

```

```
        user_name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    response.pop("ResponseMetadata", None)
    return response
```

- Pour plus de détails sur l'API, consultez [AdminInitiateAuth](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **AdminRespondToAuthChallenge** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `AdminRespondToAuthChallenge`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
```

```
/// Respond to an admin authentication challenge.
/// </summary>
/// <param name="userName">The name of the user.</param>
/// <param name="clientId">The client ID.</param>
/// <param name="mfaCode">The multi-factor authentication code.</param>
/// <param name="session">The current application session.</param>
/// <param name="clientId">The user pool ID.</param>
/// <returns>The result of the authentication response.</returns>
public async Task<AuthenticationResultType> AdminRespondToAuthChallengeAsync(
    string userName,
    string clientId,
    string mfaCode,
    string session,
    string userPoolId)
{
    Console.WriteLine("SOFTWARE_TOKEN_MFA challenge is generated");

    var challengeResponses = new Dictionary<string, string>();
    challengeResponses.Add("USERNAME", userName);
    challengeResponses.Add("SOFTWARE_TOKEN_MFA_CODE", mfaCode);


    var respondToAuthChallengeRequest = new
AdminRespondToAuthChallengeRequest
    {
        ChallengeName = ChallengeNameType.SOFTWARE_TOKEN_MFA,
        ClientId = clientId,
        ChallengeResponses = challengeResponses,
        Session = session,
        UserPoolId = userPoolId,
    };

    var response = await
_cognitoService.AdminRespondToAuthChallengeAsync(respondToAuthChallengeRequest);
    Console.WriteLine($"Response to Authentication
{response.AuthenticationResult.TokenType}");
    return response.AuthenticationResult;
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminRespondToAuthChallenge](#) à la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeRequest
request;
request.AddChallengeResponses("USERNAME", userName);
request.AddChallengeResponses("SOFTWARE_TOKEN_MFA_CODE", mfaCode);
request.SetChallengeName(

Aws::CognitoIdentityProvider::Model::ChallengeNameType::SOFTWARE_TOKEN_MFA);
request.SetClientId(clientID);
request.SetUserPoolId(userPoolID);
request.SetSession(session);

Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeOutcome
outcome =

    client.AdminRespondToAuthChallenge(request);

if (outcome.IsSuccess()) {
    std::cout << "Here is the response to the challenge.\n" <<

outcome.GetResult().GetAuthenticationResult().Jsonize().View().WriteReadable()
    << std::endl;

    accessToken =
outcome.GetResult().GetAuthenticationResult().GetAccessToken();
}
else {
```

```
        std::cerr << "Error with
CognitoIdentityProvider::AdminRespondToAuthChallenge. "
                << outcome.GetError().GetMessage()
                << std::endl;
    return false;
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminRespondToAuthChallenge](#) à la section Référence des AWS SDK for C++ API.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
// Respond to an authentication challenge.
public static void adminRespondToAuthChallenge(CognitoIdentityProviderClient
identityProviderClient,
        String userName, String clientId, String mfaCode, String session) {
    System.out.println("SOFTWARE_TOKEN_MFA challenge is generated");
    Map<String, String> challengeResponses = new HashMap<>();

    challengeResponses.put("USERNAME", userName);
    challengeResponses.put("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

    AdminRespondToAuthChallengeRequest respondToAuthChallengeRequest =
AdminRespondToAuthChallengeRequest.builder()
        .challengeName(ChallengeNameType.SOFTWARE_TOKEN_MFA)
        .clientId(clientId)
        .challengeResponses(challengeResponses)
        .session(session)
        .build();

    AdminRespondToAuthChallengeResponse respondToAuthChallengeResult =
identityProviderClient
```

```
        .adminRespondToAuthChallenge(respondToAuthChallengeRequest);

System.out.println("respondToAuthChallengeResult.getAuthenticationResult()"
    + respondToAuthChallengeResult.authenticationResult());
    }
```

- Pour plus de détails sur l'API, reportez-vous [AdminRespondToAuthChallenge](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const adminRespondToAuthChallenge = ({
  userPoolId,
  clientId,
  username,
  totp,
  session,
}) => {
  const client = new CognitoIdentityProviderClient({});
  const command = new AdminRespondToAuthChallengeCommand({
    ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
    ChallengeResponses: {
      SOFTWARE_TOKEN_MFA_CODE: totp,
      USERNAME: username,
    },
    ClientId: clientId,
    UserPoolId: userPoolId,
    Session: session,
  });

  return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [AdminRespondToAuthChallenge](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
// Respond to an authentication challenge.
suspend fun adminRespondToAuthChallenge(userName: String, clientIdVal: String?,
mfaCode: String, sessionVal: String?) {
    println("SOFTWARE_TOKEN_MFA challenge is generated")
    val challengeResponsesOb = mutableMapOf<String, String>()
    challengeResponsesOb["USERNAME"] = userName
    challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

    val adminRespondToAuthChallengeRequest = AdminRespondToAuthChallengeRequest {
        challengeName = ChallengeNameType.SoftwareTokenMfa
        clientId = clientIdVal
        challengeResponses = challengeResponsesOb
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
    { identityProviderClient ->
        val respondToAuthChallengeResult =
        identityProviderClient.adminRespondToAuthChallenge(adminRespondToAuthChallengeRequest)
        println("respondToAuthChallengeResult.getAuthenticationResult()
        ${respondToAuthChallengeResult.authenticationResult}")
    }
}
```

- Pour plus de détails sur l'API, consultez [AdminRespondToAuthChallenge](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Répondez à une stimulation MFA en fournissant un code généré par une application MFA associée.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
                 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
        client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
        pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret

    def respond_to_mfa_challenge(self, user_name, session, mfa_code):
        """
        Responds to a challenge for an MFA code. This completes the second step
        of
        a two-factor sign-in. When sign-in is successful, it returns an access
        token
```



```
that can be used to get AWS credentials from Amazon Cognito.

:param user_name: The name of the user who is signing in.
:param session: Session information returned from a previous call to
initiate
                authentication.
:param mfa_code: A code generated by the associated MFA application.
:return: The result of the authentication. When successful, this contains
an
        access token for the user.
"""
try:
    kwargs = {
        "UserPoolId": self.user_pool_id,
        "ClientId": self.client_id,
        "ChallengeName": "SOFTWARE_TOKEN_MFA",
        "Session": session,
        "ChallengeResponses": {
            "USERNAME": user_name,
            "SOFTWARE_TOKEN_MFA_CODE": mfa_code,
        },
    }
    if self.client_secret is not None:
        kwargs["ChallengeResponses"]["SECRET_HASH"] = self._secret_hash(
            user_name
        )
    response =
self.cognito_idp_client.admin_respond_to_auth_challenge(**kwargs)
    auth_result = response["AuthenticationResult"]
except ClientError as err:
    if err.response["Error"]["Code"] == "ExpiredCodeException":
        logger.warning(
            "Your MFA code has expired or has been used already. You
might have "
            "to wait a few seconds until your app shows you a new code."
        )
    else:
        logger.error(
            "Couldn't respond to mfa challenge for %s. Here's why: %s:
%s",
            user_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
```

```
        raise
    else:
        return auth_result
```

- Pour plus de détails sur l'API, consultez [AdminRespondToAuthChallenge](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **AdminSetUserPassword** avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser `AdminSetUserPassword`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito](#)

Go

Kit SDK for Go V2

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}
```

```
// AdminSetUserPassword uses administrator credentials to set a password for a
// user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(userPoolId string, userName
string, password string) error {
_, err := actor.CognitoClient.AdminSetUserPassword(context.TODO(),
&cognitoidentityprovider.AdminSetUserPasswordInput{
Password:  aws.String(password),
UserPoolId: aws.String(userPoolId),
Username:  aws.String(userName),
Permanent: true,
})
if err != nil {
var invalidPassword *types.InvalidPasswordException
if errors.As(err, &invalidPassword) {
log.Println(*invalidPassword.Message)
} else {
log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
err)
}
}
return err
}
```

- Pour plus de détails sur l'API, reportez-vous [AdminSetUserPassword](#) à la section Référence des AWS SDK for Go API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **AssociateSoftwareToken** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `AssociateSoftwareToken`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// Get an MFA token to authenticate the user with the authenticator.
/// </summary>
/// <param name="session">The session name.</param>
/// <returns>The session name.</returns>
public async Task<string> AssociateSoftwareTokenAsync(string session)
{
    var softwareTokenRequest = new AssociateSoftwareTokenRequest
    {
        Session = session,
    };

    var tokenResponse = await
        _cognitoService.AssociateSoftwareTokenAsync(softwareTokenRequest);
    var secretCode = tokenResponse.SecretCode;

    Console.WriteLine($"Use the following secret code to set up the
    authenticator: {secretCode}");

    return tokenResponse.Session;
}
```

- Pour plus de détails sur l'API, reportez-vous [AssociateSoftwareToken](#) à la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenRequest
request;
request.SetSession(session);

Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenOutcome
outcome =
    client.AssociateSoftwareToken(request);

if (outcome.IsSuccess()) {
    std::cout
        << "Enter this setup key into an authenticator app, for
example Google Authenticator."
        << std::endl;
    std::cout << "Setup key: " << outcome.GetResult().GetSecretCode()
        << std::endl;
#ifdef USING_QR
    printAsterisksLine();
    std::cout << "\nOr scan the QR code in the file '" << QR_CODE_PATH <<
    "."
        << std::endl;

    saveQRCode(std::string("otpauth://totp/" + userName + "?secret=" +
        outcome.GetResult().GetSecretCode());
#endif // USING_QR
    session = outcome.GetResult().GetSession();
```

```
    }
    else {
        std::cerr << "Error with
CognitoIdentityProvider::AssociateSoftwareToken. "
                << outcome.GetError().GetMessage()
                << std::endl;
        return false;
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [AssociateSoftwareToken](#) à la section Référence des AWS SDK for C++ API.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
public static String getSecretForAppMFA(CognitoIdentityProviderClient
identityProviderClient, String session) {
    AssociateSoftwareTokenRequest softwareTokenRequest =
AssociateSoftwareTokenRequest.builder()
        .session(session)
        .build();

    AssociateSoftwareTokenResponse tokenResponse = identityProviderClient
        .associateSoftwareToken(softwareTokenRequest);
    String secretCode = tokenResponse.secretCode();
    System.out.println("Enter this token into Google Authenticator");
    System.out.println(secretCode);
    return tokenResponse.session();
}
```

- Pour plus de détails sur l'API, reportez-vous [AssociateSoftwareToken](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const associateSoftwareToken = (session) => {
  const client = new CognitoIdentityProviderClient({});
  const command = new AssociateSoftwareTokenCommand({
    Session: session,
  });

  return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [AssociateSoftwareToken](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
suspend fun getSecretForAppMFA(sessionVal: String?): String? {
  val softwareTokenRequest = AssociateSoftwareTokenRequest {
    session = sessionVal
  }

  CognitoIdentityProviderClient { region = "us-east-1" }.use
  { identityProviderClient ->
```

```
        val tokenResponse =
identityProviderClient.associateSoftwareToken(softwareTokenRequest)
        val secretCode = tokenResponse.secretCode
        println("Enter this token into Google Authenticator")
        println(secretCode)
        return tokenResponse.session
    }
}
```

- Pour plus de détails sur l'API, consultez [AssociateSoftwareToken](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret
```



```
def get_mfa_secret(self, session):
    """
    Gets a token that can be used to associate an MFA application with the
    user.

    :param session: Session information returned from a previous call to
    initiate
                    authentication.
    :return: An MFA token that can be used to set up an MFA application.
    """
    try:
        response =
self.cognito_idp_client.associate_software_token(Session=session)
    except ClientError as err:
        logger.error(
            "Couldn't get MFA secret. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        response.pop("ResponseMetadata", None)
        return response
```

- Pour plus de détails sur l'API, consultez [AssociateSoftwareToken](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ConfirmDevice** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `ConfirmDevice`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// Initiates and confirms tracking of the device.
/// </summary>
/// <param name="accessToken">The user's access token.</param>
/// <param name="deviceKey">The key of the device from Amazon Cognito.</
param>
/// <param name="deviceName">The device name.</param>
/// <returns></returns>
public async Task<bool> ConfirmDeviceAsync(string accessToken, string
deviceKey, string deviceName)
{
    var request = new ConfirmDeviceRequest
    {
        AccessToken = accessToken,
        DeviceKey = deviceKey,
        DeviceName = deviceName
    };

    var response = await _cognitoService.ConfirmDeviceAsync(request);
    return response.UserConfirmationNecessary;
}
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmDevice](#) à la section Référence des AWS SDK for .NET API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const confirmDevice = ({ deviceKey, accessToken, passwordVerifier, salt }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmDeviceCommand({
    DeviceKey: deviceKey,
    AccessToken: accessToken,
    DeviceSecretVerifierConfig: {
      PasswordVerifier: passwordVerifier,
      Salt: salt,
    },
  });

  return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmDevice](#) à la section Référence des AWS SDK for JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class CognitoIdentityProviderWrapper:
```

```
"""Encapsulates Amazon Cognito actions"""

def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
    """
    :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
    :param user_pool_id: The ID of an existing Amazon Cognito user pool.
    :param client_id: The ID of a client application registered with the user
pool.
    :param client_secret: The client secret, if the client has a secret.
    """
    self.cognito_idp_client = cognito_idp_client
    self.user_pool_id = user_pool_id
    self.client_id = client_id
    self.client_secret = client_secret

def confirm_mfa_device(
    self,
    user_name,
    device_key,
    device_group_key,
    device_password,
    access_token,
    aws_srp,
):
    """
    Confirms an MFA device to be tracked by Amazon Cognito. When a device is
tracked, its key and password can be used to sign in without requiring a
new
MFA code from the MFA application.

    :param user_name: The user that is associated with the device.
    :param device_key: The key of the device, returned by Amazon Cognito.
    :param device_group_key: The group key of the device, returned by Amazon
Cognito.
    :param device_password: The password that is associated with the device.
    :param access_token: The user's access token.
    :param aws_srp: A class that helps with Secure Remote Password (SRP)
calculations. The scenario associated with this example
uses
the warrant package.
```

```

        :return: True when the user must confirm the device. Otherwise, False.
When
        False, the device is automatically confirmed and tracked.
"""
srp_helper = aws_srp.AWSSRP(
    username=user_name,
    password=device_password,
    pool_id="_",
    client_id=self.client_id,
    client_secret=None,
    client=self.cognito_idp_client,
)
device_and_pw = f"{device_group_key}{device_key}:{device_password}"
device_and_pw_hash = aws_srp.hash_sha256(device_and_pw.encode("utf-8"))
salt = aws_srp.pad_hex(aws_srp.get_random(16))
x_value = aws_srp.hex_to_long(aws_srp.hex_hash(salt +
device_and_pw_hash))
verifier = aws_srp.pad_hex(pow(srp_helper.val_g, x_value,
srp_helper.big_n))
device_secret_verifier_config = {
    "PasswordVerifier": base64.standard_b64encode(
        bytearray.fromhex(verifier)
    ).decode("utf-8"),
    "Salt":
base64.standard_b64encode(bytearray.fromhex(salt)).decode("utf-8"),
}
try:
    response = self.cognito_idp_client.confirm_device(
        AccessToken=access_token,
        DeviceKey=device_key,
        DeviceSecretVerifierConfig=device_secret_verifier_config,
    )
    user_confirm = response["UserConfirmationNecessary"]
except ClientError as err:
    logger.error(
        "Couldn't confirm mfa device %s. Here's why: %s: %s",
        device_key,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return user_confirm

```

- Pour plus de détails sur l'API, consultez [ConfirmDevice](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ConfirmForgotPassword** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `ConfirmForgotPassword`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Migrer automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)

CLI

AWS CLI

Pour confirmer un mot de passe oublié

Cet exemple confirme un mot de passe oublié pour le nom d'utilisateur `diego@example.com`.


Commande :

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4f13mg5e62d9ado --username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmForgotPassword](#) à la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}

// ConfirmForgotPassword confirms a user with a confirmation code and a new
// password.
func (actor CognitoActions) ConfirmForgotPassword(clientId string, code string,
    userName string, password string) error {
    _, err := actor.CognitoClient.ConfirmForgotPassword(context.TODO(),
    &cognitoidentityprovider.ConfirmForgotPasswordInput{
        ClientId:      aws.String(clientId),
        ConfirmationCode: aws.String(code),
        Password:      aws.String(password),
        Username:      aws.String(userName),
    })
    if err != nil {
        var invalidPassword *types.InvalidPasswordException
        if errors.As(err, &invalidPassword) {
            log.Println(*invalidPassword.Message)
        } else {
            log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
        }
    }
    return err
}
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmForgotPassword](#) à la section Référence des AWS SDK for Go API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ConfirmSignUp** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `ConfirmSignUp`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// Confirm that the user has signed up.
/// </summary>
/// <param name="clientId">The Id of this application.</param>
/// <param name="code">The confirmation code sent to the user.</param>
/// <param name="userName">The username.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ConfirmSignUpAsync(string clientId, string code,
string userName)
{
    var signUpRequest = new ConfirmSignUpRequest
    {
        ClientId = clientId,
```



```
        ConfirmationCode = code,
        Username = userName,
    };

    var response = await _cognitoService.ConfirmSignUpAsync(signUpRequest);
    if (response.HttpStatusCode == HttpStatusCode.OK)
    {
        Console.WriteLine($"{userName} was confirmed");
        return true;
    }
    return false;
}
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmSignUp](#) à la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

Aws::CognitoIdentityProvider::Model::ConfirmSignUpRequest request;
request.SetClientId(clientID);
request.SetConfirmationCode(confirmationCode);
request.SetUsername(userName);

Aws::CognitoIdentityProvider::Model::ConfirmSignUpOutcome outcome =
```

```
        client.ConfirmSignUp(request);

    if (outcome.IsSuccess()) {
        std::cout << "ConfirmSignup was Successful."
                  << std::endl;
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::ConfirmSignUp. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
        return false;
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmSignUp](#) à la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour confirmer l'inscription

Cet exemple confirme l'inscription pour le nom d'utilisateur `diego@example.com`.

Commande :

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4f13mg5e62d9ado --
username=diego@example.com --confirmation-code CONF_CODE
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmSignUp](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
public static void confirmSignUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String code,
    String userName) {
    try {
        ConfirmSignUpRequest signUpRequest = ConfirmSignUpRequest.builder()
            .clientId(clientId)
            .confirmationCode(code)
            .username(userName)
            .build();

        identityProviderClient.confirmSignUp(signUpRequest);
        System.out.println(userName + " was confirmed");

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmSignUp](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const confirmSignUp = ({ clientId, username, code }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmSignUpCommand({
    ClientId: clientId,
    Username: username,
    ConfirmationCode: code,
  });

  return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [ConfirmSignUp](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
suspend fun confirmSignUp(clientIdVal: String?, codeVal: String?, userNameVal:
String?) {
  val signUpRequest = ConfirmSignUpRequest {
    clientId = clientIdVal
```

```
        confirmationCode = codeVal
        username = userNameVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    identityProviderClient.confirmSignUp(signUpRequest)
    println("$userNameVal was confirmed")
}
}
```

- Pour plus de détails sur l'API, consultez [ConfirmSignUp](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
```

```
self.client_secret = client_secret

def confirm_user_sign_up(self, user_name, confirmation_code):
    """
    Confirms a previously created user. A user must be confirmed before they
    can sign in to Amazon Cognito.

    :param user_name: The name of the user to confirm.
    :param confirmation_code: The confirmation code sent to the user's
registered
                               email address.
    :return: True when the confirmation succeeds.
    """
    try:
        kwargs = {
            "ClientId": self.client_id,
            "Username": user_name,
            "ConfirmationCode": confirmation_code,
        }
        if self.client_secret is not None:
            kwargs["SecretHash"] = self._secret_hash(user_name)
        self.cognito_idp_client.confirm_sign_up(**kwargs)
    except ClientError as err:
        logger.error(
            "Couldn't confirm sign up for %s. Here's why: %s: %s",
            user_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return True
```

- Pour plus de détails sur l'API, consultez [ConfirmSignUp](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreateUserPool** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `CreateUserPool`.

CLI

AWS CLI

Pour créer un groupe d'utilisateurs configuré de manière minimale

Cet exemple crée un groupe d'utilisateurs nommé `MyUserPool` à l'aide des valeurs par défaut. Il n'y a aucun attribut obligatoire ni aucun client d'application. La MFA et la sécurité avancée sont désactivés.

Commande :

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

Sortie :

```
{
  "UserPool": {
    "SchemaAttributes": [
      {
        "Name": "sub",
        "StringAttributeConstraints": {
          "MinLength": "1",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": true,
        "AttributeDataType": "String",
        "Mutable": false
      },
      {
        "Name": "name",
        "StringAttributeConstraints": {
          "MinLength": "0",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "String",
```

```
    "Mutable": true
  },
  {
    "Name": "given_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "family_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "middle_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "nickname",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
```



```
    "Mutable": true
  },
  {
    "Name": "preferred_username",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "profile",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "picture",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "website",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
```

```
    "Mutable": true
  },
  {
    "Name": "email",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "email_verified",
    "Mutable": true
  },
  {
    "Name": "gender",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "birthdate",
    "StringAttributeConstraints": {
      "MinLength": "10",
      "MaxLength": "10"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "zoneinfo",
```

```
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "locale",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "phone_number",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "phone_number_verified",
    "Mutable": true
  },
  {
    "Name": "address",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
  },
```

```
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "String",
        "Mutable": true
    },
    {
        "Name": "updated_at",
        "NumberAttributeConstraints": {
            "MinValue": "0"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "Number",
        "Mutable": true
    }
],
"MfaConfiguration": "OFF",
"Name": "MyUserPool",
"LastModifiedDate": 1547833345.777,
"AdminCreateUserConfig": {
    "UnusedAccountValidityDays": 7,
    "AllowAdminCreateUserOnly": false
},
"EmailConfiguration": {},
"Policies": {
    "PasswordPolicy": {
        "RequireLowercase": true,
        "RequireSymbols": true,
        "RequireNumbers": true,
        "MinimumLength": 8,
        "RequireUppercase": true
    }
},
"CreationDate": 1547833345.777,
"EstimatedNumberOfUsers": 0,
"Id": "us-west-2_aaaaaaaaa",
"LambdaConfig": {}
}
}
```

Pour créer un groupe d'utilisateurs avec deux attributs requis

Cet exemple crée un groupe d'utilisateurs MyUserPool. Le groupe est configuré pour accepter l'e-mail en tant qu'attribut de nom d'utilisateur. Il définit également l'adresse e-mail source sur une adresse validée à l'aide d'Amazon Simple Email Service.

Commande :

```
aws cognito-idp create-user-pool --pool-name MyUserPool --username-attributes "email" --email-configuration=SourceArn="arn:aws:ses:us-east-1:111111111111:identity/jane@example.com",ReplyToEmailAddress="jane@example.com"
```

Sortie :

```
{
  "UserPool": {
    "SchemaAttributes": [
      {
        "Name": "sub",
        "StringAttributeConstraints": {
          "MinLength": "1",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": true,
        "AttributeDataType": "String",
        "Mutable": false
      },
      {
        "Name": "name",
        "StringAttributeConstraints": {
          "MinLength": "0",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "String",
        "Mutable": true
      },
      {
        "Name": "given_name",
        "StringAttributeConstraints": {
          "MinLength": "0",
          "MaxLength": "2048"
        }
      }
    ]
  }
}
```

```
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "family_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "middle_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "nickname",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "preferred_username",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    }
  }
}
```

```
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "profile",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "picture",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "website",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "email",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    }
  }
}
```

```
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "email_verified",
    "Mutable": true
  },
  {
    "Name": "gender",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "birthdate",
    "StringAttributeConstraints": {
      "MinLength": "10",
      "MaxLength": "10"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "zoneinfo",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
```



```
    "Mutable": true
  },
  {
    "Name": "locale",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "phone_number",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "phone_number_verified",
    "Mutable": true
  },
  {
    "Name": "address",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "updated_at",
```

```

        "NumberAttributeConstraints": {
            "MinValue": "0"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "Number",
        "Mutable": true
    }
],
"MfaConfiguration": "OFF",
"Name": "MyUserPool",
"LastModifiedDate": 1547837788.189,
"AdminCreateUserConfig": {
    "UnusedAccountValidityDays": 7,
    "AllowAdminCreateUserOnly": false
},
"EmailConfiguration": {
    "ReplyToEmailAddress": "jane@example.com",
    "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
jane@example.com"
},
"Policies": {
    "PasswordPolicy": {
        "RequireLowercase": true,
        "RequireSymbols": true,
        "RequireNumbers": true,
        "MinimumLength": 8,
        "RequireUppercase": true
    }
},
"UsernameAttributes": [
    "email"
],
"CreationDate": 1547837788.189,
"EstimatedNumberOfUsers": 0,
"Id": "us-west-2_aaaaaaaaa",
"LambdaConfig": {}
}
}

```

- Pour plus de détails sur l'API, reportez-vous [CreateUserPool](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderException;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolRequest;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class CreateUserPool {
    public static void main(String[] args) {

        final String usage = ""

            Usage:
                <userPoolName>\s

            Where:
                userPoolName - The name to give your user pool when it's
            created.

        """;
    }
}
```

```
    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String userPoolName = args[0];
    CognitoIdentityProviderClient cognitoClient =
CognitoIdentityProviderClient.builder()
        .region(Region.US_EAST_1)
        .build();

    String id = createPool(cognitoClient, userPoolName);
    System.out.println("User pool ID: " + id);
    cognitoClient.close();
}

public static String createPool(CognitoIdentityProviderClient cognitoClient,
String userPoolName) {
    try {
        CreateUserPoolRequest request = CreateUserPoolRequest.builder()
            .poolName(userPoolName)
            .build();

        CreateUserPoolResponse response =
cognitoClient.createUserPool(request);
        return response.userPool().id();

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
}
```

- Pour plus de détails sur l'API, reportez-vous [CreateUserPool](#) à la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation `CreateUserPoolClient` avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `CreateUserPoolClient`.

CLI

AWS CLI

Pour créer un client de groupe d'utilisateurs

Cet exemple crée un nouveau client de groupe d'utilisateurs avec deux flux d'autorisation explicites : `USER_PASSWORD_AUTH` et `ADMIN_NO_SRP_AUTH`.

Commande :

```
aws cognito-idp create-user-pool-client --user-pool-id us-west-2_aaaaaaaaaa
--client-name MyNewClient --no-generate-secret --explicit-auth-flows
"USER_PASSWORD_AUTH" "ADMIN_NO_SRP_AUTH"
```

Sortie :

```
{
  "UserPoolClient": {
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "ClientName": "MyNewClient",
    "ClientId": "6p3bs000no6a4ue1idruvd05ad",
    "LastModifiedDate": 1548697449.497,
    "CreationDate": 1548697449.497,
    "RefreshTokenValidity": 30,
    "ExplicitAuthFlows": [
      "USER_PASSWORD_AUTH",
      "ADMIN_NO_SRP_AUTH"
    ],
    "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

- Pour plus de détails sur l'API, reportez-vous [CreateUserPoolClient](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderException;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolClientRequest;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolClientResponse;

/**
 * A user pool client app is an application that authenticates with Amazon
 * Cognito user pools.
 * When you create a user pool, you can configure app clients that allow mobile
 * or web applications
 * to call API operations to authenticate users, manage user attributes and
 * profiles,
 * and implement sign-up and sign-in flows.
 *
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class CreateUserPoolClient {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <clientName> <userPoolId>\s
```

```
        Where:
            clientName - The name for the user pool client to create.
            userPoolId - The ID for the user pool.
        """;

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String clientName = args[0];
    String userPoolId = args[1];
    CognitoIdentityProviderClient cognitoClient =
CognitoIdentityProviderClient.builder()
        .region(Region.US_EAST_1)
        .build();

    createPoolClient(cognitoClient, clientName, userPoolId);
    cognitoClient.close();
}

    public static void createPoolClient(CognitoIdentityProviderClient
cognitoClient, String clientName,
        String userPoolId) {
    try {
        CreateUserPoolClientRequest request =
CreateUserPoolClientRequest.builder()
            .clientName(clientName)
            .userPoolId(userPoolId)
            .build();

        CreateUserPoolClientResponse response =
cognitoClient.createUserPoolClient(request);
        System.out.println("User pool " +
response.userPoolClient().clientName() + " created. ID: "
            + response.userPoolClient().clientId());

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Pour plus de détails sur l'API, reportez-vous [CreateUserPoolClient](#) à la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeleteUser** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `DeleteUser`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action dans son contexte dans les exemples de code suivants :

- [Confirmez automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)
- [Migrer automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)
- [Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito](#)

C++

SDK pour C++

Note

Il y en a plus à ce sujet [GitHub](#). Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);
```



```
Aws::CognitoIdentityProvider::Model::DeleteUserRequest request;
request.SetAccessToken(accessToken);

Aws::CognitoIdentityProvider::Model::DeleteUserOutcome outcome =
    client.DeleteUser(request);

if (outcome.IsSuccess()) {
    std::cout << "The user " << userName << " was deleted."
              << std::endl;
}
else {
    std::cerr << "Error with CognitoIdentityProvider::DeleteUser. "
              << outcome.GetError().GetMessage()
              << std::endl;
}
```

- Pour plus de détails sur l'API, reportez-vous [DeleteUser](#) à la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour supprimer un utilisateur

Cet exemple supprime un utilisateur.


Commande :

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- Pour plus de détails sur l'API, reportez-vous [DeleteUser](#) à la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}

// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(userAccessToken string) error {
    _, err := actor.CognitoClient.DeleteUser(context.TODO(),
        &cognitoidentityprovider.DeleteUserInput{
            AccessToken: aws.String(userAccessToken),
        })
    if err != nil {
        log.Printf("Couldn't delete user. Here's why: %v\n", err)
    }
    return err
}
```

- Pour plus de détails sur l'API, reportez-vous [DeleteUser](#) à la section Référence des AWS SDK for Go API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ForgotPassword** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `ForgotPassword`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Migrer automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)

CLI

AWS CLI

Pour forcer le changement de mot de passe

L'`forgot-password` exemple suivant envoie un message à `jane@example.com` pour modifier son mot de passe.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpqsnet7mpld0 --username jane@example.com
```

Sortie :

```
{
  "CodeDeliveryDetails": {
    "Destination": "j***@e***.com",
    "DeliveryMedium": "EMAIL",
    "AttributeName": "email"
  }
}
```

- Pour plus de détails sur l'API, reportez-vous [ForgotPassword](#) à la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}

// ForgotPassword starts a password recovery flow for a user. This flow typically
// sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(clientId string, userName string)
(*types.CodeDeliveryDetailsType, error) {
    output, err := actor.CognitoClient.ForgotPassword(context.TODO(),
&cognitoidentityprovider.ForgotPasswordInput{
    ClientId: aws.String(clientId),
    Username: aws.String(userName),
})
    if err != nil {
        log.Printf("Couldn't start password reset for user '%v'. Here's why: %v\n",
userName, err)
    }
    return output.CodeDeliveryDetails, err
}
```

- Pour plus de détails sur l'API, reportez-vous [ForgotPassword](#) à la section Référence des AWS SDK for Go API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **InitiateAuth** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `InitiateAuth`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- [Confirmez automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)
- [Migrer automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)
- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)
- [Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// Initiate authorization.
/// </summary>
/// <param name="clientId">The client Id of the application.</param>
/// <param name="userName">The name of the user who is authenticating.</
param>
/// <param name="password">The password for the user who is authenticating.</
param>
/// <returns>The response from the initiate auth request.</returns>
public async Task<InitiateAuthResponse> InitiateAuthAsync(string clientId,
string userName, string password)
{
    var authParameters = new Dictionary<string, string>();
    authParameters.Add("USERNAME", userName);
    authParameters.Add("PASSWORD", password);

    var authRequest = new InitiateAuthRequest

    {
        ClientId = clientId,
        AuthParameters = authParameters,
        AuthFlow = AuthFlowType.USER_PASSWORD_AUTH,
    };
};
```

```
var response = await _cognitoService.InitiateAuthAsync(authRequest);
Console.WriteLine($"Result Challenge is : {response.ChallengeName}");

return response;
}
```

- Pour plus de détails sur l'API, reportez-vous [InitiateAuth](#) à la section Référence des AWS SDK for .NET API.

Go

Kit SDK for Go V2

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}

// SignIn signs in a user to Amazon Cognito using a username and password
authentication flow.
func (actor CognitoActions) SignIn(clientId string, userName string, password
string) (*types.AuthenticationResultType, error) {
    var authResult *types.AuthenticationResultType
    output, err := actor.CognitoClient.InitiateAuth(context.TODO(),
&cognitoidentityprovider.InitiateAuthInput{
    AuthFlow:      "USER_PASSWORD_AUTH",
    ClientId:      aws.String(clientId),
    AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
})
    if err != nil {
        var resetRequired *types.PasswordResetRequiredException
        if errors.As(err, &resetRequired) {
```

```
    log.Println(*resetRequired.Message)
} else {
    log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
}
} else {
    authResult = output.AuthenticationResult
}
return authResult, err
}
```

- Pour plus de détails sur l'API, reportez-vous [InitiateAuth](#) à la section Référence des AWS SDK for Go API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const initiateAuth = ({ username, password, clientId }) => {
    const client = new CognitoIdentityProviderClient({});

    const command = new InitiateAuthCommand({
        AuthFlow: AuthFlowType.USER_PASSWORD_AUTH,
        AuthParameters: {
            USERNAME: username,
            PASSWORD: password,
        },
        ClientId: clientId,
    });

    return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [InitiateAuth](#) à la section Référence des AWS SDK for JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Cet exemple vous montre comment démarrer l'authentification avec un appareil suivi. Pour terminer la connexion, le client doit répondre correctement aux stimulations SRP (Secure Remote Password).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret

    def sign_in_with_tracked_device(
        self,
        user_name,
        password,
        device_key,
```



```

        device_group_key,
        device_password,
        aws_srp,
    ):
        """
        Signs in to Amazon Cognito as a user who has a tracked device. Signing in
        with a tracked device lets a user sign in without entering a new MFA
code.

        Signing in with a tracked device requires that the client respond to the
SRP
        protocol. The scenario associated with this example uses the warrant
package
        to help with SRP calculations.

        For more information on SRP, see https://en.wikipedia.org/wiki/Secure\_Remote\_Password\_protocol.

        :param user_name: The user that is associated with the device.
        :param password: The user's password.
        :param device_key: The key of a tracked device.
        :param device_group_key: The group key of a tracked device.
        :param device_password: The password that is associated with the device.
        :param aws_srp: A class that helps with SRP calculations. The scenario
            associated with this example uses the warrant package.
        :return: The result of the authentication. When successful, this contains
an
            access token for the user.
        """
        try:
            srp_helper = aws_srp.AWSSRP(
                username=user_name,
                password=device_password,
                pool_id="",
                client_id=self.client_id,
                client_secret=None,
                client=self.cognito_idp_client,
            )

            response_init = self.cognito_idp_client.initiate_auth(
                ClientId=self.client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": user_name,

```

```
        "PASSWORD": password,
        "DEVICE_KEY": device_key,
    },
)
if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
    raise RuntimeError(
        f"Expected DEVICE_SRP_AUTH challenge but got
{response_init['ChallengeName']}."
    )

auth_params = srp_helper.get_auth_params()
auth_params["DEVICE_KEY"] = device_key
response_auth = self.cognito_idp_client.respond_to_auth_challenge(
    ClientId=self.client_id,
    ChallengeName="DEVICE_SRP_AUTH",
    ChallengeResponses=auth_params,
)
if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
    raise RuntimeError(
        f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
        f"{response_init['ChallengeName']}."
    )

challenge_params = response_auth["ChallengeParameters"]
challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
user_name})
cr["USERNAME"] = user_name
cr["DEVICE_KEY"] = device_key
response_verifier =
self.cognito_idp_client.respond_to_auth_challenge(
    ClientId=self.client_id,
    ChallengeName="DEVICE_PASSWORD_VERIFIER",
    ChallengeResponses=cr,
)
auth_tokens = response_verifier["AuthenticationResult"]
except ClientError as err:
    logger.error(
        "Couldn't start client sign in for %s. Here's why: %s: %s",
        user_name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
raise
```

```
else:
    return auth_tokens
```

- Pour plus de détails sur l'API, consultez [InitiateAuth](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ListUserPools** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `ListUserPools`.

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// List the Amazon Cognito user pools for an account.
/// </summary>
/// <returns>A list of UserPoolDescriptionType objects.</returns>
public async Task<List<UserPoolDescriptionType>> ListUserPoolsAsync()
{
    var userPools = new List<UserPoolDescriptionType>();

    var userPoolsPaginator = _cognitoService.Paginators.ListUserPools(new
ListUserPoolsRequest());

    await foreach (var response in userPoolsPaginator.Responses)
    {
        userPools.AddRange(response.UserPools);
    }
}
```

```
    }  
  
    return userPools;  
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUserPools](#) à la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour afficher les groupes d'utilisateurs

Cet exemple répertorie jusqu'à 20 groupes d'utilisateurs.

Commande :

```
aws cognito-idp list-user-pools --max-results 20
```

Sortie :

```
{  
  "UserPools": [  
    {  
      "CreationDate": 1547763720.822,  
      "LastModifiedDate": 1547763720.822,  
      "LambdaConfig": {},  
      "Id": "us-west-2_aaaaaaaaa",  
      "Name": "MyUserPool"  
    }  
  ]  
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUserPools](#) à la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
package main

import (
    "context"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
    "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Notification
// Service
// (Amazon SNS) client and list the topics in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        fmt.Println("Couldn't load default configuration. Have you set up your AWS
account?")
        fmt.Println(err)
        return
    }
    cognitoClient := cognitoidentityprovider.NewFromConfig(sdkConfig)
    fmt.Println("Let's list the user pools for your account.")
    var pools []types.UserPoolDescriptionType
    paginator := cognitoidentityprovider.NewListUserPoolsPaginator(
```

```
cognitoClient, &cognitoidentityprovider.ListUserPoolsInput{MaxResults:
aws.Int32(10)})
for paginator.HasMorePages() {
    output, err := paginator.NextPage(context.TODO())
    if err != nil {
        log.Printf("Couldn't get user pools. Here's why: %v\n", err)
    } else {
        pools = append(pools, output.UserPools...)
    }
}
if len(pools) == 0 {
    fmt.Println("You don't have any user pools!")
} else {
    for _, pool := range pools {
        fmt.Printf("\t\t%v: %v\n", *pool.Name, *pool.Id)
    }
}
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUserPools](#) à la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderException;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsResponse;
```

```
import
software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListUserPools {
    public static void main(String[] args) {
        CognitoIdentityProviderClient cognitoClient =
CognitoIdentityProviderClient.builder()
            .region(Region.US_EAST_1)
            .build();

        listAllUserPools(cognitoClient);
        cognitoClient.close();
    }

    public static void listAllUserPools(CognitoIdentityProviderClient
cognitoClient) {
        try {
            ListUserPoolsRequest request = ListUserPoolsRequest.builder()
                .maxResults(10)
                .build();

            ListUserPoolsResponse response =
cognitoClient.listUserPools(request);
            response.userPools().forEach(userpool -> {
                System.out.println("User pool " + userpool.name() + ", User ID "
+ userpool.id());
            });

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUserPools](#) à la section Référence des AWS SDK for Java 2.x API.

Rust

SDK pour Rust

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
async fn show_pools(client: &Client) -> Result<(), Error> {
    let response = client.list_user_pools().max_results(10).send().await?;
    let pools = response.user_pools();
    println!("User pools:");
    for pool in pools {
        println!(" ID:           {}", pool.id().unwrap_or_default());
        println!(" Name:           {}", pool.name().unwrap_or_default());
        println!(" Lambda Config:  {:?}", pool.lambda_config().unwrap());
        println!(
            "   Last modified:  {}",
            pool.last_modified_date().unwrap().to_chrono_utc()?
        );
        println!(
            "   Creation date:   {:?}",
            pool.creation_date().unwrap().to_chrono_utc()
        );
        println!();
    }
    println!("Next token: {}", response.next_token().unwrap_or_default());

    Ok(())
}
```

- Pour plus de détails sur l'API, voir [ListUserPools](#) la section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ListUsers** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `ListUsers`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// Get a list of users for the Amazon Cognito user pool.
/// </summary>
/// <param name="userPoolId">The user pool ID.</param>
/// <returns>A list of users.</returns>
public async Task<List<UserType>> ListUsersAsync(string userPoolId)
{
    var request = new ListUsersRequest
    {
        UserPoolId = userPoolId
    };

    var users = new List<UserType>();

    var usersPaginator = _cognitoService.Paginators.ListUsers(request);
    await foreach (var response in usersPaginator.Responses)
    {
```

```
        users.AddRange(response.Users);
    }

    return users;
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUsers](#) à la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour afficher les utilisateurs

Cet exemple répertorie jusqu'à 20 utilisateurs.

Commande :

```
aws cognito-idp list-users --user-pool-id us-west-2_aaaaaaaaa --limit 20
```

Sortie :

```
{
  "Users": [
    {
      "Username": "22704aa3-fc10-479a-97eb-2af5806bd327",
      "Enabled": true,
      "UserStatus": "FORCE_CHANGE_PASSWORD",
      "UserCreateDate": 1548089817.683,
      "UserLastModifiedDate": 1548089817.683,
      "Attributes": [
        {
          "Name": "sub",
          "Value": "22704aa3-fc10-479a-97eb-2af5806bd327"
        },
        {
          "Name": "email_verified",
          "Value": "true"
        },
        {
```

```
        "Name": "email",
        "Value": "mary@example.com"
    }
  ]
}
]
```

- Pour plus de détails sur l'API, reportez-vous [ListUsers](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import
  software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
  software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderException;
import
  software.amazon.awssdk.services.cognitoidentityprovider.model.ListUsersRequest;
import
  software.amazon.awssdk.services.cognitoidentityprovider.model.ListUsersResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ListUsers {
```

```
public static void main(String[] args) {

    final String usage = ""

        Usage:
            <userPoolId>\s

        Where:
            userPoolId - The ID given to your user pool when it's
created.

        """;

    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String userPoolId = args[0];
    CognitoIdentityProviderClient cognitoClient =
CognitoIdentityProviderClient.builder()
        .region(Region.US_EAST_1)
        .build();

    listAllUsers(cognitoClient, userPoolId);
    listUsersFilter(cognitoClient, userPoolId);
    cognitoClient.close();
}

public static void listAllUsers(CognitoIdentityProviderClient cognitoClient,
String userPoolId) {
    try {
        ListUsersRequest usersRequest = ListUsersRequest.builder()
            .userPoolId(userPoolId)
            .build();

        ListUsersResponse response = cognitoClient.listUsers(usersRequest);
        response.users().forEach(user -> {
            System.out.println("User " + user.username() + " Status " +
user.userStatus() + " Created "
                + user.userCreateDate());
        });

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
    }
}
```

```
        System.exit(1);
    }
}

// Shows how to list users by using a filter.
public static void listUsersFilter(CognitoIdentityProviderClient
cognitoClient, String userPoolId) {

    try {
        String filter = "email = \"tblue@noserver.com\"";
        ListUsersRequest usersRequest = ListUsersRequest.builder()
            .userPoolId(userPoolId)
            .filter(filter)
            .build();

        ListUsersResponse response = cognitoClient.listUsers(usersRequest);
        response.users().forEach(user -> {
            System.out.println("User with filter applied " + user.username()
+ " Status " + user.userStatus()
            + " Created " + user.userCreateDate());
        });

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Pour plus de détails sur l'API, reportez-vous [ListUsers](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const listUsers = ({ userPoolId }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ListUsersCommand({
    UserPoolId: userPoolId,
  });

  return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [ListUsers](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
suspend fun listAllUsers(userPoolId: String) {

  val request = ListUsersRequest {
    this.userPoolId = userPoolId
  }

  CognitoIdentityProviderClient { region = "us-east-1" }.use { cognitoClient ->
    val response = cognitoClient.listUsers(request)
    response.users?.forEach { user ->
      println("The user name is ${user.username}")
    }
  }
}
```

- Pour plus de détails sur l'API, consultez [ListUsers](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
                 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
        client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
        pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret

    def list_users(self):
        """
        Returns a list of the users in the current user pool.

        :return: The list of users.
        """
        try:
            response =
self.cognito_idp_client.list_users(UserPoolId=self.user_pool_id)
```

```
        users = response["Users"]
    except ClientError as err:
        logger.error(
            "Couldn't list users for %s. Here's why: %s: %s",
            self.user_pool_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return users
```

- Pour plus de détails sur l'API, consultez [ListUsers](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ResendConfirmationCode** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `ResendConfirmationCode`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).


```
/// <summary>
/// Send a new confirmation code to a user.
/// </summary>
/// <param name="clientId">The Id of the client application.</param>
/// <param name="userName">The username of user who will receive the code.</
param>
/// <returns>The delivery details.</returns>
public async Task<CodeDeliveryDetailsType> ResendConfirmationCodeAsync(string
clientId, string userName)
{
    var codeRequest = new ResendConfirmationCodeRequest
    {
        ClientId = clientId,
        Username = userName,
    };

    var response = await
_cognitoService.ResendConfirmationCodeAsync(codeRequest);

    Console.WriteLine($"Method of delivery is
{response.CodeDeliveryDetails.DeliveryMedium}");

    return response.CodeDeliveryDetails;
}
```

- Pour plus de détails sur l'API, reportez-vous [ResendConfirmationCode](#) à la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
```

```
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

    Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeRequest
request;
    request.SetUsername(userName);
    request.SetClientId(clientID);

    Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeOutcome
outcome =
        client.ResendConfirmationCode(request);

    if (outcome.IsSuccess()) {
        std::cout
            << "CognitoIdentityProvider::ResendConfirmationCode was
successful."
            << std::endl;
    }
    else {
        std::cerr << "Error with
CognitoIdentityProvider::ResendConfirmationCode. "
            << outcome.GetError().GetMessage()
            << std::endl;
        return false;
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [ResendConfirmationCode](#) à la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour renvoyer un code de confirmation

L'exemple `resend-confirmation-code` suivant envoie un code de confirmation à l'utilisateur `jane`.

```
aws cognito-idp resend-confirmation-code \
```

```
--client-id 12a3b456c7de890f11g123hijk \  
--username jane
```

Sortie :

```
{  
  "CodeDeliveryDetails": {  
    "Destination": "j***@e***.com",  
    "DeliveryMedium": "EMAIL",  
    "AttributeName": "email"  
  }  
}
```

Pour plus d'informations, consultez [Inscription et confirmation de comptes utilisateur](#) dans le Guide du développeur Amazon Cognito.

- Pour plus de détails sur l'API, reportez-vous [ResendConfirmationCode](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
public static void resendConfirmationCode(CognitoIdentityProviderClient  
identityProviderClient, String clientId,  
    String userName) {  
    try {  
        ResendConfirmationCodeRequest codeRequest =  
ResendConfirmationCodeRequest.builder()  
            .clientId(clientId)  
            .username(userName)  
            .build();  
  
        ResendConfirmationCodeResponse response =  
identityProviderClient.resendConfirmationCode(codeRequest);
```

```
        System.out.println("Method of delivery is " +
response.codeDeliveryDetails().deliveryMediumAsString());

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [ResendConfirmationCode](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const resendConfirmationCode = ({ clientId, username }) => {
    const client = new CognitoIdentityProviderClient({});

    const command = new ResendConfirmationCodeCommand({
        ClientId: clientId,
        Username: username,
    });

    return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [ResendConfirmationCode](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
suspend fun resendConfirmationCode(clientIdVal: String?, userNameVal: String?) {
    val codeRequest = ResendConfirmationCodeRequest {
        clientId = clientIdVal
        username = userNameVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    val response = identityProviderClient.resendConfirmationCode(codeRequest)
    println("Method of delivery is " +
(response.codeDeliveryDetails?.deliveryMedium))
    }
}
```

- Pour plus de détails sur l'API, consultez [ResendConfirmationCode](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""
```

```
def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
    """
    :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
    :param user_pool_id: The ID of an existing Amazon Cognito user pool.
    :param client_id: The ID of a client application registered with the user
pool.
    :param client_secret: The client secret, if the client has a secret.
    """
    self.cognito_idp_client = cognito_idp_client
    self.user_pool_id = user_pool_id
    self.client_id = client_id
    self.client_secret = client_secret

def resend_confirmation(self, user_name):
    """
    Prompts Amazon Cognito to resend an email with a new confirmation code.

    :param user_name: The name of the user who will receive the email.
    :return: Delivery information about where the email is sent.
    """
    try:
        kwargs = {"ClientId": self.client_id, "Username": user_name}
        if self.client_secret is not None:
            kwargs["SecretHash"] = self._secret_hash(user_name)
        response = self.cognito_idp_client.resend_confirmation_code(**kwargs)
        delivery = response["CodeDeliveryDetails"]
    except ClientError as err:
        logger.error(
            "Couldn't resend confirmation to %s. Here's why: %s: %s",
            user_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return delivery
```

- Pour plus de détails sur l'API, consultez [ResendConfirmationCode](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **RespondToAuthChallenge** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `RespondToAuthChallenge`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

CLI

AWS CLI

Pour répondre à une stimulation d'authentification

Cet exemple répond à une stimulation d'autorisation lancée avec `initiate-auth`. Il s'agit d'une réponse à la stimulation `NEW_PASSWORD_REQUIRED`. Elle définit un mot de passe pour l'utilisateur `jane@example.com`.

Commande :

```
aws cognito-idp respond-to-auth-challenge --client-id 3n4b5urk1ft4fl3mg5e62d9ado
--challenge-name NEW_PASSWORD_REQUIRED --challenge-responses
USERNAME=jane@example.com,NEW_PASSWORD="password" --session "SESSION_TOKEN"
```

Sortie :

```
{
  "ChallengeParameters": {},
  "AuthenticationResult": {
    "AccessToken": "ACCESS_TOKEN",
    "ExpiresIn": 3600,
    "TokenType": "Bearer",
```

```
"RefreshToken": "REFRESH_TOKEN",
"IdToken": "ID_TOKEN",
"NewDeviceMetadata": {
  "DeviceKey": "us-west-2_fec070d2-fa88-424a-8ec8-b26d7198eb23",
  "DeviceGroupKey": "-wt2ha1Zd"
}
}
}
```

- Pour plus de détails sur l'API, reportez-vous [RespondToAuthChallenge](#) à la section Référence des AWS CLI commandes.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const respondToAuthChallenge = ({
  clientId,
  username,
  session,
  userPoolId,
  code,
}) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new RespondToAuthChallengeCommand({
    ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
    ChallengeResponses: {
      SOFTWARE_TOKEN_MFA_CODE: code,
      USERNAME: username,
    },
    ClientId: clientId,
    UserPoolId: userPoolId,
    Session: session,
  });
```



```
return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [RespondToAuthChallenge](#) à la section Référence des AWS SDK for JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Connectez-vous à l'aide d'un appareil suivi. Pour terminer la connexion, le client doit répondre correctement aux stimulations SRP (Secure Remote Password).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret

    def sign_in_with_tracked_device(
```

```
self,
user_name,
password,
device_key,
device_group_key,
device_password,
aws_srp,
):
    """
    Signs in to Amazon Cognito as a user who has a tracked device. Signing in
    with a tracked device lets a user sign in without entering a new MFA
code.

    Signing in with a tracked device requires that the client respond to the
SRP
    protocol. The scenario associated with this example uses the warrant
package
    to help with SRP calculations.

    For more information on SRP, see https://en.wikipedia.org/wiki/Secure\_Remote\_Password\_protocol.

    :param user_name: The user that is associated with the device.
    :param password: The user's password.
    :param device_key: The key of a tracked device.
    :param device_group_key: The group key of a tracked device.
    :param device_password: The password that is associated with the device.
    :param aws_srp: A class that helps with SRP calculations. The scenario
        associated with this example uses the warrant package.
    :return: The result of the authentication. When successful, this contains
an
        access token for the user.
    """
    try:
        srp_helper = aws_srp.AWSSRP(
            username=user_name,
            password=device_password,
            pool_id="_",
            client_id=self.client_id,
            client_secret=None,
            client=self.cognito_idp_client,
        )

        response_init = self.cognito_idp_client.initiate_auth(
```

```
        ClientId=self.client_id,
        AuthFlow="USER_PASSWORD_AUTH",
        AuthParameters={
            "USERNAME": user_name,
            "PASSWORD": password,
            "DEVICE_KEY": device_key,
        },
    )
    if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
        raise RuntimeError(
            f"Expected DEVICE_SRP_AUTH challenge but got {response_init['ChallengeName']}."
        )

    auth_params = srp_helper.get_auth_params()
    auth_params["DEVICE_KEY"] = device_key
    response_auth = self.cognito_idp_client.respond_to_auth_challenge(
        ClientId=self.client_id,
        ChallengeName="DEVICE_SRP_AUTH",
        ChallengeResponses=auth_params,
    )
    if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
        raise RuntimeError(
            f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
            f"{response_init['ChallengeName']}."
        )

    challenge_params = response_auth["ChallengeParameters"]
    challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
    cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
user_name})
    cr["USERNAME"] = user_name
    cr["DEVICE_KEY"] = device_key
    response_verifier =
self.cognito_idp_client.respond_to_auth_challenge(
        ClientId=self.client_id,
        ChallengeName="DEVICE_PASSWORD_VERIFIER",
        ChallengeResponses=cr,
    )
    auth_tokens = response_verifier["AuthenticationResult"]
except ClientError as err:
    logger.error(
        "Couldn't start client sign in for %s. Here's why: %s: %s",
        user_name,
```

```
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return auth_tokens
```

- Pour plus de détails sur l'API, consultez [RespondToAuthChallenge](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **SignUp** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `SignUp`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- [Confirmez automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)
- [Migrer automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)
- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
```

```
/// Sign up a new user.
/// </summary>
/// <param name="clientId">The client Id of the application.</param>
/// <param name="userName">The username to use.</param>
/// <param name="password">The user's password.</param>
/// <param name="email">The email address of the user.</param>
/// <returns>A Boolean value indicating whether the user was confirmed.</
returns>
public async Task<bool> SignUpAsync(string clientId, string userName, string
password, string email)
{
    var userAttrs = new AttributeType
    {
        Name = "email",
        Value = email,
    };

    var userAttrsList = new List<AttributeType>();

    userAttrsList.Add(userAttrs);


    var signUpRequest = new SignUpRequest
    {
        UserAttributes = userAttrsList,
        Username = userName,
        ClientId = clientId,
        Password = password
    };

    var response = await _cognitoService.SignUpAsync(signUpRequest);
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- Pour plus de détails sur l'API, reportez-vous [SignUp](#) à la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

Aws::CognitoIdentityProvider::Model::SignUpRequest request;
request.AddUserAttributes(
    Aws::CognitoIdentityProvider::Model::AttributeType().WithName(
        "email").WithValue(email));
request.SetUsername(userName);
request.SetPassword(password);
request.SetClientId(clientID);
Aws::CognitoIdentityProvider::Model::SignUpOutcome outcome =
    client.SignUp(request);

if (outcome.IsSuccess()) {
    std::cout << "The signup request for " << userName << " was
successful."
                << std::endl;
}
else if (outcome.GetError().GetErrorType() ==
Aws::CognitoIdentityProvider::CognitoIdentityProviderErrors::USERNAME_EXISTS) {
    std::cout
        << "The username already exists. Please enter a different
username."
        << std::endl;
    userExists = true;
}
else {
```

```
        std::cerr << "Error with CognitoIdentityProvider::SignUpRequest. "
                << outcome.GetError().GetMessage()
                << std::endl;
    return false;
}
```

- Pour plus de détails sur l'API, reportez-vous [SignUp](#) à la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour inscrire un utilisateur

Cet exemple inscrit jane@example.com.

Commande :

```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4f13mg5e62d9ado --
username jane@example.com --password PASSWORD --user-attributes
Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

Sortie :

```
{
  "UserConfirmed": false,
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"
}
```

- Pour plus de détails sur l'API, reportez-vous [SignUp](#) à la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}

// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(clientId string, userName string, password
string, userEmail string) (bool, error) {
    confirmed := false
    output, err := actor.CognitoClient.SignUp(context.TODO(),
&cognitoidentityprovider.SignUpInput{
    ClientId: aws.String(clientId),
    Password: aws.String(password),
    Username: aws.String(userName),
    UserAttributes: []types.AttributeType{
        {Name: aws.String("email"), Value: aws.String(userEmail)},
    },
})
    if err != nil {
        var invalidPassword *types.InvalidPasswordException
        if errors.As(err, &invalidPassword) {
            log.Println(*invalidPassword.Message)
        } else {
            log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
        }
    } else {
        confirmed = output.UserConfirmed
    }
    return confirmed, err
}
```


- Pour plus de détails sur l'API, reportez-vous [SignUp](#) à la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
public static void signUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String userName,
    String password, String email) {
    AttributeType userAttrs = AttributeType.builder()
        .name("email")
        .value(email)
        .build();

    List<AttributeType> userAttrsList = new ArrayList<>();
    userAttrsList.add(userAttrs);
    try {
        SignUpRequest signUpRequest = SignUpRequest.builder()
            .userAttributes(userAttrsList)
            .username(userName)
            .clientId(clientId)
            .password(password)
            .build();

        identityProviderClient.signUp(signUpRequest);
        System.out.println("User has been signed up ");
    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous [SignUp](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const signUp = ({ clientId, username, password, email }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new SignUpCommand({
    ClientId: clientId,
    Username: username,
    Password: password,
    UserAttributes: [{ Name: "email", Value: email }],
  });

  return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [SignUp](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
suspend fun signUp(clientIdVal: String?, userNameVal: String?, passwordVal:
String?, emailVal: String?) {
    val userAttrs = AttributeType {
        name = "email"
        value = emailVal
    }

    val userAttrsList = mutableListOf<AttributeType>()
    userAttrsList.add(userAttrs)
    val signUpRequest = SignUpRequest {
        userAttributes = userAttrsList
        username = userNameVal
        clientId = clientIdVal
        password = passwordVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    identityProviderClient.signUp(signUpRequest)
    println("User has been signed up")
}
}
```

- Pour plus de détails sur l'API, consultez [SignUp](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
                 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
        client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
        pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret

    def sign_up_user(self, user_name, password, user_email):
        """
        Signs up a new user with Amazon Cognito. This action prompts Amazon
        Cognito
        to send an email to the specified email address. The email contains a
        code that
        can be used to confirm the user.

        When the user already exists, the user status is checked to determine
        whether
        the user has been confirmed.

        :param user_name: The user name that identifies the new user.
```

```
:param password: The password for the new user.
:param user_email: The email address for the new user.
:return: True when the user is already confirmed with Amazon Cognito.
        Otherwise, false.
"""
try:
    kwargs = {
        "ClientId": self.client_id,
        "Username": user_name,
        "Password": password,
        "UserAttributes": [{"Name": "email", "Value": user_email}],
    }
    if self.client_secret is not None:
        kwargs["SecretHash"] = self._secret_hash(user_name)
    response = self.cognito_idp_client.sign_up(**kwargs)
    confirmed = response["UserConfirmed"]
except ClientError as err:
    if err.response["Error"]["Code"] == "UsernameExistsException":
        response = self.cognito_idp_client.admin_get_user(
            UserPoolId=self.user_pool_id, Username=user_name
        )
        logger.warning(
            "User %s exists and is %s.", user_name,
            response["UserStatus"]
        )
        confirmed = response["UserStatus"] == "CONFIRMED"
    else:
        logger.error(
            "Couldn't sign up %s. Here's why: %s: %s",
            user_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
return confirmed
```

- Pour plus de détails sur l'API, consultez [SignUp](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **UpdateUserPool** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser `UpdateUserPool`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- [Confirmez automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)
- [Migrer automatiquement les utilisateurs connus à l'aide d'une fonction Lambda](#)
- [Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito](#)

CLI

AWS CLI

Pour mettre à jour un groupe d'utilisateurs

Cet exemple ajoute des balises à un groupe d'utilisateurs.

Commande :

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_aaaaaaaaa --user-pool-tags Team=Blue,Area=West
```

- Pour plus de détails sur l'API, reportez-vous [UpdateUserPool](#) à la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}

// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
// trigger.
type Trigger int

const (
    PreSignUp Trigger = iota
    UserMigration
    PostAuthentication
)

type TriggerInfo struct {
    Trigger    Trigger
    HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
// is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(userPoolId string,
    triggers ...TriggerInfo) error {
    output, err := actor.CognitoClient.DescribeUserPool(context.TODO(),
    &cognitoidentityprovider.DescribeUserPoolInput{
        UserPoolId: aws.String(userPoolId),
    })
    if err != nil {
```

```
    log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
userPoolId, err)
    return err
}
lambdaConfig := output.UserPool.LambdaConfig
for _, trigger := range triggers {
    switch trigger.Trigger {
    case PreSignUp:
        lambdaConfig.PreSignUp = trigger.HandlerArn
    case UserMigration:
        lambdaConfig.UserMigration = trigger.HandlerArn
    case PostAuthentication:
        lambdaConfig.PostAuthentication = trigger.HandlerArn
    }
}
_, err = actor.CognitoClient.UpdateUserPool(context.TODO(),
&cognitoidentityprovider.UpdateUserPoolInput{
    UserPoolId:    aws.String(userPoolId),
    LambdaConfig: lambdaConfig,
})
if err != nil {
    log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
}
return err
}
```

- Pour plus de détails sur l'API, reportez-vous [UpdateUserPool](#) à la section Référence des AWS SDK for Go API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **VerifySoftwareToken** avec un AWS SDK ou une CLI


Les exemples de code suivants montrent comment utiliser `VerifySoftwareToken`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- [Inscription d'un utilisateur auprès d'un groupe d'utilisateurs nécessitant l'authentification MFA](#)

.NET

AWS SDK for .NET

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/// <summary>
/// Verify the TOTP and register for MFA.
/// </summary>
/// <param name="session">The name of the session.</param>
/// <param name="code">The MFA code.</param>
/// <returns>The status of the software token.</returns>
public async Task<VerifySoftwareTokenResponseType>
VerifySoftwareTokenAsync(string session, string code)
{
    var tokenRequest = new VerifySoftwareTokenRequest
    {
        UserCode = code,
        Session = session,
    };

    var verifyResponse = await
_cognitoService.VerifySoftwareTokenAsync(tokenRequest);

    return verifyResponse.Status;
}
```

- Pour plus de détails sur l'API, reportez-vous [VerifySoftwareToken](#) à la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenRequest request;
request.SetUserCode(userCode);
request.SetSession(session);

Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenOutcome outcome =
    client.VerifySoftwareToken(request);

if (outcome.IsSuccess()) {
    std::cout << "Verification of the code was successful."
              << std::endl;
    session = outcome.GetResult().GetSession();
}
else {
    std::cerr << "Error with
CognitoIdentityProvider::VerifySoftwareToken. "
              << outcome.GetError().GetMessage()
              << std::endl;
    return false;
}
```

- Pour plus de détails sur l'API, reportez-vous [VerifySoftwareToken](#) à la section Référence des AWS SDK for C++ API.

Java

SDK pour Java 2.x

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
// Verify the TOTP and register for MFA.
public static void verifyTOTP(CognitoIdentityProviderClient
identityProviderClient, String session, String code) {
    try {
        VerifySoftwareTokenRequest tokenRequest =
VerifySoftwareTokenRequest.builder()
            .userCode(code)
            .session(session)
            .build();

        VerifySoftwareTokenResponse verifyResponse =
identityProviderClient.verifySoftwareToken(tokenRequest);
        System.out.println("The status of the token is " +
verifyResponse.statusAsString());

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Pour plus de détails sur l'API, reportez-vous [VerifySoftwareToken](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
const verifySoftwareToken = (totp) => {
  const client = new CognitoIdentityProviderClient({});

  // The 'Session' is provided in the response to 'AssociateSoftwareToken'.
  const session = process.env.SESSION;

  if (!session) {
    throw new Error(
      "Missing a valid Session. Did you run 'admin-initiate-auth'?",
    );
  }

  const command = new VerifySoftwareTokenCommand({
    Session: session,
    UserCode: totp,
  });

  return client.send(command);
};
```

- Pour plus de détails sur l'API, reportez-vous [VerifySoftwareToken](#) à la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
// Verify the TOTP and register for MFA.
suspend fun verifyTOTP(sessionVal: String?, codeVal: String?) {
    val tokenRequest = VerifySoftwareTokenRequest {
        userCode = codeVal
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    val verifyResponse =
identityProviderClient.verifySoftwareToken(tokenRequest)
    println("The status of the token is ${verifyResponse.status}")
}
}
```

- Pour plus de détails sur l'API, consultez [VerifySoftwareToken](#) la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class CognitoIdentityProviderWrapper:
```

```
"""Encapsulates Amazon Cognito actions"""

def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
    """
    :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
    :param user_pool_id: The ID of an existing Amazon Cognito user pool.
    :param client_id: The ID of a client application registered with the user
pool.
    :param client_secret: The client secret, if the client has a secret.
    """
    self.cognito_idp_client = cognito_idp_client
    self.user_pool_id = user_pool_id
    self.client_id = client_id
    self.client_secret = client_secret

def verify_mfa(self, session, user_code):
    """
    Verify a new MFA application that is associated with a user.

    :param session: Session information returned from a previous call to
initiate
                    authentication.
    :param user_code: A code generated by the associated MFA application.
    :return: Status that indicates whether the MFA application is verified.
    """
    try:
        response = self.cognito_idp_client.verify_software_token(
            Session=session, UserCode=user_code
        )
    except ClientError as err:
        logger.error(
            "Couldn't verify MFA. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        response.pop("ResponseMetadata", None)
        return response
```

- Pour plus de détails sur l'API, consultez [VerifySoftwareToken](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Scénarios pour le fournisseur d'identité Amazon Cognito utilisant des kits de développement logiciel AWS

Les exemples de code suivants vous montrent comment implémenter des scénarios courants dans le fournisseur d'identité Amazon Cognito à l'aide AWS de kits SDK. Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions dans le fournisseur d'identité Amazon Cognito. Chaque scénario inclut un lien vers GitHub, où vous pouvez trouver des instructions sur la façon de configurer et d'exécuter le code.

Exemples

- [Confirmez automatiquement les utilisateurs Amazon Cognito connus à l'aide d'une fonction Lambda à l'aide d'un SDK AWS](#)
- [Migrez automatiquement les utilisateurs connus d'Amazon Cognito à l'aide d'une fonction Lambda à l'aide d'un SDK AWS](#)
- [Inscrire un utilisateur auprès d'un groupe d'utilisateurs Amazon Cognito qui nécessite l'authentification multifacteur à l'aide d'un SDK AWS](#)
- [Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito à l'aide d'un SDK AWS](#)

Confirmez automatiquement les utilisateurs Amazon Cognito connus à l'aide d'une fonction Lambda à l'aide d'un SDK AWS


L'exemple de code suivant montre comment confirmer automatiquement les utilisateurs connus d'Amazon Cognito à l'aide d'une fonction Lambda.

- Configurez un groupe d'utilisateurs pour appeler une fonction Lambda pour le PreSignUp déclencheur.

- Inscrivez un utilisateur avec Amazon Cognito.
- La fonction Lambda analyse une table DynamoDB et confirme automatiquement les utilisateurs connus.
- Connectez-vous en tant que nouvel utilisateur, puis nettoyez les ressources.

Go

Kit SDK for Go V2

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Exécutez un scénario interactif à une invite de commande.

```
// AutoConfirm separates the steps of this scenario into individual functions so
// that
// they are simpler to read and understand.
type AutoConfirm struct {
    helper      IScenarioHelper
    questioner  demotools.IQuestioner
    resources   Resources
    cognitoActor *actions.CognitoActions
}

// NewAutoConfirm constructs a new auto confirm runner.
func NewAutoConfirm(sdkConfig aws.Config, questioner demotools.IQuestioner,
    helper IScenarioHelper) AutoConfirm {
    scenario := AutoConfirm{
        helper:      helper,
        questioner:  questioner,
        resources:   Resources{},
        cognitoActor: &actions.CognitoActions{CognitoClient:
        cognitoidentityprovider.NewFromConfig(sdkConfig)},
    }
    scenario.resources.init(scenario.cognitoActor, questioner)
    return scenario
}
```



```
// AddPreSignUpTrigger adds a Lambda handler as an invocation target for the
PreSignUp trigger.
func (runner *AutoConfirm) AddPreSignUpTrigger(userPoolId string, functionArn
string) {
    log.Printf("Let's add a Lambda function to handle the PreSignUp trigger from
Cognito.\n" +
        "This trigger happens when a user signs up, and lets your function take action
before the main Cognito\n" +
        "sign up processing occurs.\n")
    err := runner.cognitoActor.UpdateTriggers(
        userPoolId,
        actions.TriggerInfo{Trigger: actions.PreSignUp, HandlerArn:
aws.String(functionArn)})
    if err != nil {
        panic(err)
    }
    log.Printf("Lambda function %v added to user pool %v to handle the PreSignUp
trigger.\n",
        functionArn, userPoolId)
}

// SignUpUser signs up a user from the known user table with a password you
specify.
func (runner *AutoConfirm) SignUpUser(clientId string, usersTable string)
(string, string) {
    log.Println("Let's sign up a user to your Cognito user pool. When the user's
email matches an email in the\n" +
        "DynamoDB known users table, it is automatically verified and the user is
confirmed.")

    knownUsers, err := runner.helper.GetKnownUsers(usersTable)
    if err != nil {
        panic(err)
    }
    userChoice := runner.questioner.AskChoice("Which user do you want to use?\n",
knownUsers.UserNameList())
    user := knownUsers.Users[userChoice]

    var signedUp bool
    var userConfirmed bool
    password := runner.questioner.AskPassword("Enter a password that has at least
eight characters, uppercase, lowercase, numbers and symbols.\n"+
        "(the password will not display as you type):", 8)
```

```
for !signedUp {
    log.Printf("Signing up user '%v' with email '%v' to Cognito.\n", user.UserName,
user.UserEmail)
    userConfirmed, err = runner.cognitoActor.SignUp(clientId, user.UserName,
password, user.UserEmail)
    if err != nil {
        var invalidPassword *types.InvalidPasswordException
        if errors.As(err, &invalidPassword) {
            password = runner.questioner.AskPassword("Enter another password:", 8)
        } else {
            panic(err)
        }
    } else {
        signedUp = true
    }
}
log.Printf("User %v signed up, confirmed = %v.\n", user.UserName, userConfirmed)

log.Println(strings.Repeat("-", 88))

return user.UserName, password
}

// SignInUser signs in a user.
func (runner *AutoConfirm) SignInUser(clientId string, userName string, password
string) string {
    runner.questioner.Ask("Press Enter when you're ready to continue.")
    log.Printf("Let's sign in as %v...\n", userName)
    authResult, err := runner.cognitoActor.SignIn(clientId, userName, password)
    if err != nil {
        panic(err)
    }
    log.Printf("Successfully signed in. Your access token starts with: %v...\n",
(*authResult.AccessToken)[:10])
    log.Println(strings.Repeat("-", 88))
    return *authResult.AccessToken
}

// Run runs the scenario.
func (runner *AutoConfirm) Run(stackName string) {
    defer func() {
        if r := recover(); r != nil {
            log.Println("Something went wrong with the demo.")
            runner.resources.Cleanup()
        }
    }()
}
```

```

    }
  }()

  log.Println(strings.Repeat("-", 88))
  log.Printf("Welcome\n")

  log.Println(strings.Repeat("-", 88))

  stackOutputs, err := runner.helper.GetStackOutputs(stackName)
  if err != nil {
    panic(err)
  }
  runner.resources.userPoolId = stackOutputs["UserPoolId"]
  runner.helper.PopulateUserTable(stackOutputs["TableName"])

  runner.AddPreSignUpTrigger(stackOutputs["UserPoolId"],
    stackOutputs["AutoConfirmFunctionArn"])
  runner.resources.triggers = append(runner.resources.triggers, actions.PreSignUp)
  userName, password := runner.SignUpUser(stackOutputs["UserPoolClientId"],
    stackOutputs["TableName"])
  runner.helper.ListRecentLogEvents(stackOutputs["AutoConfirmFunction"])
  runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
    runner.SignInUser(stackOutputs["UserPoolClientId"], userName, password))

  runner.resources.Cleanup()

  log.Println(strings.Repeat("-", 88))
  log.Println("Thanks for watching!")
  log.Println(strings.Repeat("-", 88))
}

```

Manipulez le PreSignUp déclencheur à l'aide d'une fonction Lambda.

```

const TABLE_NAME = "TABLE_NAME"

// UserInfo defines structured user data that can be marshalled to a DynamoDB
// format.
type UserInfo struct {
  UserName string `dynamodbav:"UserName"`
  UserEmail string `dynamodbav:"UserEmail"`
}

```

```
}

// GetKey marshals the user email value to a DynamoDB key format.
func (user UserInfo) GetKey() map[string]dynamodbtypes.AttributeValue {
    userEmail, err := attributevalue.Marshal(user.UserEmail)
    if err != nil {
        panic(err)
    }
    return map[string]dynamodbtypes.AttributeValue{"UserEmail": userEmail}
}

type handler struct {
    dynamoClient *dynamodb.Client
}

// HandleRequest handles the PreSignUp event by looking up a user in an Amazon
// DynamoDB table and
// specifying whether they should be confirmed and verified.
func (h *handler) HandleRequest(ctx context.Context, event
events.CognitoEventUserPoolsPreSignup) (events.CognitoEventUserPoolsPreSignup,
error) {
    log.Printf("Received presignup from %v for user '%v'", event.TriggerSource,
event.UserName)
    if event.TriggerSource != "PreSignUp_SignUp" {
        // Other trigger sources, such as PreSignUp_AdminInitiateAuth, ignore the
        // response from this handler.
        return event, nil
    }
    tableName := os.Getenv(TABLE_NAME)
    user := UserInfo{
        UserEmail: event.Request.UserAttributes["email"],
    }
    log.Printf("Looking up email %v in table %v.\n", user.UserEmail, tableName)
    output, err := h.dynamoClient.GetItem(ctx, &dynamodb.GetItemInput{
        Key:      user.GetKey(),
        TableName: aws.String(tableName),
    })
    if err != nil {
        log.Printf("Error looking up email %v.\n", user.UserEmail)
        return event, err
    }
    if output.Item == nil {
        log.Printf("Email %v not found. Email verification is required.\n",
user.UserEmail)
    }
}
```

```
    return event, err
}

err = attributevalue.UnmarshalMap(output.Item, &user)
if err != nil {
    log.Printf("Couldn't unmarshal DynamoDB item. Here's why: %v\n", err)
    return event, err
}

if user.UserName != event.UserName {
    log.Printf("UserEmail %v found, but stored UserName '%v' does not match
supplied UserName '%v'. Verification is required.\n",
    user.UserEmail, user.UserName, event.UserName)
} else {
    log.Printf("UserEmail %v found with matching UserName %v. User is confirmed.
\n", user.UserEmail, user.UserName)
    event.Response.AutoConfirmUser = true
    event.Response.AutoVerifyEmail = true
}

return event, err
}

func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        log.Panicln(err)
    }
    h := handler{
        dynamoClient: dynamodb.NewFromConfig(sdkConfig),
    }
    lambda.Start(h.HandleRequest)
}
```

Créez une structure qui exécute des tâches courantes.

```
// IScenarioHelper defines common functions used by the workflows in this
example.
type IScenarioHelper interface {
    Pause(secs int)
```

```
GetStackOutputs(stackName string) (actions.StackOutputs, error)
PopulateUserTable(tableName string)
GetKnownUsers(tableName string) (actions.UserList, error)
AddKnownUser(tableName string, user actions.User)
ListRecentLogEvents(functionName string)
}

// ScenarioHelper contains AWS wrapper structs used by the workflows in this
// example.
type ScenarioHelper struct {
    questioner demotools.IQuestioner
    dynamoActor *actions.DynamoActions
    cfnActor *actions.CloudFormationActions
    cwActor *actions.CloudWatchLogsActions
    isTestRun bool
}

// NewScenarioHelper constructs a new scenario helper.
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
ScenarioHelper {
    scenario := ScenarioHelper{
        questioner: questioner,
        dynamoActor: &actions.DynamoActions{DynamoClient:
        dynamodb.NewFromConfig(sdkConfig)},
        cfnActor: &actions.CloudFormationActions{CfnClient:
        cloudformation.NewFromConfig(sdkConfig)},
        cwActor: &actions.CloudWatchLogsActions{CwlClient:
        cloudwatchlogs.NewFromConfig(sdkConfig)},
    }
    return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
    if !helper.isTestRun {
        time.Sleep(time.Duration(secs) * time.Second)
    }
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
// structured format.
func (helper ScenarioHelper) GetStackOutputs(stackName string)
(actions.StackOutputs, error) {
    return helper.cfnActor.GetOutputs(stackName), nil
}
```

```
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(tableName string) {
    log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
this example.\n", tableName)
    err := helper.dynamoActor.PopulateTable(tableName)
    if err != nil {
        panic(err)
    }
}

// GetKnownUsers gets the users from the known users table in a structured
format.
func (helper ScenarioHelper) GetKnownUsers(tableName string) (actions.UserList,
error) {
    knownUsers, err := helper.dynamoActor.Scan(tableName)
    if err != nil {
        log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
tableName, err)
    }
    return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(tableName string, user actions.User) {
    log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
table...\n",
    user.UserName, user.UserEmail)
    err := helper.dynamoActor.AddUser(tableName, user)
    if err != nil {
        panic(err)
    }
}

// ListRecentLogEvents gets the most recent log stream and events for the
specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(functionName string) {
    log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
    helper.Pause(10)
    log.Println("Okay, let's check the logs to find what's happened recently with
your Lambda function.")
    logStream, err := helper.cwlActor.GetLatestLogStream(functionName)
    if err != nil {
```

```
panic(err)
}
log.Printf("Getting some recent events from log stream %v\n",
*logStream.LogStreamName)
events, err := helper.cwlActor.GetLogEvents(functionName,
*logStream.LogStreamName, 10)
if err != nil {
panic(err)
}
for _, event := range events {
log.Printf("\t%v", *event.Message)
}
log.Println(strings.Repeat("-", 88))
}
```

Créez une structure qui englobe les actions Amazon Cognito.

```
type CognitoActions struct {
CognitoClient *cognitoidentityprovider.Client
}

// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
trigger.
type Trigger int

const (
PreSignUp Trigger = iota
UserMigration
PostAuthentication
)

type TriggerInfo struct {
Trigger Trigger
HandlerArn *string
}
```



```
// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
// is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(userPoolId string,
triggers ...TriggerInfo) error {
    output, err := actor.CognitoClient.DescribeUserPool(context.TODO(),
&cognitoidentityprovider.DescribeUserPoolInput{
        UserPoolId: aws.String(userPoolId),
    })
    if err != nil {
        log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
userPoolId, err)
        return err
    }
    lambdaConfig := output.UserPool.LambdaConfig
    for _, trigger := range triggers {
        switch trigger.Trigger {
        case PreSignUp:
            lambdaConfig.PreSignUp = trigger.HandlerArn
        case UserMigration:
            lambdaConfig.UserMigration = trigger.HandlerArn
        case PostAuthentication:
            lambdaConfig.PostAuthentication = trigger.HandlerArn
        }
    }
    _, err = actor.CognitoClient.UpdateUserPool(context.TODO(),
&cognitoidentityprovider.UpdateUserPoolInput{
        UserPoolId: aws.String(userPoolId),
        LambdaConfig: lambdaConfig,
    })
    if err != nil {
        log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
    }
    return err
}

// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(clientId string, userName string, password
string, userEmail string) (bool, error) {
    confirmed := false
    output, err := actor.CognitoClient.SignUp(context.TODO(),
&cognitoidentityprovider.SignUpInput{
```

```
    ClientId: aws.String(clientId),
    Password: aws.String(password),
    Username: aws.String(userName),
    UserAttributes: []types.AttributeType{
        {Name: aws.String("email"), Value: aws.String(userEmail)},
    },
})
if err != nil {
    var invalidPassword *types.InvalidPasswordException
    if errors.As(err, &invalidPassword) {
        log.Println(*invalidPassword.Message)
    } else {
        log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
    }
} else {
    confirmed = output.UserConfirmed
}
return confirmed, err
}

// SignIn signs in a user to Amazon Cognito using a username and password
authentication flow.
func (actor CognitoActions) SignIn(clientId string, userName string, password
string) (*types.AuthenticationResultType, error) {
    var authResult *types.AuthenticationResultType
    output, err := actor.CognitoClient.InitiateAuth(context.TODO(),
&cognitoidentityprovider.InitiateAuthInput{
        AuthFlow:      "USER_PASSWORD_AUTH",
        ClientId:      aws.String(clientId),
        AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
    })
    if err != nil {
        var resetRequired *types.PasswordResetRequiredException
        if errors.As(err, &resetRequired) {
            log.Println(*resetRequired.Message)
        } else {
            log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
        }
    } else {
        authResult = output.AuthenticationResult
    }
    return authResult, err
}
```

```
}

// ForgotPassword starts a password recovery flow for a user. This flow typically
// sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(clientId string, userName string)
(*types.CodeDeliveryDetailsType, error) {
    output, err := actor.CognitoClient.ForgotPassword(context.TODO(),
&cognitoidentityprovider.ForgotPasswordInput{
    ClientId: aws.String(clientId),
    Username: aws.String(userName),
})
    if err != nil {
        log.Printf("Couldn't start password reset for user '%v'. Here's why: %v\n",
userName, err)
    }
    return output.CodeDeliveryDetails, err
}

// ConfirmForgotPassword confirms a user with a confirmation code and a new
// password.
func (actor CognitoActions) ConfirmForgotPassword(clientId string, code string,
userName string, password string) error {
    _, err := actor.CognitoClient.ConfirmForgotPassword(context.TODO(),
&cognitoidentityprovider.ConfirmForgotPasswordInput{
    ClientId:      aws.String(clientId),
    ConfirmationCode: aws.String(code),
    Password:      aws.String(password),
    Username:      aws.String(userName),
})
    if err != nil {
        var invalidPassword *types.InvalidPasswordException
        if errors.As(err, &invalidPassword) {
            log.Println(*invalidPassword.Message)
        } else {
            log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
        }
    }
    return err
}
```

```
// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(userAccessToken string) error {
    _, err := actor.CognitoClient.DeleteUser(context.TODO(),
        &cognitoidentityprovider.DeleteUserInput{
            AccessToken: aws.String(userAccessToken),
        })
    if err != nil {
        log.Printf("Couldn't delete user. Here's why: %v\n", err)
    }
    return err
}

// AdminCreateUser uses administrator credentials to add a user to a user pool.
// This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(userPoolId string, userName string,
    userEmail string) error {
    _, err := actor.CognitoClient.AdminCreateUser(context.TODO(),
        &cognitoidentityprovider.AdminCreateUserInput{
            UserPoolId:      aws.String(userPoolId),
            Username:      aws.String(userName),
            MessageAction: types.MessageActionTypeSuppress,
            UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
                aws.String(userEmail)}}},
        })
    if err != nil {
        var userExists *types.UsernameExistsException
        if errors.As(err, &userExists) {
            log.Printf("User %v already exists in the user pool.", userName)
            err = nil
        } else {
            log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
        }
    }
    return err
}
```

```
// AdminSetUserPassword uses administrator credentials to set a password for a
// user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(userPoolId string, userName
string, password string) error {
_, err := actor.CognitoClient.AdminSetUserPassword(context.TODO(),
&cognitoidentityprovider.AdminSetUserPasswordInput{
Password:  aws.String(password),
UserPoolId: aws.String(userPoolId),
Username:  aws.String(userName),
Permanent: true,
})
if err != nil {
var invalidPassword *types.InvalidPasswordException
if errors.As(err, &invalidPassword) {
log.Println(*invalidPassword.Message)
} else {
log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
err)
}
}
return err
}
```

Créez une structure qui enveloppe les actions DynamoDB.

```
// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)
// actions
// used in the examples.
type DynamoActions struct {
DynamoClient *dynamodb.Client
}

// User defines structured user data.
type User struct {
UserName string
UserEmail string
LastLogin *LoginInfo `dynamodbav:",omitempty"`
}
```

```
// LoginInfo defines structured custom login data.
type LoginInfo struct {
    UserPoolId string
    ClientId   string
    Time      string
}

// UserList defines a list of users.
type UserList struct {
    Users []User
}

// UserNameList returns the usernames contained in a UserList as a list of
strings.
func (users *UserList) UserNameList() []string {
    names := make([]string, len(users.Users))
    for i := 0; i < len(users.Users); i++ {
        names[i] = users.Users[i].UserName
    }
    return names
}

// PopulateTable adds a set of test users to the table.
func (actor DynamoActions) PopulateTable(tableName string) error {
    var err error
    var item map[string]types.AttributeValue
    var writeReqs []types.WriteRequest
    for i := 1; i < 4; i++ {
        item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
%v", i), UserEmail: fmt.Sprintf("test_email_%v@example.com", i)})
        if err != nil {
            log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
err)
            return err
        }
        writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
&types.PutRequest{Item: item}})
    }
    _, err = actor.DynamoClient.BatchWriteItem(context.TODO(),
&dynamodb.BatchWriteItemInput{
    RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
})
    if err != nil {
```

```
    log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
        tableName, err)
    }
    return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(tableName string) (UserList, error) {
    var userList UserList
    output, err := actor.DynamoClient.Scan(context.TODO(), &dynamodb.ScanInput{
        TableName: aws.String(tableName),
    })
    if err != nil {
        log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
            err)
    } else {
        err = attributevalue.UnmarshalListOfMaps(output.Items, &userList.Users)
        if err != nil {
            log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
        }
    }
    return userList, err
}

// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(tableName string, user User) error {
    userItem, err := attributevalue.MarshalMap(user)
    if err != nil {
        log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
    }
    _, err = actor.DynamoClient.PutItem(context.TODO(), &dynamodb.PutItemInput{
        Item:      userItem,
        TableName: aws.String(tableName),
    })
    if err != nil {
        log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
    }
    return err
}
```

Créez une structure qui englobe les actions CloudWatch Logs.

```
type CloudWatchLogsActions struct {
    CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(functionName string)
(types.LogStream, error) {
    var logStream types.LogStream
    logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
    output, err := actor.CwlClient.DescribeLogStreams(context.TODO(),
&cloudwatchlogs.DescribeLogStreamsInput{
    Descending:    aws.Bool(true),
    Limit:         aws.Int32(1),
    LogGroupName: aws.String(logGroupName),
    OrderBy:      types.OrderByLastEventTime,
})
    if err != nil {
        log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
logGroupName, err)
    } else {
        logStream = output.LogStreams[0]
    }
    return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
stream.
func (actor CloudWatchLogsActions) GetLogEvents(functionName string,
logStreamName string, eventCount int32) (
[]types.OutputLogEvent, error) {
    var events []types.OutputLogEvent
    logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
    output, err := actor.CwlClient.GetLogEvents(context.TODO(),
&cloudwatchlogs.GetLogEventsInput{
    LogStreamName: aws.String(logStreamName),
    Limit:         aws.Int32(eventCount),
    LogGroupName:  aws.String(logGroupName),
})
    if err != nil {
        log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
logStreamName, err)
    } else {
```



```

    events = output.Events
  }
  return events, err
}

```

Créez une structure qui englobe les actions. AWS CloudFormation

```

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
  CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
// structured format.
func (actor CloudFormationActions) GetOutputs(stackName string) StackOutputs {
  output, err := actor.CfnClient.DescribeStacks(context.TODO(),
    &cloudformation.DescribeStacksInput{
      StackName: aws.String(stackName),
    })
  if err != nil || len(output.Stacks) == 0 {
    log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
      stackName, err)
  }
  stackOutputs := StackOutputs{}
  for _, out := range output.Stacks[0].Outputs {
    stackOutputs[*out.OutputKey] = *out.OutputValue
  }
  return stackOutputs
}

```

Nettoyez les ressources.

```

// Resources keeps track of AWS resources created during an example and handles
// cleanup when the example finishes.
type Resources struct {

```

```

userPoolId      string
userAccessTokens []string
triggers        []actions.Trigger

cognitoActor *actions.CognitoActions
questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
demotools.IQuestioner) {
resources.userAccessTokens = []string{}
resources.triggers = []actions.Trigger{}
resources.cognitoActor = cognitoActor
resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
func (resources *Resources) Cleanup() {
defer func() {
if r := recover(); r != nil {
log.Printf("Something went wrong during cleanup.\n%v\n", r)
log.Println("Use the AWS Management Console to remove any remaining resources
\n" +
"that were created for this scenario.")
}
}()

wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
resources that were created "+
"during this demo (y/n)?", "y")
if wantDelete {
for _, accessToken := range resources.userAccessTokens {
err := resources.cognitoActor.DeleteUser(accessToken)
if err != nil {
log.Println("Couldn't delete user during cleanup.")
panic(err)
}
log.Println("Deleted user.")
}
triggerList := make([]actions.TriggerInfo, len(resources.triggers))
for i := 0; i < len(resources.triggers); i++ {
triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
HandlerArn: nil}
}
}

```

```
err := resources.cognitoActor.UpdateTriggers(resources.userPoolId,
triggerList...)
if err != nil {
    log.Println("Couldn't update Cognito triggers during cleanup.")
    panic(err)
}
log.Println("Removed Cognito triggers from user pool.")
} else {
    log.Println("Be sure to remove resources when you're done with them to avoid
unexpected charges!")
}
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Go .
 - [DeleteUser](#)
 - [InitiateAuth](#)
 - [SignUp](#)
 - [UpdateUserPool](#)

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Migrez automatiquement les utilisateurs connus d'Amazon Cognito à l'aide d'une fonction Lambda à l'aide d'un SDK AWS

L'exemple de code suivant montre comment migrer automatiquement des utilisateurs Amazon Cognito connus à l'aide d'une fonction Lambda.

- Configurez un groupe d'utilisateurs pour appeler une fonction Lambda pour le `MigrateUser` déclencheur.
- Connectez-vous à Amazon Cognito avec un nom d'utilisateur et une adresse e-mail qui ne figurent pas dans le groupe d'utilisateurs.
- La fonction Lambda analyse une table DynamoDB et migre automatiquement les utilisateurs connus vers le groupe d'utilisateurs.

- Exécutez le flux de mots de passe oubliés pour réinitialiser le mot de passe de l'utilisateur migré.
- Connectez-vous en tant que nouvel utilisateur, puis nettoyez les ressources.

Go

Kit SDK for Go V2

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Exécutez un scénario interactif à une invite de commande.

```
import (  
    "errors"  
    "fmt"  
    "log"  
    "strings"  
    "user_pools_and_lambda_triggers/actions"  
  
    "github.com/aws/aws-sdk-go-v2/aws"  
    "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"  
    "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"  
    "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"  
)  
  
// MigrateUser separates the steps of this scenario into individual functions so  
// that  
// they are simpler to read and understand.  
type MigrateUser struct {  
    helper          IScenarioHelper  
    questioner     demotools.IQuestioner  
    resources      Resources  
    cognitoActor   *actions.CognitoActions  
}  
  
// NewMigrateUser constructs a new migrate user runner.
```

```

func NewMigrateUser(sdkConfig aws.Config, questioner demotools.IQuestioner,
helper IScenarioHelper) MigrateUser {
scenario := MigrateUser{
helper:      helper,
questioner:  questioner,
resources:   Resources{},
cognitoActor: &actions.CognitoActions{CognitoClient:
cognitoidentityprovider.NewFromConfig(sdkConfig)},
}
scenario.resources.init(scenario.cognitoActor, questioner)
return scenario
}

// AddMigrateUserTrigger adds a Lambda handler as an invocation target for the
MigrateUser trigger.
func (runner *MigrateUser) AddMigrateUserTrigger(userPoolId string, functionArn
string) {
log.Printf("Let's add a Lambda function to handle the MigrateUser trigger from
Cognito.\n" +
"This trigger happens when an unknown user signs in, and lets your function
take action before Cognito\n" +
"rejects the user.\n\n")
err := runner.cognitoActor.UpdateTriggers(
userPoolId,
actions.TriggerInfo{Trigger: actions.UserMigration, HandlerArn:
aws.String(functionArn)})
if err != nil {
panic(err)
}
log.Printf("Lambda function %v added to user pool %v to handle the MigrateUser
trigger.\n",
functionArn, userPoolId)

log.Println(strings.Repeat("-", 88))
}

// SignInUser adds a new user to the known users table and signs that user in to
Amazon Cognito.
func (runner *MigrateUser) SignInUser(usersTable string, clientId string) (bool,
actions.User) {
log.Println("Let's sign in a user to your Cognito user pool. When the username
and email matches an entry in the\n" +
"DynamoDB known users table, the email is automatically verified and the user
is migrated to the Cognito user pool.")

```

```
user := actions.User{}
user.UserName = runner.questioner.Ask("\nEnter a username:")
user.UserEmail = runner.questioner.Ask("\nEnter an email that you own. This
email will be used to confirm user migration\n" +
"during this example:")

runner.helper.AddKnownUser(usersTable, user)

var err error
var resetRequired *types.PasswordResetRequiredException
var authResult *types.AuthenticationResultType
signedIn := false
for !signedIn && resetRequired == nil {
    log.Printf("Signing in to Cognito as user '%v'. The expected result is a
PasswordResetRequiredException.\n\n", user.UserName)
    authResult, err = runner.cognitoActor.SignIn(clientId, user.UserName, "_")
    if err != nil {
        if errors.As(err, &resetRequired) {
            log.Printf("\nUser '%v' is not in the Cognito user pool but was found in the
DynamoDB known users table.\n"+
"User migration is started and a password reset is required.",
user.UserName)
        } else {
            panic(err)
        }
    } else {
        log.Printf("User '%v' successfully signed in. This is unexpected and probably
means you have not\n"+
"cleaned up a previous run of this scenario, so the user exist in the Cognito
user pool.\n"+
"You can continue this example and select to clean up resources, or manually
remove\n"+
"the user from your user pool and try again.", user.UserName)
        runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
*authResult.AccessToken)
        signedIn = true
    }
}

log.Println(strings.Repeat("-", 88))
return resetRequired != nil, user
}
```

```
// ResetPassword starts a password recovery flow.
func (runner *MigrateUser) ResetPassword(clientId string, user actions.User) {
    wantCode := runner.questioner.AskBool(fmt.Sprintf("In order to migrate the user
    to Cognito, you must be able to receive a confirmation\n"+
    "code by email at %v. Do you want to send a code (y/n)?", user.UserEmail), "y")
    if !wantCode {
        log.Println("To complete this example and successfully migrate a user to
        Cognito, you must enter an email\n" +
        "you own that can receive a confirmation code.")
        return
    }
    codeDelivery, err := runner.cognitoActor.ForgotPassword(clientId, user.UserName)
    if err != nil {
        panic(err)
    }
    log.Printf("\nA confirmation code has been sent to %v.",
    *codeDelivery.Destination)
    code := runner.questioner.Ask("Check your email and enter it here:")

    confirmed := false
    password := runner.questioner.AskPassword("\nEnter a password that has at least
    eight characters, uppercase, lowercase, numbers and symbols.\n"+
    "(the password will not display as you type):", 8)
    for !confirmed {
        log.Printf("\nConfirming password reset for user '%v'.\n", user.UserName)
        err = runner.cognitoActor.ConfirmForgotPassword(clientId, code, user.UserName,
        password)
        if err != nil {
            var invalidPassword *types.InvalidPasswordException
            if errors.As(err, &invalidPassword) {
                password = runner.questioner.AskPassword("\nEnter another password:", 8)
            } else {
                panic(err)
            }
        } else {
            confirmed = true
        }
    }
    log.Printf("User '%v' successfully confirmed and migrated.\n", user.UserName)
    log.Println("Signing in with your username and password...")
    authResult, err := runner.cognitoActor.SignIn(clientId, user.UserName, password)
    if err != nil {
        panic(err)
    }
}
```

```
log.Printf("Successfully signed in. Your access token starts with: %v...\n",
(*authResult.AccessToken)[:10])
runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
*authResult.AccessToken)

log.Println(strings.Repeat("-", 88))
}

// Run runs the scenario.
func (runner *MigrateUser) Run(stackName string) {
defer func() {
if r := recover(); r != nil {
log.Println("Something went wrong with the demo.")
runner.resources.Cleanup()
}
}()

log.Println(strings.Repeat("-", 88))
log.Printf("Welcome\n")

log.Println(strings.Repeat("-", 88))

stackOutputs, err := runner.helper.GetStackOutputs(stackName)
if err != nil {
panic(err)
}
runner.resources.userPoolId = stackOutputs["UserPoolId"]

runner.AddMigrateUserTrigger(stackOutputs["UserPoolId"],
stackOutputs["MigrateUserFunctionArn"])
runner.resources.triggers = append(runner.resources.triggers,
actions.UserMigration)
resetNeeded, user := runner.SignInUser(stackOutputs["TableName"],
stackOutputs["UserPoolClientId"])
if resetNeeded {
runner.helper.ListRecentLogEvents(stackOutputs["MigrateUserFunction"])
runner.ResetPassword(stackOutputs["UserPoolClientId"], user)
}

runner.resources.Cleanup()

log.Println(strings.Repeat("-", 88))
log.Println("Thanks for watching!")
log.Println(strings.Repeat("-", 88))
```



```
}
```

Manipulez le `MigrateUser` déclencheur à l'aide d'une fonction Lambda.

```
const TABLE_NAME = "TABLE_NAME"

// UserInfo defines structured user data that can be marshalled to a DynamoDB
// format.
type UserInfo struct {
    UserName string `dynamodbav:"UserName"`
    UserEmail string `dynamodbav:"UserEmail"`
}

type handler struct {
    dynamoClient *dynamodb.Client
}

// HandleRequest handles the MigrateUser event by looking up a user in an Amazon
// DynamoDB table and
// specifying whether they should be migrated to the user pool.
func (h *handler) HandleRequest(ctx context.Context, event
events.CognitoEventUserPoolsMigrateUser)
(events.CognitoEventUserPoolsMigrateUser, error) {
    log.Printf("Received migrate trigger from %v for user '%v'",
event.TriggerSource, event.UserName)
    if event.TriggerSource != "UserMigration_Authentication" {
        return event, nil
    }
    tableName := os.Getenv(TABLE_NAME)
    user := UserInfo{
        UserName: event.UserName,
    }
    log.Printf("Looking up user '%v' in table %v.\n", user.UserName, tableName)
    filterEx := expression.Name("UserName").Equal(expression.Value(user.UserName))
    expr, err := expression.NewBuilder().WithFilter(filterEx).Build()
    if err != nil {
        log.Printf("Error building expression to query for user '%v'.\n",
user.UserName)
        return event, err
    }
}
```

```
output, err := h.dynamoClient.Scan(ctx, &dynamodb.ScanInput{
    TableName:          aws.String(tableName),
    FilterExpression:   expr.Filter(),
    ExpressionAttributeNames: expr.Names(),
    ExpressionAttributeValues: expr.Values(),
})
if err != nil {
    log.Printf("Error looking up user '%v'.\n", user.UserName)
    return event, err
}
if output.Items == nil || len(output.Items) == 0 {
    log.Printf("User '%v' not found, not migrating user.\n", user.UserName)
    return event, err
}

var users []UserInfo
err = attributevalue.UnmarshalListOfMaps(output.Items, &users)
if err != nil {
    log.Printf("Couldn't unmarshal DynamoDB items. Here's why: %v\n", err)
    return event, err
}

user = users[0]
log.Printf("UserName '%v' found with email %v. User is migrated and must reset
password.\n", user.UserName, user.UserEmail)
event.CognitoEventUserPoolsMigrateUserResponse.UserAttributes =
map[string]string{
    "email":          user.UserEmail,
    "email_verified": "true", // email_verified is required for the forgot password
flow.
}
event.CognitoEventUserPoolsMigrateUserResponse.FinalUserStatus =
"RESET_REQUIRED"
event.CognitoEventUserPoolsMigrateUserResponse.MessageAction = "SUPPRESS"

return event, err
}

func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        log.Panicln(err)
    }
    h := handler{
```

```

    dynamoClient: dynamodb.NewFromConfig(sdkConfig),
  }
  lambda.Start(h.HandleRequest)
}

```

Créez une structure qui exécute des tâches courantes.

```

// IScenarioHelper defines common functions used by the workflows in this
// example.
type IScenarioHelper interface {
  Pause(secs int)
  GetStackOutputs(stackName string) (actions.StackOutputs, error)
  PopulateUserTable(tableName string)
  GetKnownUsers(tableName string) (actions.UserList, error)
  AddKnownUser(tableName string, user actions.User)
  ListRecentLogEvents(functionName string)
}

// ScenarioHelper contains AWS wrapper structs used by the workflows in this
// example.
type ScenarioHelper struct {
  questioner demotools.IQuestioner
  dynamoActor *actions.DynamoActions
  cfnActor *actions.CloudFormationActions
  cwlActor *actions.CloudWatchLogsActions
  isTestRun bool
}

// NewScenarioHelper constructs a new scenario helper.
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
  ScenarioHelper {
  scenario := ScenarioHelper{
    questioner: questioner,
    dynamoActor: &actions.DynamoActions{DynamoClient:
dynamodb.NewFromConfig(sdkConfig)},
    cfnActor: &actions.CloudFormationActions{CfnClient:
cloudformation.NewFromConfig(sdkConfig)},
    cwlActor: &actions.CloudWatchLogsActions{CwlClient:
cloudwatchlogs.NewFromConfig(sdkConfig)},
  }
}

```

```
    return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
    if !helper.isTestRun {
        time.Sleep(time.Duration(secs) * time.Second)
    }
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
// structured format.
func (helper ScenarioHelper) GetStackOutputs(stackName string)
(actions.StackOutputs, error) {
    return helper.cfnActor.GetOutputs(stackName), nil
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(tableName string) {
    log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
this example.\n", tableName)
    err := helper.dynamoActor.PopulateTable(tableName)
    if err != nil {
        panic(err)
    }
}

// GetKnownUsers gets the users from the known users table in a structured
// format.
func (helper ScenarioHelper) GetKnownUsers(tableName string) (actions.UserList,
error) {
    knownUsers, err := helper.dynamoActor.Scan(tableName)
    if err != nil {
        log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
tableName, err)
    }
    return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(tableName string, user actions.User) {
    log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
table...\n",
user.UserName, user.UserEmail)
```

```
err := helper.dynamoActor.AddUser(tableName, user)
if err != nil {
    panic(err)
}

// ListRecentLogEvents gets the most recent log stream and events for the
// specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(functionName string) {
    log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
    helper.Pause(10)
    log.Println("Okay, let's check the logs to find what's happened recently with
    your Lambda function.")
    logStream, err := helper.cwlActor.GetLatestLogStream(functionName)
    if err != nil {
        panic(err)
    }
    log.Printf("Getting some recent events from log stream %v\n",
    *logStream.LogStreamName)
    events, err := helper.cwlActor.GetLogEvents(functionName,
    *logStream.LogStreamName, 10)
    if err != nil {
        panic(err)
    }
    for _, event := range events {
        log.Printf("\t%v", *event.Message)
    }
    log.Println(strings.Repeat("-", 88))
}
```

Créez une structure qui englobe les actions Amazon Cognito.

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}
```

```
// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
trigger.
type Trigger int

const (
    PreSignUp Trigger = iota
    UserMigration
    PostAuthentication
)

type TriggerInfo struct {
    Trigger    Trigger
    HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
// is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(userPoolId string,
    triggers ...TriggerInfo) error {
    output, err := actor.CognitoClient.DescribeUserPool(context.TODO(),
    &cognitoidentityprovider.DescribeUserPoolInput{
        UserPoolId: aws.String(userPoolId),
    })
    if err != nil {
        log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
        userPoolId, err)
        return err
    }
    lambdaConfig := output.UserPool.LambdaConfig
    for _, trigger := range triggers {
        switch trigger.Trigger {
            case PreSignUp:
                lambdaConfig.PreSignUp = trigger.HandlerArn
            case UserMigration:
                lambdaConfig.UserMigration = trigger.HandlerArn
            case PostAuthentication:
                lambdaConfig.PostAuthentication = trigger.HandlerArn
        }
    }
    _, err = actor.CognitoClient.UpdateUserPool(context.TODO(),
    &cognitoidentityprovider.UpdateUserPoolInput{
        UserPoolId:    aws.String(userPoolId),
        LambdaConfig: lambdaConfig,
    })
}
```

```
    })
    if err != nil {
        log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
    }
    return err
}
```

// SignUp signs up a user with Amazon Cognito.

```
func (actor CognitoActions) SignUp(clientId string, userName string, password
string, userEmail string) (bool, error) {
    confirmed := false
    output, err := actor.CognitoClient.SignUp(context.TODO(),
&cognitoidentityprovider.SignUpInput{
    ClientId: aws.String(clientId),
    Password: aws.String(password),
    Username: aws.String(userName),
    UserAttributes: []types.AttributeType{
        {Name: aws.String("email"), Value: aws.String(userEmail)},
    },
})
    if err != nil {
        var invalidPassword *types.InvalidPasswordException
        if errors.As(err, &invalidPassword) {
            log.Println(*invalidPassword.Message)
        } else {
            log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
        }
    } else {
        confirmed = output.UserConfirmed
    }
    return confirmed, err
}
```

// SignIn signs in a user to Amazon Cognito using a username and password authentication flow.

```
func (actor CognitoActions) SignIn(clientId string, userName string, password
string) (*types.AuthenticationResultType, error) {
    var authResult *types.AuthenticationResultType
    output, err := actor.CognitoClient.InitiateAuth(context.TODO(),
&cognitoidentityprovider.InitiateAuthInput{
```

```
AuthFlow:      "USER_PASSWORD_AUTH",
ClientId:      aws.String(clientId),
AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
})
if err != nil {
    var resetRequired *types.PasswordResetRequiredException
    if errors.As(err, &resetRequired) {
        log.Println(*resetRequired.Message)
    } else {
        log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
    }
} else {
    authResult = output.AuthenticationResult
}
return authResult, err
}

// ForgotPassword starts a password recovery flow for a user. This flow typically
// sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(clientId string, userName string)
(*types.CodeDeliveryDetailsType, error) {
    output, err := actor.CognitoClient.ForgotPassword(context.TODO(),
&cognitoidentityprovider.ForgotPasswordInput{
    ClientId: aws.String(clientId),
    Username: aws.String(userName),
})
    if err != nil {
        log.Printf("Couldn't start password reset for user '%v'. Here's why: %v\n",
userName, err)
    }
    return output.CodeDeliveryDetails, err
}

// ConfirmForgotPassword confirms a user with a confirmation code and a new
// password.
func (actor CognitoActions) ConfirmForgotPassword(clientId string, code string,
userName string, password string) error {
    _, err := actor.CognitoClient.ConfirmForgotPassword(context.TODO(),
&cognitoidentityprovider.ConfirmForgotPasswordInput{
```



```
    ClientId:      aws.String(clientId),
    ConfirmationCode: aws.String(code),
    Password:      aws.String(password),
    Username:      aws.String(userName),
  })
  if err != nil {
    var invalidPassword *types.InvalidPasswordException
    if errors.As(err, &invalidPassword) {
      log.Println(*invalidPassword.Message)
    } else {
      log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
    }
  }
  return err
}

// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(userAccessToken string) error {
  _, err := actor.CognitoClient.DeleteUser(context.TODO(),
    &cognitoidentityprovider.DeleteUserInput{
      AccessToken: aws.String(userAccessToken),
    })
  if err != nil {
    log.Printf("Couldn't delete user. Here's why: %v\n", err)
  }
  return err
}

// AdminCreateUser uses administrator credentials to add a user to a user pool.
// This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(userPoolId string, userName string,
  userEmail string) error {
  _, err := actor.CognitoClient.AdminCreateUser(context.TODO(),
    &cognitoidentityprovider.AdminCreateUserInput{
      UserPoolId:      aws.String(userPoolId),
      Username:        aws.String(userName),
      MessageAction:   types.MessageActionTypeSuppress,
      UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
        aws.String(userEmail)}}},
```

```
    })
    if err != nil {
        var userExists *types.UsernameExistsException
        if errors.As(err, &userExists) {
            log.Printf("User %v already exists in the user pool.", userName)
            err = nil
        } else {
            log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
        }
    }
    return err
}

// AdminSetUserPassword uses administrator credentials to set a password for a
// user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(userPoolId string, userName
string, password string) error {
    _, err := actor.CognitoClient.AdminSetUserPassword(context.TODO(),
&cognitoidentityprovider.AdminSetUserPasswordInput{
    Password:    aws.String(password),
    UserPoolId:  aws.String(userPoolId),
    Username:    aws.String(userName),
    Permanent:   true,
})
    if err != nil {
        var invalidPassword *types.InvalidPasswordException
        if errors.As(err, &invalidPassword) {
            log.Println(*invalidPassword.Message)
        } else {
            log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
err)
        }
    }
    return err
}
```

Créez une structure qui enveloppe les actions DynamoDB.

```
// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)
// actions
// used in the examples.
type DynamoActions struct {
    DynamoClient *dynamodb.Client
}

// User defines structured user data.
type User struct {
    UserName string
    UserEmail string
    LastLogin *LoginInfo `dynamodbav:",omitempty"`
}

// LoginInfo defines structured custom login data.
type LoginInfo struct {
    UserPoolId string
    ClientId string
    Time string
}

// UserList defines a list of users.
type UserList struct {
    Users []User
}

// UserNameList returns the usernames contained in a UserList as a list of
// strings.
func (users *UserList) UserNameList() []string {
    names := make([]string, len(users.Users))
    for i := 0; i < len(users.Users); i++ {
        names[i] = users.Users[i].UserName
    }
    return names
}

// PopulateTable adds a set of test users to the table.
func (actor DynamoActions) PopulateTable(tableName string) error {
    var err error
    var item map[string]types.AttributeValue
    var writeReqs []types.WriteRequest
    for i := 1; i < 4; i++ {
```

```

    item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
%v", i), UserEmail: fmt.Sprintf("test_email_%v@example.com", i)})
    if err != nil {
        log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
err)
        return err
    }
    writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
&types.PutRequest{Item: item}})
}
_, err = actor.DynamoClient.BatchWriteItem(context.TODO(),
&dynamodb.BatchWriteItemInput{
    RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
})
if err != nil {
    log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
tableName, err)
}
return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(tableName string) (UserList, error) {
    var userList UserList
    output, err := actor.DynamoClient.Scan(context.TODO(), &dynamodb.ScanInput{
        TableName: aws.String(tableName),
    })
    if err != nil {
        log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
err)
    } else {
        err = attributevalue.UnmarshallListOfMaps(output.Items, &userList.Users)
        if err != nil {
            log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
        }
    }
    return userList, err
}

// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(tableName string, user User) error {
    userItem, err := attributevalue.MarshalMap(user)
    if err != nil {
        log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
    }
}

```

```

}
_, err = actor.DynamoClient.PutItem(context.TODO(), &dynamodb.PutItemInput{
    Item:      userItem,
    TableName: aws.String(tableName),
})
if err != nil {
    log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
}
return err
}

```

Créez une structure qui englobe les actions CloudWatch Logs.

```

type CloudWatchLogsActions struct {
    CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(functionName string)
(types.LogStream, error) {
    var logStream types.LogStream
    logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
    output, err := actor.CwlClient.DescribeLogStreams(context.TODO(),
    &cloudwatchlogs.DescribeLogStreamsInput{
        Descending:  aws.Bool(true),
        Limit:        aws.Int32(1),
        LogGroupName: aws.String(logGroupName),
        OrderBy:     types.OrderByLastEventTime,
    })
    if err != nil {
        log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
        logGroupName, err)
    } else {
        logStream = output.LogStreams[0]
    }
    return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
stream.

```

```

func (actor CloudWatchLogsActions) GetLogEvents(functionName string,
logStreamName string, eventCount int32) (
[]types.OutputLogEvent, error) {
var events []types.OutputLogEvent
logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
output, err := actor.CwlClient.GetLogEvents(context.TODO(),
&cloudwatchlogs.GetLogEventsInput{
LogStreamName: aws.String(logStreamName),
Limit:         aws.Int32(eventCount),
LogGroupName:  aws.String(logGroupName),
})
if err != nil {
log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
logStreamName, err)
} else {
events = output.Events
}
return events, err
}

```

Créez une structure qui englobe les actions. AWS CloudFormation

```

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
structured format.
func (actor CloudFormationActions) GetOutputs(stackName string) StackOutputs {
output, err := actor.CfnClient.DescribeStacks(context.TODO(),
&cloudformation.DescribeStacksInput{
StackName: aws.String(stackName),
})
if err != nil || len(output.Stacks) == 0 {
log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
stackName, err)
}
}

```

```
stackOutputs := StackOutputs{}
for _, out := range output.Stacks[0].Outputs {
    stackOutputs[*out.OutputKey] = *out.OutputValue
}
return stackOutputs
}
```

Nettoyez les ressources.

```
// Resources keeps track of AWS resources created during an example and handles
// cleanup when the example finishes.
type Resources struct {
    userPoolId      string
    userAccessTokens []string
    triggers        []actions.Trigger

    cognitoActor *actions.CognitoActions
    questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
demotools.IQuestioner) {
    resources.userAccessTokens = []string{}
    resources.triggers = []actions.Trigger{}
    resources.cognitoActor = cognitoActor
    resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
func (resources *Resources) Cleanup() {
    defer func() {
        if r := recover(); r != nil {
            log.Printf("Something went wrong during cleanup.\n%v\n", r)
            log.Println("Use the AWS Management Console to remove any remaining resources
\n" +
                "that were created for this scenario.")
        }
    }()
}
```

```
wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
resources that were created "+
"during this demo (y/n)?", "y")
if wantDelete {
    for _, accessToken := range resources.userAccessTokens {
        err := resources.cognitoActor.DeleteUser(accessToken)
        if err != nil {
            log.Println("Couldn't delete user during cleanup.")
            panic(err)
        }
        log.Println("Deleted user.")
    }
    triggerList := make([]actions.TriggerInfo, len(resources.triggers))
    for i := 0; i < len(resources.triggers); i++ {
        triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
HandlerArn: nil}
    }
    err := resources.cognitoActor.UpdateTriggers(resources.userPoolId,
triggerList...)
    if err != nil {
        log.Println("Couldn't update Cognito triggers during cleanup.")
        panic(err)
    }
    log.Println("Removed Cognito triggers from user pool.")
} else {
    log.Println("Be sure to remove resources when you're done with them to avoid
unexpected charges!")
}
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Go .
 - [ConfirmForgotPassword](#)
 - [DeleteUser](#)
 - [ForgotPassword](#)
 - [InitiateAuth](#)
 - [SignUp](#)
 - [UpdateUserPool](#)

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Inscrire un utilisateur auprès d'un groupe d'utilisateurs Amazon Cognito qui nécessite l'authentification multifacteur à l'aide d'un SDK AWS

Les exemples de code suivants montrent comment :

- Inscrivez et confirmez un utilisateur avec un nom d'utilisateur, un mot de passe et une adresse e-mail.
- Configurez l'authentification multifactorielle en associant une application MFA à l'utilisateur.
- Connectez-vous à l'aide d'un mot de passe et d'un code MFA.

.NET

AWS SDK for .NET

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
namespace CognitoBasics;

public class CognitoBasics
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for Amazon Cognito.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
                        LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
                        LogLevel.Trace))
```

```
.ConfigureServices( (_, services) =>
services.AddAWSService<IAmazonCognitoIdentityProvider>()
.AddTransient<CognitoWrapper>()
)
.Build();

logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
.CreateLogger<CognitoBasics>();

var configuration = new ConfigurationBuilder()
.SetBasePath(Directory.GetCurrentDirectory())
.AddJsonFile("settings.json") // Load settings from .json file.
.AddJsonFile("settings.local.json",
true) // Optionally load local settings.
.Build();

var cognitoWrapper = host.Services.GetRequiredService<CognitoWrapper>();

Console.WriteLine(new string('-', 80));
UiMethods.DisplayOverview();
Console.WriteLine(new string('-', 80));

// clientId - The app client Id value that you get from the AWS CDK
script.
var clientId = configuration["ClientId"]; // **** REPLACE WITH CLIENT ID
VALUE FROM CDK SCRIPT";

// poolId - The pool Id that you get from the AWS CDK script.
var poolId = configuration["PoolId"]!; // **** REPLACE WITH POOL ID VALUE
FROM CDK SCRIPT";
var userName = configuration["UserName"];
var password = configuration["Password"];
var email = configuration["Email"];

// If the username wasn't set in the configuration file,
// get it from the user now.
if (userName is null)
{
do
{
Console.Write("Username: ");
userName = Console.ReadLine();
}
while (string.IsNullOrEmpty(userName));
```

```
    }
    Console.WriteLine($"\\nUsername: {userName}");

    // If the password wasn't set in the configuration file,
    // get it from the user now.
    if (password is null)
    {
        do
        {
            Console.Write("Password: ");
            password = Console.ReadLine();
        }
        while (string.IsNullOrEmpty(password));
    }

    // If the email address wasn't set in the configuration file,
    // get it from the user now.
    if (email is null)
    {
        do
        {
            Console.Write("Email: ");
            email = Console.ReadLine();
        } while (string.IsNullOrEmpty(email));
    }

    // Now sign up the user.
    Console.WriteLine($"\\nSigning up {userName} with email address:
{email}");
    await cognitoWrapper.SignUpAsync(clientId, userName, password, email);

    // Add the user to the user pool.
    Console.WriteLine($"Adding {userName} to the user pool");
    await cognitoWrapper.GetAdminUserAsync(userName, poolId);

    UiMethods.DisplayTitle("Get confirmation code");
    Console.WriteLine($"Conformation code sent to {userName}.");
    Console.Write("Would you like to send a new code? (Y/N) ");
    var answer = Console.ReadLine();

    if (answer!.ToLower() == "y")
    {
        await cognitoWrapper.ResendConfirmationCodeAsync(clientId, userName);
        Console.WriteLine("Sending a new confirmation code");
    }
}
```

```
}

Console.WriteLine("Enter confirmation code (from Email): ");
var code = Console.ReadLine();

await cognitoWrapper.ConfirmSignupAsync(clientId, code, userName);

UiMethods.DisplayTitle("Checking status");
Console.WriteLine($"Rechecking the status of {userName} in the user
pool");
await cognitoWrapper.GetAdminUserAsync(userName, poolId);

Console.WriteLine($"Setting up authenticator for {userName} in the user
pool");
var setupResponse = await cognitoWrapper.InitiateAuthAsync(clientId,
userName, password);

var setupSession = await
cognitoWrapper.AssociateSoftwareTokenAsync(setupResponse.Session);
Console.WriteLine("Enter the 6-digit code displayed in Google Authenticator:
");
var setupCode = Console.ReadLine();

var setupResult = await
cognitoWrapper.VerifySoftwareTokenAsync(setupSession, setupCode);
Console.WriteLine($"Setup status: {setupResult}");

Console.WriteLine($"Now logging in {userName} in the user pool");
var authSession = await cognitoWrapper.AdminInitiateAuthAsync(clientId,
poolId, userName, password);

Console.WriteLine("Enter a new 6-digit code displayed in Google
Authenticator: ");
var authCode = Console.ReadLine();

var authResult = await
cognitoWrapper.AdminRespondToAuthChallengeAsync(userName, clientId, authCode,
authSession, poolId);
Console.WriteLine($"Authenticated and received access token:
{authResult.AccessToken}");

Console.WriteLine(new string('-', 80));
Console.WriteLine("Cognito scenario is complete.");
Console.WriteLine(new string('-', 80));
```

```
    }
}

using System.Net;

namespace CognitoActions;

/// <summary>
/// Methods to perform Amazon Cognito Identity Provider actions.
/// </summary>
public class CognitoWrapper
{
    private readonly IAmazonCognitoIdentityProvider _cognitoService;

    /// <summary>
    /// Constructor for the wrapper class containing Amazon Cognito actions.
    /// </summary>
    /// <param name="cognitoService">The Amazon Cognito client object.</param>
    public CognitoWrapper(IAmazonCognitoIdentityProvider cognitoService)
    {
        _cognitoService = cognitoService;
    }

    /// <summary>
    /// List the Amazon Cognito user pools for an account.
    /// </summary>
    /// <returns>A list of UserPoolDescriptionType objects.</returns>
    public async Task<List<UserPoolDescriptionType>> ListUserPoolsAsync()
    {
        var userPools = new List<UserPoolDescriptionType>();

        var userPoolsPaginator = _cognitoService.Paginators.ListUserPools(new
ListUserPoolsRequest());

        await foreach (var response in userPoolsPaginator.Responses)
        {
            userPools.AddRange(response.UserPools);
        }

        return userPools;
    }
}
```

```
/// <summary>
/// Get a list of users for the Amazon Cognito user pool.
/// </summary>
/// <param name="userPoolId">The user pool ID.</param>
/// <returns>A list of users.</returns>
public async Task<List<UserType>> ListUsersAsync(string userPoolId)
{
    var request = new ListUsersRequest
    {
        UserPoolId = userPoolId
    };

    var users = new List<UserType>();

    var usersPaginator = _cognitoService.Paginators.ListUsers(request);
    await foreach (var response in usersPaginator.Responses)
    {
        users.AddRange(response.Users);
    }

    return users;
}

/// <summary>
/// Respond to an admin authentication challenge.
/// </summary>
/// <param name="userName">The name of the user.</param>
/// <param name="clientId">The client ID.</param>
/// <param name="mfaCode">The multi-factor authentication code.</param>
/// <param name="session">The current application session.</param>
/// <param name="clientId">The user pool ID.</param>
/// <returns>The result of the authentication response.</returns>
public async Task<AuthenticationResultType> AdminRespondToAuthChallengeAsync(
    string userName,
    string clientId,
    string mfaCode,
    string session,
    string userPoolId)
{
    Console.WriteLine("SOFTWARE_TOKEN_MFA challenge is generated");

    var challengeResponses = new Dictionary<string, string>();
    challengeResponses.Add("USERNAME", userName);
}
```

```
        challengeResponses.Add("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

        var respondToAuthChallengeRequest = new
AdminRespondToAuthChallengeRequest
        {
            ChallengeName = ChallengeNameType.SOFTWARE_TOKEN_MFA,
            ClientId = clientId,
            ChallengeResponses = challengeResponses,
            Session = session,
            UserPoolId = userPoolId,
        };

        var response = await
_cognitoService.AdminRespondToAuthChallengeAsync(respondToAuthChallengeRequest);
        Console.WriteLine($"Response to Authentication
{response.AuthenticationResult.TokenType}");
        return response.AuthenticationResult;
    }

    /// <summary>
    /// Verify the TOTP and register for MFA.
    /// </summary>
    /// <param name="session">The name of the session.</param>
    /// <param name="code">The MFA code.</param>
    /// <returns>The status of the software token.</returns>
    public async Task<VerifySoftwareTokenResponseType>
VerifySoftwareTokenAsync(string session, string code)
    {
        var tokenRequest = new VerifySoftwareTokenRequest
        {
            UserCode = code,
            Session = session,
        };

        var verifyResponse = await
_cognitoService.VerifySoftwareTokenAsync(tokenRequest);

        return verifyResponse.Status;
    }

    /// <summary>
    /// Get an MFA token to authenticate the user with the authenticator.
```

```
/// </summary>
/// <param name="session">The session name.</param>
/// <returns>The session name.</returns>
public async Task<string> AssociateSoftwareTokenAsync(string session)
{
    var softwareTokenRequest = new AssociateSoftwareTokenRequest
    {
        Session = session,
    };

    var tokenResponse = await
_cognitoService.AssociateSoftwareTokenAsync(softwareTokenRequest);
    var secretCode = tokenResponse.SecretCode;

    Console.WriteLine($"Use the following secret code to set up the
authenticator: {secretCode}");

    return tokenResponse.Session;
}

/// <summary>
/// Initiate an admin auth request.
/// </summary>
/// <param name="clientId">The client ID to use.</param>
/// <param name="userPoolId">The ID of the user pool.</param>
/// <param name="userName">The username to authenticate.</param>
/// <param name="password">The user's password.</param>
/// <returns>The session to use in challenge-response.</returns>
public async Task<string> AdminInitiateAuthAsync(string clientId, string
userPoolId, string userName, string password)
{
    var authParameters = new Dictionary<string, string>();
    authParameters.Add("USERNAME", userName);
    authParameters.Add("PASSWORD", password);

    var request = new AdminInitiateAuthRequest
    {
        ClientId = clientId,
        UserPoolId = userPoolId,
        AuthParameters = authParameters,
        AuthFlow = AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
    };
};
```



```
        var response = await _cognitoService.AdminInitiateAuthAsync(request);
        return response.Session;
    }

    /// <summary>
    /// Initiate authorization.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The name of the user who is authenticating.</
param>
    /// <param name="password">The password for the user who is authenticating.</
param>
    /// <returns>The response from the initiate auth request.</returns>
    public async Task<InitiateAuthResponse> InitiateAuthAsync(string clientId,
string userName, string password)
    {
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var authRequest = new InitiateAuthRequest

        {
            ClientId = clientId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.USER_PASSWORD_AUTH,
        };

        var response = await _cognitoService.InitiateAuthAsync(authRequest);
        Console.WriteLine($"Result Challenge is : {response.ChallengeName}");

        return response;
    }

    /// <summary>
    /// Confirm that the user has signed up.
    /// </summary>
    /// <param name="clientId">The Id of this application.</param>
    /// <param name="code">The confirmation code sent to the user.</param>
    /// <param name="userName">The username.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ConfirmSignupAsync(string clientId, string code,
string userName)
    {
```

```
var signUpRequest = new ConfirmSignUpRequest
{
    ClientId = clientId,
    ConfirmationCode = code,
    Username = userName,
};

var response = await _cognitoService.ConfirmSignUpAsync(signUpRequest);
if (response.HttpStatusCode == HttpStatusCode.OK)
{
    Console.WriteLine($"{userName} was confirmed");
    return true;
}
return false;
}

/// <summary>
/// Initiates and confirms tracking of the device.
/// </summary>
/// <param name="accessToken">The user's access token.</param>
/// <param name="deviceKey">The key of the device from Amazon Cognito.</
param>
/// <param name="deviceName">The device name.</param>
/// <returns></returns>
public async Task<bool> ConfirmDeviceAsync(string accessToken, string
deviceKey, string deviceName)
{
    var request = new ConfirmDeviceRequest
    {
        AccessToken = accessToken,
        DeviceKey = deviceKey,
        DeviceName = deviceName
    };

    var response = await _cognitoService.ConfirmDeviceAsync(request);
    return response.UserConfirmationNecessary;
}

/// <summary>
/// Send a new confirmation code to a user.
/// </summary>
/// <param name="clientId">The Id of the client application.</param>
```

```
    /// <param name="userName">The username of user who will receive the code.</  
param>  
    /// <returns>The delivery details.</returns>  
    public async Task<CodeDeliveryDetailsType> ResendConfirmationCodeAsync(string  
clientId, string userName)  
    {  
        var codeRequest = new ResendConfirmationCodeRequest  
        {  
            ClientId = clientId,  
            Username = userName,  
        };  
  
        var response = await  
_cognitoService.ResendConfirmationCodeAsync(codeRequest);  
  
        Console.WriteLine($"Method of delivery is  
{response.CodeDeliveryDetails.DeliveryMedium}");  
  
        return response.CodeDeliveryDetails;  
    }  
  
    /// <summary>  
    /// Get the specified user from an Amazon Cognito user pool with  
administrator access.  
    /// </summary>  
    /// <param name="userName">The name of the user.</param>  
    /// <param name="poolId">The Id of the Amazon Cognito user pool.</param>  
    /// <returns>Async task.</returns>  
    public async Task<UserStatusType> GetAdminUserAsync(string userName, string  
poolId)  
    {  
        AdminGetUserRequest userRequest = new AdminGetUserRequest  
        {  
            Username = userName,  
            UserPoolId = poolId,  
        };  
  
        var response = await _cognitoService.AdminGetUserAsync(userRequest);  
  
        Console.WriteLine($"User status {response.UserStatus}");  
        return response.UserStatus;  
    }  
}
```

```
/// <summary>
/// Sign up a new user.
/// </summary>
/// <param name="clientId">The client Id of the application.</param>
/// <param name="userName">The username to use.</param>
/// <param name="password">The user's password.</param>
/// <param name="email">The email address of the user.</param>
/// <returns>A Boolean value indicating whether the user was confirmed.</
returns>
    public async Task<bool> SignUpAsync(string clientId, string userName, string
password, string email)
    {
        var userAttrs = new AttributeType
        {
            Name = "email",
            Value = email,
        };

        var userAttrsList = new List<AttributeType>();

        userAttrsList.Add(userAttrs);

        var signUpRequest = new SignUpRequest
        {
            UserAttributes = userAttrsList,
            Username = userName,
            ClientId = clientId,
            Password = password
        };

        var response = await _cognitoService.SignUpAsync(signUpRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for .NET .
 - [AdminGetUser](#)

- [AdminInitiateAuth](#)
- [AdminRespondToAuthChallenge](#)
- [AssociateSoftwareToken](#)
- [ConfirmDevice](#)
- [ConfirmSignUp](#)
- [InitiateAuth](#)
- [ListUsers](#)
- [ResendConfirmationCode](#)
- [RespondToAuthChallenge](#)
- [SignUp](#)
- [VerifySoftwareToken](#)

C++

SDK pour C++

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

//! Scenario that adds a user to an Amazon Cognito user pool.
/*!
 \sa gettingStartedWithUserPools()
 \param clientID: Client ID associated with an Amazon Cognito user pool.
 \param userPoolID: An Amazon Cognito user pool ID.
 \param clientConfig: Aws client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::Cognito::gettingStartedWithUserPools(const Aws::String &clientID,
                                                    const Aws::String &userPoolID,
```

```

                                                    const
Aws::Client::ClientConfiguration &clientConfig) {
    printAsterisksLine();
    std::cout
        << "Welcome to the Amazon Cognito example scenario."
        << std::endl;
    printAsterisksLine();

    std::cout
        << "This scenario will add a user to an Amazon Cognito user pool."
        << std::endl;
    const Aws::String userName = askQuestion("Enter a new username: ");
    const Aws::String password = askQuestion("Enter a new password: ");
    const Aws::String email = askQuestion("Enter a valid email for the user: ");

    std::cout << "Signing up " << userName << std::endl;

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);
    bool userExists = false;
    do {
        // 1. Add a user with a username, password, and email address.
        Aws::CognitoIdentityProvider::Model::SignUpRequest request;
        request.AddUserAttributes(
            Aws::CognitoIdentityProvider::Model::AttributeType().WithName(
                "email").WithValue(email));
        request.SetUsername(userName);
        request.SetPassword(password);
        request.SetClientId(clientID);
        Aws::CognitoIdentityProvider::Model::SignUpOutcome outcome =
            client.SignUp(request);

        if (outcome.IsSuccess()) {
            std::cout << "The signup request for " << userName << " was
successful."
                << std::endl;
        }
        else if (outcome.GetError().GetErrorType() ==
Aws::CognitoIdentityProvider::CognitoIdentityProviderErrors::USERNAME_EXISTS) {
            std::cout
                << "The username already exists. Please enter a different
username."
                << std::endl;
        }
    }
}
```

```
        userExists = true;
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::SignUpRequest. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
        return false;
    }
} while (userExists);

printAsterisksLine();
std::cout << "Retrieving status of " << userName << " in the user pool."
          << std::endl;
// 2. Confirm that the user was added to the user pool.
if (!checkAdminUserStatus(userName, userPoolID, client)) {
    return false;
}

std::cout << "A confirmation code was sent to " << email << "." << std::endl;

bool resend = askYesNoQuestion("Would you like to send a new code? (y/n) ");
if (resend) {
    // Request a resend of the confirmation code to the email address.
    (ResendConfirmationCode)
    Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeRequest
request;
    request.SetUsername(userName);
    request.SetClientId(clientID);

    Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeOutcome
outcome =
        client.ResendConfirmationCode(request);

    if (outcome.IsSuccess()) {
        std::cout
            << "CognitoIdentityProvider::ResendConfirmationCode was
successful."
            << std::endl;
    }
    else {
        std::cerr << "Error with
CognitoIdentityProvider::ResendConfirmationCode. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
}
```

```
        return false;
    }
}

printAsterisksLine();

{
    // 4. Send the confirmation code that's received in the email.
(ConfirmSignUp)
    const Aws::String confirmationCode = askQuestion(
        "Enter the confirmation code that was emailed: ");
    Aws::CognitoIdentityProvider::Model::ConfirmSignUpRequest request;
    request.SetClientId(clientID);
    request.SetConfirmationCode(confirmationCode);
    request.SetUsername(userName);

    Aws::CognitoIdentityProvider::Model::ConfirmSignUpOutcome outcome =
        client.ConfirmSignUp(request);

    if (outcome.IsSuccess()) {
        std::cout << "ConfirmSignup was Successful."
            << std::endl;
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::ConfirmSignUp. "
            << outcome.GetError().GetMessage()
            << std::endl;
        return false;
    }
}

std::cout << "Rechecking the status of " << userName << " in the user pool."
    << std::endl;
if (!checkAdminUserStatus(userName, userPoolID, client)) {
    return false;
}

printAsterisksLine();

std::cout << "Initiating authorization using the username and password."
    << std::endl;

Aws::String session;
// 5. Initiate authorization with username and password. (AdminInitiateAuth)
```



```

    if (!adminInitiateAuthorization(clientID, userPoolID,  userName, password,
    session, client)) {
        return false;
    }

    printAsterisksLine();

    std::cout
        << "Starting setup of time-based one-time password (TOTP) multi-
    factor authentication (MFA)."
        << std::endl;

    {
        // 6. Request a setup key for one-time password (TOTP)
        //    multi-factor authentication (MFA). (AssociateSoftwareToken)
        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenRequest
    request;
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenOutcome
    outcome =
            client.AssociateSoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout
                << "Enter this setup key into an authenticator app, for
            example Google Authenticator."
                << std::endl;
            std::cout << "Setup key: " << outcome.GetResult().GetSecretCode()
                << std::endl;
#ifdef USING_QR
            printAsterisksLine();
            std::cout << "\nOr scan the QR code in the file '" << QR_CODE_PATH <<
            "."
                << std::endl;

            saveQRCode(std::string("otpauth://totp/") + userName + "?secret=" +
                outcome.GetResult().GetSecretCode());
#endif // USING_QR
            session = outcome.GetResult().GetSession();
        }
        else {
            std::cerr << "Error with
        CognitoIdentityProvider::AssociateSoftwareToken. "

```

```
        << outcome.GetError().GetMessage()
        << std::endl;
    return false;
}
}
askQuestion("Type enter to continue...", alwaysTrueTest);

printAsterisksLine();

{
    Aws::String userCode = askQuestion(
        "Enter the 6 digit code displayed in the authenticator app: ");

    // 7. Send the MFA code copied from an authenticator app.
(VerifySoftwareToken)
    Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenRequest request;
    request.SetUserCode(userCode);
    request.SetSession(session);

    Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenOutcome outcome =
        client.VerifySoftwareToken(request);

    if (outcome.IsSuccess()) {
        std::cout << "Verification of the code was successful."
            << std::endl;
        session = outcome.GetResult().GetSession();
    }
    else {
        std::cerr << "Error with
CognitoIdentityProvider::VerifySoftwareToken. "
            << outcome.GetError().GetMessage()
            << std::endl;
        return false;
    }
}

printAsterisksLine();
std::cout << "You have completed the MFA authentication setup." << std::endl;
std::cout << "Now, sign in." << std::endl;

// 8. Initiate authorization again with username and password.
(AdminInitiateAuth)
    if (!adminInitiateAuthorization(clientID, userPoolID, userName, password,
session, client)) {
```

```
        return false;
    }

    Aws::String accessToken;
    {
        Aws::String mfaCode = askQuestion(
            "Re-enter the 6 digit code displayed in the authenticator app:
");

        // 9. Send a new MFA code copied from an authenticator app.
        (AdminRespondToAuthChallenge)
        Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeRequest
        request;
        request.AddChallengeResponses("USERNAME", userName);
        request.AddChallengeResponses("SOFTWARE_TOKEN_MFA_CODE", mfaCode);
        request.SetChallengeName(

Aws::CognitoIdentityProvider::Model::ChallengeNameType::SOFTWARE_TOKEN_MFA);
        request.SetClientId(clientID);
        request.SetUserPoolId(userPoolID);
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeOutcome
        outcome =
            client.AdminRespondToAuthChallenge(request);

        if (outcome.IsSuccess()) {
            std::cout << "Here is the response to the challenge.\n" <<

outcome.GetResult().GetAuthenticationResult().Jsonize().View().WriteReadable()
            << std::endl;

            accessToken =
outcome.GetResult().GetAuthenticationResult().GetAccessToken();
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::AdminRespondToAuthChallenge. "
                << outcome.GetError().GetMessage()
                << std::endl;
            return false;
        }

        std::cout << "You have successfully added a user to Amazon Cognito."
```

```

        << std::endl;
    }

    if (askYesNoQuestion("Would you like to delete the user that you just added?
(y/n) ")) {
        // 10. Delete the user that you just added. (DeleteUser)
        Aws::CognitoIdentityProvider::Model::DeleteUserRequest request;
        request.SetAccessToken(accessToken);

        Aws::CognitoIdentityProvider::Model::DeleteUserOutcome outcome =
            client.DeleteUser(request);

        if (outcome.IsSuccess()) {
            std::cout << "The user " << userName << " was deleted."
                << std::endl;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::DeleteUser. "
                << outcome.GetError().GetMessage()
                << std::endl;
        }
    }

    return true;
}

//! Routine which checks the user status in an Amazon Cognito user pool.
/*!
 \sa checkAdminUserStatus()
 \param userName: A username.
 \param userPoolID: An Amazon Cognito user pool ID.
 \return bool: Successful completion.
 */
bool AwsDoc::Cognito::checkAdminUserStatus(const Aws::String &userName,
                                           const Aws::String &userPoolID,
                                           const
Aws::CognitoIdentityProvider::CognitoIdentityProviderClient &client) {
    Aws::CognitoIdentityProvider::Model::AdminGetUserRequest request;
    request.SetUsername(userName);
    request.SetUserPoolId(userPoolID);

    Aws::CognitoIdentityProvider::Model::AdminGetUserOutcome outcome =
        client.AdminGetUser(request);

```

```

    if (outcome.IsSuccess()) {
        std::cout << "The status for " << userName << " is " <<

Aws::CognitoIdentityProvider::Model::UserStatusTypeMapper::GetNameForUserStatusType(
        outcome.GetResult().GetUserStatus()) << std::endl;
        std::cout << "Enabled is " << outcome.GetResult().GetEnabled() <<
std::endl;
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminGetUser. "
        << outcome.GetError().GetMessage()
        << std::endl;
    }

    return outcome.IsSuccess();
}

//! Routine which starts authorization of an Amazon Cognito user.
//! This routine requires administrator credentials.
/*!
 \sa adminInitiateAuthorization()
 \param clientID: Client ID of tracked device.
 \param userPoolID: An Amazon Cognito user pool ID.
 \param userName: A username.
 \param password: A password.
 \param sessionResult: String to receive a session token.
 \return bool: Successful completion.
 */
bool AwsDoc::Cognito::adminInitiateAuthorization(const Aws::String &clientID,
                                                const Aws::String &userPoolID,
                                                const Aws::String &userName,
                                                const Aws::String &password,
                                                Aws::String &sessionResult,
                                                const
Aws::CognitoIdentityProvider::CognitoIdentityProviderClient &client) {
    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthRequest request;
    request.SetClientId(clientID);
    request.SetUserPoolId(userPoolID);
    request.AddAuthParameters("USERNAME", userName);
    request.AddAuthParameters("PASSWORD", password);
    request.SetAuthFlow(

Aws::CognitoIdentityProvider::Model::AuthFlowType::ADMIN_USER_PASSWORD_AUTH);

```

```
Aws::CognitoIdentityProvider::Model::AdminInitiateAuthOutcome outcome =
    client.AdminInitiateAuth(request);


if (outcome.IsSuccess()) {
    std::cout << "Call to AdminInitiateAuth was successful." << std::endl;
    sessionResult = outcome.GetResult().GetSession();
}
else {
    std::cerr << "Error with CognitoIdentityProvider::AdminInitiateAuth. "
        << outcome.GetError().GetMessage()
        << std::endl;
}

return outcome.IsSuccess();
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for C++ .
 - [AdminGetUser](#)
 - [AdminInitiateAuth](#)
 - [AdminRespondToAuthChallenge](#)
 - [AssociateSoftwareToken](#)
 - [ConfirmDevice](#)
 - [ConfirmSignUp](#)
 - [InitiateAuth](#)
 - [ListUsers](#)
 - [ResendConfirmationCode](#)
 - [RespondToAuthChallenge](#)
 - [SignUp](#)
 - [VerifySoftwareToken](#)

Java

SDK pour Java 2.x

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AdminGetUserRequest;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AdminGetUserResponse;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AdminInitiateAuthRequest;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AdminInitiateAuthResponse;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AdminRespondToAuthChallenge;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AdminRespondToAuthChallengeResponse;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AssociateSoftwareTokenRequest;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AssociateSoftwareTokenResponse;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AttributeType;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.AuthFlowType;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.ChallengeNameType;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderException;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.ConfirmSignUpRequest;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.ResendConfirmationCodeRequest;
```

```
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.ResendConfirmationCodeResp
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.SignUpRequest;
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.VerifySoftwareTokenRequest
import
    software.amazon.awssdk.services.cognitoidentityprovider.model.VerifySoftwareTokenResponse
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 *
 * TIP: To set up the required user pool, run the AWS Cloud Development Kit (AWS
 * CDK) script provided in this GitHub repo at
 * resources/cdk/cognito\_scenario\_user\_pool\_with\_mfa.
 *
 * This code example performs the following operations:
 *
 * 1. Invokes the signUp method to sign up a user.
 * 2. Invokes the adminGetUser method to get the user's confirmation status.
 * 3. Invokes the ResendConfirmationCode method if the user requested another
 * code.
 * 4. Invokes the confirmSignUp method.
 * 5. Invokes the AdminInitiateAuth to sign in. This results in being prompted
 * to set up TOTP (time-based one-time password). (The response is
 * "ChallengeName": "MFA_SETUP").
 * 6. Invokes the AssociateSoftwareToken method to generate a TOTP MFA private
 * key. This can be used with Google Authenticator.
 * 7. Invokes the VerifySoftwareToken method to verify the TOTP and register for
 * MFA.
 * 8. Invokes the AdminInitiateAuth to sign in again. This results in being
```



```
* prompted to submit a TOTP (Response: "ChallengeName": "SOFTWARE_TOKEN_MFA").
* 9. Invokes the AdminRespondToAuthChallenge to get back a token.
*/

public class CognitoMVP {
    public static final String DASHES = new String(new char[80]).replace("\0",
    "-");

    public static void main(String[] args) throws NoSuchAlgorithmException,
    InvalidKeyException {
        final String usage = ""

            Usage:
                <clientId> <poolId>

            Where:
                clientId - The app client Id value that you can get from the
AWS CDK script.
                poolId - The pool Id that you can get from the AWS CDK
script.\s
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String clientId = args[0];
        String poolId = args[1];
        CognitoIdentityProviderClient identityProviderClient =
CognitoIdentityProviderClient.builder()
            .region(Region.US_EAST_1)
            .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon Cognito example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("**** Enter your user name");
        Scanner in = new Scanner(System.in);
        String userName = in.nextLine();

        System.out.println("**** Enter your password");
```

```
String password = in.nextLine();

System.out.println("*** Enter your email");
String email = in.nextLine();

System.out.println("1. Signing up " + userName);
signUp(identityProviderClient, clientId, userName, password, email);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("2. Getting " + userName + " in the user pool");
getAdminUser(identityProviderClient, userName, poolId);

System.out
    .println("*** Confirmation code sent to " + userName + ". Would
you like to send a new code? (Yes/No)");
System.out.println(DASHES);

System.out.println(DASHES);
String ans = in.nextLine();

if (ans.compareTo("Yes") == 0) {
    resendConfirmationCode(identityProviderClient, clientId, userName);
    System.out.println("3. Sending a new confirmation code");
}
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("4. Enter confirmation code that was emailed");
String code = in.nextLine();
confirmSignUp(identityProviderClient, clientId, code, userName);
System.out.println("Rechecking the status of " + userName + " in the user
pool");
getAdminUser(identityProviderClient, userName, poolId);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("5. Invokes the initiateAuth to sign in");
AdminInitiateAuthResponse authResponse =
initiateAuth(identityProviderClient, clientId, userName, password,
    poolId);
String mySession = authResponse.session();
System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("6. Invokes the AssociateSoftwareToken method to
generate a TOTP key");
        String newSession = getSecretForAppMFA(identityProviderClient,
mySession);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Enter the 6-digit code displayed in Google
Authenticator");
        String myCode = in.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Verify the TOTP and register for MFA");
        verifyTOTP(identityProviderClient, newSession, myCode);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Re-enter a 6-digit code displayed in Google
Authenticator");
        String mfaCode = in.nextLine();
        AdminInitiateAuthResponse authResponse1 =
initiateAuth(identityProviderClient, clientId, userName, password,
                poolId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9. Invokes the AdminRespondToAuthChallenge");
        String session2 = authResponse1.session();
        adminRespondToAuthChallenge(identityProviderClient, userName, clientId,
mfaCode, session2);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("All Amazon Cognito operations were successfully
performed");
        System.out.println(DASHES);
    }

    // Respond to an authentication challenge.
    public static void adminRespondToAuthChallenge(CognitoIdentityProviderClient
identityProviderClient,
            String userName, String clientId, String mfaCode, String session) {
```

```
System.out.println("SOFTWARE_TOKEN_MFA challenge is generated");
Map<String, String> challengeResponses = new HashMap<>();

challengeResponses.put("USERNAME", userName);
challengeResponses.put("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

AdminRespondToAuthChallengeRequest respondToAuthChallengeRequest =
AdminRespondToAuthChallengeRequest.builder()
    .challengeName(ChallengeNameType.SOFTWARE_TOKEN_MFA)
    .clientId(clientId)
    .challengeResponses(challengeResponses)
    .session(session)
    .build();

AdminRespondToAuthChallengeResponse respondToAuthChallengeResult =
identityProviderClient
    .adminRespondToAuthChallenge(respondToAuthChallengeRequest);

System.out.println("respondToAuthChallengeResult.getAuthenticationResult()"
    + respondToAuthChallengeResult.authenticationResult());
}

// Verify the TOTP and register for MFA.
public static void verifyTOTP(CognitoIdentityProviderClient
identityProviderClient, String session, String code) {
    try {
        VerifySoftwareTokenRequest tokenRequest =
VerifySoftwareTokenRequest.builder()
            .userCode(code)
            .session(session)
            .build();

        VerifySoftwareTokenResponse verifyResponse =
identityProviderClient.verifySoftwareToken(tokenRequest);
        System.out.println("The status of the token is " +
verifyResponse.statusAsString());

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

```
public static AdminInitiateAuthResponse
initiateAuth(CognitoIdentityProviderClient identityProviderClient,
             String clientId, String userName, String password, String userPoolId)
{
    try {
        Map<String, String> authParameters = new HashMap<>();
        authParameters.put("USERNAME", userName);
        authParameters.put("PASSWORD", password);

        AdminInitiateAuthRequest authRequest =
AdminInitiateAuthRequest.builder()
            .clientId(clientId)
            .userPoolId(userPoolId)
            .authParameters(authParameters)
            .authFlow(AuthFlowType.ADMIN_USER_PASSWORD_AUTH)
            .build();

        AdminInitiateAuthResponse response =
identityProviderClient.adminInitiateAuth(authRequest);
        System.out.println("Result Challenge is : " +
response.challengeName());
        return response;

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return null;
}

public static String getSecretForAppMFA(CognitoIdentityProviderClient
identityProviderClient, String session) {
    AssociateSoftwareTokenRequest softwareTokenRequest =
AssociateSoftwareTokenRequest.builder()
        .session(session)
        .build();

    AssociateSoftwareTokenResponse tokenResponse = identityProviderClient
        .associateSoftwareToken(softwareTokenRequest);
    String secretCode = tokenResponse.secretCode();
    System.out.println("Enter this token into Google Authenticator");
    System.out.println(secretCode);
    return tokenResponse.session();
}
```

```
    }

    public static void confirmSignUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String code,
    String userName) {
        try {
            ConfirmSignUpRequest signUpRequest = ConfirmSignUpRequest.builder()
                .clientId(clientId)
                .confirmationCode(code)
                .username(userName)
                .build();

            identityProviderClient.confirmSignUp(signUpRequest);
            System.out.println(userName + " was confirmed");

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void resendConfirmationCode(CognitoIdentityProviderClient
identityProviderClient, String clientId,
    String userName) {
        try {
            ResendConfirmationCodeRequest codeRequest =
ResendConfirmationCodeRequest.builder()
                .clientId(clientId)
                .username(userName)
                .build();

            ResendConfirmationCodeResponse response =
identityProviderClient.resendConfirmationCode(codeRequest);
            System.out.println("Method of delivery is " +
response.codeDeliveryDetails().deliveryMediumAsString());

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void signUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String userName,
```

```
        String password, String email) {
    AttributeType userAttrs = AttributeType.builder()
        .name("email")
        .value(email)
        .build();

    List<AttributeType> userAttrsList = new ArrayList<>();
    userAttrsList.add(userAttrs);
    try {
        SignUpRequest signUpRequest = SignUpRequest.builder()
            .userAttributes(userAttrsList)
            .username(userName)
            .clientId(clientId)
            .password(password)
            .build();

        identityProviderClient.signUp(signUpRequest);
        System.out.println("User has been signed up ");

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void getAdminUser(CognitoIdentityProviderClient
identityProviderClient, String userName,
    String poolId) {
    try {
        AdminGetUserRequest userRequest = AdminGetUserRequest.builder()
            .username(userName)
            .userPoolId(poolId)
            .build();

        AdminGetUserResponse response =
identityProviderClient.adminGetUser(userRequest);
        System.out.println("User status " + response.userStatusAsString());

    } catch (CognitoIdentityProviderException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
 - [AdminGetUser](#)
 - [AdminInitiateAuth](#)
 - [AdminRespondToAuthChallenge](#)
 - [AssociateSoftwareToken](#)
 - [ConfirmDevice](#)
 - [ConfirmSignUp](#)
 - [InitiateAuth](#)
 - [ListUsers](#)
 - [ResendConfirmationCode](#)
 - [RespondToAuthChallenge](#)
 - [SignUp](#)
 - [VerifySoftwareToken](#)

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Pour une expérience optimale, clonez le GitHub référentiel et exécutez cet exemple. Le code suivant représente un échantillon de l'exemple d'application complet.

```
import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { signUp } from "../../actions/sign-up.js";
import { FILE_USER_POOLS } from "./constants.js";
import { getSecondValuesFromEntries } from "@aws-doc-sdk-examples/lib/utils/util-csv.js";
```



```
const validateClient = (clientId) => {
  if (!clientId) {
    throw new Error(
      `App client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};

const validateUser = (username, password, email) => {
  if (!(username && password && email)) {
    throw new Error(
      `Username, password, and email must be provided as arguments to the 'sign-
up' command.`,
    );
  }
};

const signUpHandler = async (commands) => {
  const [_ , username, password, email] = commands;

  try {
    validateUser(username, password, email);
    /**
     * @type {string[]}
     */
    const values = getSecondValuesFromEntries(FILE_USER_POOLS);
    const clientId = values[0];
    validateClient(clientId);
    log(`Signing up.`);
    await signUp({ clientId, username, password, email });
    log(`Signed up. A confirmation email has been sent to: ${email}.`);
    log(`Run 'confirm-sign-up ${username} <code>' to confirm your account.`);
  } catch (err) {
    log(err);
  }
};

export { signUpHandler };

const signUp = ({ clientId, username, password, email }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new SignUpCommand({
    ClientId: clientId,
```

```
    Username: username,
    Password: password,
    UserAttributes: [{ Name: "email", Value: email }],
  });

  return client.send(command);
};

import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { confirmSignUp } from "../../actions/confirm-sign-up.js";
import { FILE_USER_POOLS } from "./constants.js";
import { getSecondValuesFromEntries } from "@aws-doc-sdk-examples/lib/utils/util-csv.js";

const validateClient = (clientId) => {
  if (!clientId) {
    throw new Error(
      `App client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};

const validateUser = (username) => {
  if (!username) {
    throw new Error(
      `Username name is missing. It must be provided as an argument to the 'confirm-sign-up' command.`,
    );
  }
};

const validateCode = (code) => {
  if (!code) {
    throw new Error(
      `Verification code is missing. It must be provided as an argument to the 'confirm-sign-up' command.`,
    );
  }
};

const confirmSignUpHandler = async (commands) => {
  const [, username, code] = commands;

  try {
```

```
validateUser(username);
validateCode(code);
/**
 * @type {string[]}
 */
const values = getSecondValuesFromEntries(FILE_USER_POOLS);
const clientId = values[0];
validateClient(clientId);
log(`Confirming user.`);
await confirmSignUp({ clientId, username, code });
log(
  `User confirmed. Run 'admin-initiate-auth ${username} <password>' to sign
  in.` ,
);
} catch (err) {
  log(err);
}
};

export { confirmSignUpHandler };

const confirmSignUp = ({ clientId, username, code }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmSignUpCommand({
    ClientId: clientId,
    Username: username,
    ConfirmationCode: code,
  });

  return client.send(command);
};

import qrcode from "qrcode-terminal";
import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { adminInitiateAuth } from "../../actions/admin-initiate-auth.js";
import { associateSoftwareToken } from "../../actions/associate-software-token.js";
import { FILE_USER_POOLS } from "../constants.js";
import { getFirstEntry } from "@aws-doc-sdk-examples/lib/utils/util-csv.js";

const handleMfaSetup = async (session, username) => {
  const { SecretCode, Session } = await associateSoftwareToken(session);
```

```
// Store the Session for use with 'VerifySoftwareToken'.
process.env.SESSION = Session;

console.log(
  "Scan this code in your preferred authenticator app, then run 'verify-
software-token' to finish the setup.",
);
qrcode.generate(
  `otpauth://totp/${username}?secret=${SecretCode}`,
  { small: true },
  console.log,
);
};

const handleSoftwareTokenMfa = (session) => {
  // Store the Session for use with 'AdminRespondToAuthChallenge'.
  process.env.SESSION = session;
};

const validateClient = (id) => {
  if (!id) {
    throw new Error(
      `User pool client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};

const validateId = (id) => {
  if (!id) {
    throw new Error(`User pool id is missing. Did you run 'create-user-pool'?`);
  }
};

const validateUser = (username, password) => {
  if (!(username && password)) {
    throw new Error(
      `Username and password must be provided as arguments to the 'admin-
initiate-auth' command.`,
    );
  }
};

const adminInitiateAuthHandler = async (commands) => {
  const [, username, password] = commands;
```

```
try {
  validateUser(username, password);

  const [userPoolId, clientId] = getFirstEntry(FILE_USER_POOLS);
  validateId(userPoolId);
  validateClient(clientId);

  log("Signing in.");
  const { ChallengeName, Session } = await adminInitiateAuth({
    clientId,
    userPoolId,
    username,
    password,
  });

  if (ChallengeName === "MFA_SETUP") {
    log("MFA setup is required.");
    return handleMfaSetup(Session, username);
  }

  if (ChallengeName === "SOFTWARE_TOKEN_MFA") {
    handleSoftwareTokenMfa(Session);
    log(`Run 'admin-respond-to-auth-challenge ${username} <totp>'`);
  }
} catch (err) {
  log(err);
}

export { adminInitiateAuthHandler };

const adminInitiateAuth = ({ clientId, userPoolId, username, password }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new AdminInitiateAuthCommand({
    ClientId: clientId,
    UserPoolId: userPoolId,
    AuthFlow: AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
    AuthParameters: { USERNAME: username, PASSWORD: password },
  });

  return client.send(command);
};
```

```
import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { adminRespondToAuthChallenge } from "../../actions/admin-respond-to-auth-challenge.js";
import { getFirstEntry } from "@aws-doc-sdk-examples/lib/utils/util-csv.js";
import { FILE_USER_POOLS } from "./constants.js";

const verifyUsername = (username) => {
  if (!username) {
    throw new Error(
      `Username is missing. It must be provided as an argument to the 'admin-respond-to-auth-challenge' command.`
    );
  }
};

const verifyTotp = (totp) => {
  if (!totp) {
    throw new Error(
      `Time-based one-time password (TOTP) is missing. It must be provided as an argument to the 'admin-respond-to-auth-challenge' command.`
    );
  }
};

const storeAccessToken = (token) => {
  process.env.AccessToken = token;
};

const adminRespondToAuthChallengeHandler = async (commands) => {
  const [, username, totp] = commands;

  try {
    verifyUsername(username);
    verifyTotp(totp);

    const [userPoolId, clientId] = getFirstEntry(FILE_USER_POOLS);
    const session = process.env.SESSION;

    const { AuthenticationResult } = await adminRespondToAuthChallenge({
      clientId,
      userPoolId,
      username,
      totp,
    });
  }
};
```

```
        session,
    });

    storeAccessToken(AuthenticationResult.AccessToken);

    log("Successfully authenticated.");
} catch (err) {
    log(err);
}
};

export { adminRespondToAuthChallengeHandler };

const respondToAuthChallenge = ({
    clientId,
    username,
    session,
    userPoolId,
    code,
}) => {
    const client = new CognitoIdentityProviderClient({});

    const command = new RespondToAuthChallengeCommand({
        ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
        ChallengeResponses: {
            SOFTWARE_TOKEN_MFA_CODE: code,
            USERNAME: username,
        },
        ClientId: clientId,
        UserPoolId: userPoolId,
        Session: session,
    });

    return client.send(command);
};

import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { verifySoftwareToken } from "../../../../../actions/verify-software-token.js";

const validateTotp = (totp) => {
    if (!totp) {
        throw new Error(
            `Time-based one-time password (TOTP) must be provided to the 'validate-software-token' command.`
        );
    }
};
```

```
    );
  }
};

const verifySoftwareTokenHandler = async (commands) => {
  const [_ , totp] = commands;

  try {
    validateTotp(totp);

    log("Verifying TOTP.");
    await verifySoftwareToken(totp);
    log("TOTP Verified. Run 'admin-initiate-auth' again to sign-in.");
  } catch (err) {
    console.log(err);
  }
};

export { verifySoftwareTokenHandler };

const verifySoftwareToken = (totp) => {
  const client = new CognitoIdentityProviderClient({});

  // The 'Session' is provided in the response to 'AssociateSoftwareToken'.
  const session = process.env.SESSION;

  if (!session) {
    throw new Error(
      "Missing a valid Session. Did you run 'admin-initiate-auth'?",
    );
  }

  const command = new VerifySoftwareTokenCommand({
    Session: session,
    UserCode: totp,
  });

  return client.send(command);
};
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for JavaScript .
 - [AdminGetUser](#)

- [AdminInitiateAuth](#)
- [AdminRespondToAuthChallenge](#)
- [AssociateSoftwareToken](#)
- [ConfirmDevice](#)
- [ConfirmSignUp](#)
- [InitiateAuth](#)
- [ListUsers](#)
- [ResendConfirmationCode](#)
- [RespondToAuthChallenge](#)
- [SignUp](#)
- [VerifySoftwareToken](#)

Kotlin

SDK pour Kotlin

Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

TIP: To set up the required user pool, run the AWS Cloud Development
Kit (AWS CDK) script provided in this GitHub repo at resources/cdk/
cognito_scenario_user_pool_with_mfa.

This code example performs the following operations:

1. Invokes the signUp method to sign up a user.
2. Invokes the adminGetUser method to get the user's confirmation status.
```

3. Invokes the `ResendConfirmationCode` method if the user requested another code.
 4. Invokes the `confirmSignUp` method.
 5. Invokes the `initiateAuth` to sign in. This results in being prompted to set up TOTP (time-based one-time password). (The response is "ChallengeName": "MFA_SETUP").
 6. Invokes the `AssociateSoftwareToken` method to generate a TOTP MFA private key. This can be used with Google Authenticator.
 7. Invokes the `VerifySoftwareToken` method to verify the TOTP and register for MFA.
 8. Invokes the `AdminInitiateAuth` to sign in again. This results in being prompted to submit a TOTP (Response: "ChallengeName": "SOFTWARE_TOKEN_MFA").
 9. Invokes the `AdminRespondToAuthChallenge` to get back a token.
- */

```
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <clientId> <poolId>
        Where:
            clientId - The app client Id value that you can get from the AWS CDK
script.
            poolId - The pool Id that you can get from the AWS CDK script.
        """

    if (args.size != 2) {
        println(usage)
        exitProcess(1)
    }

    val clientId = args[0]
    val poolId = args[1]

    // Use the console to get data from the user.
    println("**** Enter your use name")
    val in0b = Scanner(System.`in`)
    val userName = in0b.nextLine()
    println(userName)

    println("**** Enter your password")
    val password: String = in0b.nextLine()

    println("**** Enter your email")
    val email = in0b.nextLine()
}
```

```
println("*** Signing up $userName")
signUp(clientId, userName, password, email)

println("*** Getting $userName in the user pool")
getAdminUser(userName, poolId)

println("*** Confirmation code sent to $userName. Would you like to send a
new code? (Yes/No)")
val ans = in0b.nextLine()

if (ans.compareTo("Yes") == 0) {
    println("*** Sending a new confirmation code")
    resendConfirmationCode(clientId, userName)
}
println("*** Enter the confirmation code that was emailed")
val code = in0b.nextLine()
confirmSignUp(clientId, code, userName)

println("*** Rechecking the status of $userName in the user pool")
getAdminUser(userName, poolId)

val authResponse = checkAuthMethod(clientId, userName, password, poolId)
val mySession = authResponse.session
val newSession = getSecretForAppMFA(mySession)
println("*** Enter the 6-digit code displayed in Google Authenticator")
val myCode = in0b.nextLine()

// Verify the TOTP and register for MFA.
verifyTOTP(newSession, myCode)
println("*** Re-enter a 6-digit code displayed in Google Authenticator")
val mfaCode: String = in0b.nextLine()
val authResponse1 = checkAuthMethod(clientId, userName, password, poolId)
val session2 = authResponse1.session
adminRespondToAuthChallenge(userName, clientId, mfaCode, session2)
}

suspend fun checkAuthMethod(clientIdVal: String, userNameVal: String,
passwordVal: String, userPoolIdVal: String): AdminInitiateAuthResponse {
    val authParas = mutableMapOf<String, String>()
    authParas["USERNAME"] = userNameVal
    authParas["PASSWORD"] = passwordVal

    val authRequest = AdminInitiateAuthRequest {
        clientId = clientIdVal
```

```

        userPoolId = userPoolIdVal
        authParameters = authParas
        authFlow = AuthFlowType.AdminUserPasswordAuth
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    val response = identityProviderClient.adminInitiateAuth(authRequest)
    println("Result Challenge is ${response.challengeName}")
    return response
}
}

suspend fun resendConfirmationCode(clientIdVal: String?, userNameVal: String?) {
    val codeRequest = ResendConfirmationCodeRequest {
        clientId = clientIdVal
        username = userNameVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    val response = identityProviderClient.resendConfirmationCode(codeRequest)
    println("Method of delivery is " +
(response.codeDeliveryDetails?.deliveryMedium))
}
}

// Respond to an authentication challenge.
suspend fun adminRespondToAuthChallenge(userName: String, clientIdVal: String?,
mfaCode: String, sessionVal: String?) {
    println("SOFTWARE_TOKEN_MFA challenge is generated")
    val challengeResponsesOb = mutableMapOf<String, String>()
    challengeResponsesOb["USERNAME"] = userName
    challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

    val adminRespondToAuthChallengeRequest = AdminRespondToAuthChallengeRequest {
        challengeName = ChallengeNameType.SoftwareTokenMfa
        clientId = clientIdVal
        challengeResponses = challengeResponsesOb
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->

```

```

        val respondToAuthChallengeResult =
identityProviderClient.adminRespondToAuthChallenge(adminRespondToAuthChallengeRequest)
        println("respondToAuthChallengeResult.getAuthenticationResult()
${respondToAuthChallengeResult.authenticationResult}")
    }
}

// Verify the TOTP and register for MFA.
suspend fun verifyTOTP(sessionVal: String?, codeVal: String?) {
    val tokenRequest = VerifySoftwareTokenRequest {
        userCode = codeVal
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
        val verifyResponse =
identityProviderClient.verifySoftwareToken(tokenRequest)
        println("The status of the token is ${verifyResponse.status}")
    }
}

suspend fun getSecretForAppMFA(sessionVal: String?): String? {
    val softwareTokenRequest = AssociateSoftwareTokenRequest {
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
        val tokenResponse =
identityProviderClient.associateSoftwareToken(softwareTokenRequest)
        val secretCode = tokenResponse.secretCode
        println("Enter this token into Google Authenticator")
        println(secretCode)
        return tokenResponse.session
    }
}

suspend fun confirmSignUp(clientIdVal: String?, codeVal: String?, userNameVal:
String?) {
    val signUpRequest = ConfirmSignUpRequest {
        clientId = clientIdVal
        confirmationCode = codeVal
        username = userNameVal
    }
}

```

```
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    identityProviderClient.confirmSignUp(signUpRequest)
    println("$userNameVal was confirmed")
}
}

suspend fun getAdminUser(userNameVal: String?, poolIdVal: String?) {
    val userRequest = AdminGetUserRequest {
        username = userNameVal
        userPoolId = poolIdVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    val response = identityProviderClient.adminGetUser(userRequest)
    println("User status ${response.userStatus}")
}
}

suspend fun signUp(clientIdVal: String?, userNameVal: String?, passwordVal:
String?, emailVal: String?) {
    val userAttrs = AttributeType {
        name = "email"
        value = emailVal
    }

    val userAttrsList = mutableListOf<AttributeType>()
    userAttrsList.add(userAttrs)
    val signUpRequest = SignUpRequest {
        userAttributes = userAttrsList
        username = userNameVal
        clientId = clientIdVal
        password = passwordVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
{ identityProviderClient ->
    identityProviderClient.signUp(signUpRequest)
    println("User has been signed up")
}
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Kotlin API reference.
 - [AdminGetUser](#)
 - [AdminInitiateAuth](#)
 - [AdminRespondToAuthChallenge](#)
 - [AssociateSoftwareToken](#)
 - [ConfirmDevice](#)
 - [ConfirmSignUp](#)
 - [InitiateAuth](#)
 - [ListUsers](#)
 - [ResendConfirmationCode](#)
 - [RespondToAuthChallenge](#)
 - [SignUp](#)
 - [VerifySoftwareToken](#)

Python

SDK pour Python (Boto3)

Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Créez une classe qui encapsule les fonctions Amazon Cognito utilisées dans le scénario.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
                 client_secret=None):
        """
```

```
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret

def _secret_hash(self, user_name):
    """
    Calculates a secret hash from a user name and a client secret.

    :param user_name: The user name to use when calculating the hash.
    :return: The secret hash.
    """
    key = self.client_secret.encode()
    msg = bytes(user_name + self.client_id, "utf-8")
    secret_hash = base64.b64encode(
        hmac.new(key, msg, digestmod=hashlib.sha256).digest()
    ).decode()
    logger.info("Made secret hash for %s: %s.", user_name, secret_hash)
    return secret_hash

def sign_up_user(self, user_name, password, user_email):
    """
    Signs up a new user with Amazon Cognito. This action prompts Amazon
Cognito
    to send an email to the specified email address. The email contains a
code that
    can be used to confirm the user.

    When the user already exists, the user status is checked to determine
whether
    the user has been confirmed.

    :param user_name: The user name that identifies the new user.
    :param password: The password for the new user.
    :param user_email: The email address for the new user.
    :return: True when the user is already confirmed with Amazon Cognito.
```



```
        Otherwise, false.
    """
    try:
        kwargs = {
            "ClientId": self.client_id,
            "Username": user_name,
            "Password": password,
            "UserAttributes": [{"Name": "email", "Value": user_email}],
        }
        if self.client_secret is not None:
            kwargs["SecretHash"] = self._secret_hash(user_name)
        response = self.cognito_idp_client.sign_up(**kwargs)
        confirmed = response["UserConfirmed"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "UsernameExistsException":
            response = self.cognito_idp_client.admin_get_user(
                UserPoolId=self.user_pool_id, Username=user_name
            )
            logger.warning(
                "User %s exists and is %s.", user_name,
                response["UserStatus"]
            )
            confirmed = response["UserStatus"] == "CONFIRMED"
        else:
            logger.error(
                "Couldn't sign up %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
    return confirmed

def resend_confirmation(self, user_name):
    """
    Prompts Amazon Cognito to resend an email with a new confirmation code.

    :param user_name: The name of the user who will receive the email.
    :return: Delivery information about where the email is sent.
    """
    try:
        kwargs = {"ClientId": self.client_id, "Username": user_name}
        if self.client_secret is not None:
```

```
        kwargs["SecretHash"] = self._secret_hash(user_name)
        response = self.cognito_idp_client.resend_confirmation_code(**kwargs)
        delivery = response["CodeDeliveryDetails"]
    except ClientError as err:
        logger.error(
            "Couldn't resend confirmation to %s. Here's why: %s: %s",
            user_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return delivery

def confirm_user_sign_up(self, user_name, confirmation_code):
    """
    Confirms a previously created user. A user must be confirmed before they
    can sign in to Amazon Cognito.

    :param user_name: The name of the user to confirm.
    :param confirmation_code: The confirmation code sent to the user's
    registered
                           email address.
    :return: True when the confirmation succeeds.
    """
    try:
        kwargs = {
            "ClientId": self.client_id,
            "Username": user_name,
            "ConfirmationCode": confirmation_code,
        }
        if self.client_secret is not None:
            kwargs["SecretHash"] = self._secret_hash(user_name)
        self.cognito_idp_client.confirm_sign_up(**kwargs)
    except ClientError as err:
        logger.error(
            "Couldn't confirm sign up for %s. Here's why: %s: %s",
            user_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
```

```
        return True

def list_users(self):
    """
    Returns a list of the users in the current user pool.

    :return: The list of users.
    """
    try:
        response =
self.cognito_idp_client.list_users(UserPoolId=self.user_pool_id)
        users = response["Users"]
    except ClientError as err:
        logger.error(
            "Couldn't list users for %s. Here's why: %s: %s",
            self.user_pool_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return users

def start_sign_in(self, user_name, password):
    """
    Starts the sign-in process for a user by using administrator credentials.
    This method of signing in is appropriate for code running on a secure
server.

    If the user pool is configured to require MFA and this is the first sign-
in
    for the user, Amazon Cognito returns a challenge response to set up an
MFA application. When this occurs, this function gets an MFA secret from
Amazon Cognito and returns it to the caller.

    :param user_name: The name of the user to sign in.
    :param password: The user's password.
    :return: The result of the sign-in attempt. When sign-in is successful,
this
        returns an access token that can be used to get AWS credentials.
    Otherwise,
        Amazon Cognito returns a challenge to set up an MFA application,
```

```
        or a challenge to enter an MFA code from a registered MFA
application.
    """
    try:
        kwargs = {
            "UserPoolId": self.user_pool_id,
            "ClientId": self.client_id,
            "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
            "AuthParameters": {"USERNAME": user_name, "PASSWORD": password},
        }
        if self.client_secret is not None:
            kwargs["AuthParameters"]["SECRET_HASH"] =
self._secret_hash(user_name)
        response = self.cognito_idp_client.admin_initiate_auth(**kwargs)
        challenge_name = response.get("ChallengeName", None)
        if challenge_name == "MFA_SETUP":
            if (
                "SOFTWARE_TOKEN_MFA"
                in response["ChallengeParameters"]["MFAS_CAN_SETUP"]
            ):
                response.update(self.get_mfa_secret(response["Session"]))
            else:
                raise RuntimeError(
                    "The user pool requires MFA setup, but the user pool is
not "
                    "configured for TOTP MFA. This example requires TOTP
MFA."
                )
        except ClientError as err:
            logger.error(
                "Couldn't start sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response

    def get_mfa_secret(self, session):
        """
```

```
Gets a token that can be used to associate an MFA application with the
user.

:param session: Session information returned from a previous call to
initiate
                authentication.
:return: An MFA token that can be used to set up an MFA application.
"""
try:
    response =
self.cognito_idp_client.associate_software_token(Session=session)
except ClientError as err:
    logger.error(
        "Couldn't get MFA secret. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    response.pop("ResponseMetadata", None)
    return response

def verify_mfa(self, session, user_code):
    """
    Verify a new MFA application that is associated with a user.

    :param session: Session information returned from a previous call to
initiate
                    authentication.
    :param user_code: A code generated by the associated MFA application.
    :return: Status that indicates whether the MFA application is verified.
    """
    try:
        response = self.cognito_idp_client.verify_software_token(
            Session=session, UserCode=user_code
        )
    except ClientError as err:
        logger.error(
            "Couldn't verify MFA. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
```

```
else:
    response.pop("ResponseMetadata", None)
    return response

def respond_to_mfa_challenge(self, user_name, session, mfa_code):
    """
    Responds to a challenge for an MFA code. This completes the second step
of
a two-factor sign-in. When sign-in is successful, it returns an access
token
that can be used to get AWS credentials from Amazon Cognito.

:param user_name: The name of the user who is signing in.
:param session: Session information returned from a previous call to
initiate
authentication.
:param mfa_code: A code generated by the associated MFA application.
:return: The result of the authentication. When successful, this contains
an
access token for the user.
    """
    try:
        kwargs = {
            "UserPoolId": self.user_pool_id,
            "ClientId": self.client_id,
            "ChallengeName": "SOFTWARE_TOKEN_MFA",
            "Session": session,
            "ChallengeResponses": {
                "USERNAME": user_name,
                "SOFTWARE_TOKEN_MFA_CODE": mfa_code,
            },
        }
        if self.client_secret is not None:
            kwargs["ChallengeResponses"]["SECRET_HASH"] = self._secret_hash(
                user_name
            )
        response =
self.cognito_idp_client.admin_respond_to_auth_challenge(**kwargs)
        auth_result = response["AuthenticationResult"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "ExpiredCodeException":
            logger.warning(
```

```

        "Your MFA code has expired or has been used already. You
might have "
        "to wait a few seconds until your app shows you a new code."
    )
    else:
        logger.error(
            "Couldn't respond to mfa challenge for %s. Here's why: %s:
%s",
            user_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return auth_result

def confirm_mfa_device(
    self,
    user_name,
    device_key,
    device_group_key,
    device_password,
    access_token,
    aws_srp,
):
    """
    Confirms an MFA device to be tracked by Amazon Cognito. When a device is
    tracked, its key and password can be used to sign in without requiring a
    new
    MFA code from the MFA application.

    :param user_name: The user that is associated with the device.
    :param device_key: The key of the device, returned by Amazon Cognito.
    :param device_group_key: The group key of the device, returned by Amazon
    Cognito.
    :param device_password: The password that is associated with the device.
    :param access_token: The user's access token.
    :param aws_srp: A class that helps with Secure Remote Password (SRP)
    calculations. The scenario associated with this example
    uses
    the warrant package.
    :return: True when the user must confirm the device. Otherwise, False.
    When

```

```
        False, the device is automatically confirmed and tracked.
    """
    srp_helper = aws_srp.AWSSRP(
        username=user_name,
        password=device_password,
        pool_id="_",
        client_id=self.client_id,
        client_secret=None,
        client=self.cognito_idp_client,
    )
    device_and_pw = f"{device_group_key}{device_key}:{device_password}"
    device_and_pw_hash = aws_srp.hash_sha256(device_and_pw.encode("utf-8"))
    salt = aws_srp.pad_hex(aws_srp.get_random(16))
    x_value = aws_srp.hex_to_long(aws_srp.hex_hash(salt +
device_and_pw_hash))
    verifier = aws_srp.pad_hex(pow(srp_helper.val_g, x_value,
srp_helper.big_n))
    device_secret_verifier_config = {
        "PasswordVerifier": base64.standard_b64encode(
            bytearray.fromhex(verifier)
        ).decode("utf-8"),
        "Salt":
base64.standard_b64encode(bytearray.fromhex(salt)).decode("utf-8"),
    }
    try:
        response = self.cognito_idp_client.confirm_device(
            AccessToken=access_token,
            DeviceKey=device_key,
            DeviceSecretVerifierConfig=device_secret_verifier_config,
        )
        user_confirm = response["UserConfirmationNecessary"]
    except ClientError as err:
        logger.error(
            "Couldn't confirm mfa device %s. Here's why: %s: %s",
            device_key,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return user_confirm

def sign_in_with_tracked_device(
```



```
self,
user_name,
password,
device_key,
device_group_key,
device_password,
aws_srp,
):
    """
    Signs in to Amazon Cognito as a user who has a tracked device. Signing in
    with a tracked device lets a user sign in without entering a new MFA
code.

    Signing in with a tracked device requires that the client respond to the
SRP
    protocol. The scenario associated with this example uses the warrant
package
    to help with SRP calculations.

    For more information on SRP, see https://en.wikipedia.org/wiki/Secure\_Remote\_Password\_protocol.

    :param user_name: The user that is associated with the device.
    :param password: The user's password.
    :param device_key: The key of a tracked device.
    :param device_group_key: The group key of a tracked device.
    :param device_password: The password that is associated with the device.
    :param aws_srp: A class that helps with SRP calculations. The scenario
        associated with this example uses the warrant package.
    :return: The result of the authentication. When successful, this contains
an
        access token for the user.
    """
    try:
        srp_helper = aws_srp.AWSSRP(
            username=user_name,
            password=device_password,
            pool_id="",
            client_id=self.client_id,
            client_secret=None,
            client=self.cognito_idp_client,
        )

        response_init = self.cognito_idp_client.initiate_auth(
```

```
        ClientId=self.client_id,
        AuthFlow="USER_PASSWORD_AUTH",
        AuthParameters={
            "USERNAME": user_name,
            "PASSWORD": password,
            "DEVICE_KEY": device_key,
        },
    )
    if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
        raise RuntimeError(
            f"Expected DEVICE_SRP_AUTH challenge but got {response_init['ChallengeName']}."
        )

    auth_params = srp_helper.get_auth_params()
    auth_params["DEVICE_KEY"] = device_key
    response_auth = self.cognito_idp_client.respond_to_auth_challenge(
        ClientId=self.client_id,
        ChallengeName="DEVICE_SRP_AUTH",
        ChallengeResponses=auth_params,
    )
    if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
        raise RuntimeError(
            f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
            f"{response_init['ChallengeName']}."
        )

    challenge_params = response_auth["ChallengeParameters"]
    challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
    cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
user_name})
    cr["USERNAME"] = user_name
    cr["DEVICE_KEY"] = device_key
    response_verifier =
self.cognito_idp_client.respond_to_auth_challenge(
        ClientId=self.client_id,
        ChallengeName="DEVICE_PASSWORD_VERIFIER",
        ChallengeResponses=cr,
    )
    auth_tokens = response_verifier["AuthenticationResult"]
except ClientError as err:
    logger.error(
        "Couldn't start client sign in for %s. Here's why: %s: %s",
        user_name,
```

```

        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return auth_tokens

```

Créez une classe qui exécute le scénario. Cet exemple enregistre également un appareil MFA devant être suivi par Amazon Cognito et vous montre comment vous connecter à l'aide d'un mot de passe et d'informations provenant de l'appareil suivi. Cela évite d'avoir à saisir un nouveau code MFA.

```

def run_scenario(cognito_idp_client, user_pool_id, client_id):
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Welcome to the Amazon Cognito user signup with MFA demo.")
    print("-" * 88)

    cog_wrapper = CognitoIdentityProviderWrapper(
        cognito_idp_client, user_pool_id, client_id
    )

    user_name = q.ask("Let's sign up a new user. Enter a user name: ",
q.non_empty)
    password = q.ask("Enter a password for the user: ", q.non_empty)
    email = q.ask("Enter a valid email address that you own: ", q.non_empty)
    confirmed = cog_wrapper.sign_up_user(user_name, password, email)
    while not confirmed:
        print(
            f"User {user_name} requires confirmation. Check {email} for "
            f"a verification code."
        )
        confirmation_code = q.ask("Enter the confirmation code from the email: ")
        if not confirmation_code:
            if q.ask("Do you need another confirmation code (y/n)? ",
q.is_yesno):
                delivery = cog_wrapper.resend_confirmation(user_name)
                print(

```

```

        f"Confirmation code sent by {delivery['DeliveryMedium']} "
        f"to {delivery['Destination']})."
    )
    else:
        confirmed = cog_wrapper.confirm_user_sign_up(user_name,
confirmation_code)
        print(f"User {user_name} is confirmed and ready to use.")
        print("-" * 88)

        print("Let's get a list of users in the user pool.")
        q.ask("Press Enter when you're ready.")
        users = cog_wrapper.list_users()
        if users:
            print(f"Found {len(users)} users:")
            pp(users)
        else:
            print("No users found.")
        print("-" * 88)

        print("Let's sign in and get an access token.")
        auth_tokens = None
        challenge = "ADMIN_USER_PASSWORD_AUTH"
        response = {}
        while challenge is not None:
            if challenge == "ADMIN_USER_PASSWORD_AUTH":
                response = cog_wrapper.start_sign_in(user_name, password)
                challenge = response["ChallengeName"]
            elif response["ChallengeName"] == "MFA_SETUP":
                print("First, we need to set up an MFA application.")
                qr_img = qrcode.make(
                    f"otpauth://totp/{user_name}?secret={response['SecretCode']}"
                )
                qr_img.save("qr.png")
                q.ask(
                    "Press Enter to see a QR code on your screen. Scan it into an MFA
"
                    "application, such as Google Authenticator."
                )
                webbrowser.open("qr.png")
                mfa_code = q.ask(
                    "Enter the verification code from your MFA application: ",
q.non_empty
                )
                response = cog_wrapper.verify_mfa(response["Session"], mfa_code)

```

```
print(f"MFA device setup {response['Status']}")
print("Now that an MFA application is set up, let's sign in again.")
print(
    "You might have to wait a few seconds for a new MFA code to
appear in "
    "your MFA application."
)
challenge = "ADMIN_USER_PASSWORD_AUTH"
elif response["ChallengeName"] == "SOFTWARE_TOKEN_MFA":
    auth_tokens = None
    while auth_tokens is None:
        mfa_code = q.ask(
            "Enter a verification code from your MFA application: ",
q.non_empty
        )
        auth_tokens = cog_wrapper.respond_to_mfa_challenge(
            user_name, response["Session"], mfa_code
        )
    print(f"You're signed in as {user_name}.")
    print("Here's your access token:")
    pp(auth_tokens["AccessToken"])
    print("And your device information:")
    pp(auth_tokens["NewDeviceMetadata"])
    challenge = None
else:
    raise Exception(f"Got unexpected challenge
{response['ChallengeName']}")
print("-" * 88)

device_group_key = auth_tokens["NewDeviceMetadata"]["DeviceGroupKey"]
device_key = auth_tokens["NewDeviceMetadata"]["DeviceKey"]
device_password = base64.standard_b64encode(os.urandom(40)).decode("utf-8")

print("Let's confirm your MFA device so you don't have re-enter MFA tokens
for it.")
q.ask("Press Enter when you're ready.")
cog_wrapper.confirm_mfa_device(
    user_name,
    device_key,
    device_group_key,
    device_password,
    auth_tokens["AccessToken"],
    aws_srp,
)
```

```
print(f"Your device {device_key} is confirmed.")
print("-" * 88)

print(
    f"Now let's sign in as {user_name} from your confirmed device
{device_key}.\n"
    f"Because this device is tracked by Amazon Cognito, you won't have to re-
enter an MFA code."
)
q.ask("Press Enter when ready.")
auth_tokens = cog_wrapper.sign_in_with_tracked_device(
    user_name, password, device_key, device_group_key, device_password,
aws_srp
)
print("You're signed in. Your access token is:")
pp(auth_tokens["AccessToken"])
print("-" * 88)

print("Don't forget to delete your user pool when you're done with this
example.")
print("\nThanks for watching!")
print("-" * 88)

def main():
    parser = argparse.ArgumentParser(
        description="Shows how to sign up a new user with Amazon Cognito and
associate "
        "the user with an MFA application for multi-factor authentication."
    )
    parser.add_argument(
        "user_pool_id", help="The ID of the user pool to use for the example."
    )
    parser.add_argument(
        "client_id", help="The ID of the client application to use for the
example."
    )
    args = parser.parse_args()
    try:
        run_scenario(boto3.client("cognito-idp"), args.user_pool_id,
args.client_id)
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

```
if __name__ == "__main__":  
    main()
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Python (Boto3) API Reference.
 - [AdminGetUser](#)
 - [AdminInitiateAuth](#)
 - [AdminRespondToAuthChallenge](#)
 - [AssociateSoftwareToken](#)
 - [ConfirmDevice](#)
 - [ConfirmSignUp](#)
 - [InitiateAuth](#)
 - [ListUsers](#)
 - [ResendConfirmationCode](#)
 - [RespondToAuthChallenge](#)
 - [SignUp](#)
 - [VerifySoftwareToken](#)

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Rédigez des données d'activité personnalisées à l'aide d'une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito à l'aide d'un SDK AWS


L'exemple de code suivant montre comment écrire des données d'activité personnalisées avec une fonction Lambda après l'authentification de l'utilisateur Amazon Cognito.

- Utilisez les fonctions d'administrateur pour ajouter un utilisateur à un groupe d'utilisateurs.
- Configurez un groupe d'utilisateurs pour appeler une fonction Lambda pour le `PostAuthentication` déclencheur.
- Connectez le nouvel utilisateur à Amazon Cognito.

- La fonction Lambda écrit des informations personnalisées dans des CloudWatch journaux et dans une table DynamoDB.
- Obtenez et affichez des données personnalisées à partir de la table DynamoDB, puis nettoyez les ressources.

Go

Kit SDK for Go V2

 Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Exécutez un scénario interactif à une invite de commande.

```
// ActivityLog separates the steps of this scenario into individual functions so
// that
// they are simpler to read and understand.
type ActivityLog struct {
    helper      IScenarioHelper
    questioner  demotools.IQuestioner
    resources   Resources
    cognitoActor *actions.CognitoActions
}

// NewActivityLog constructs a new activity log runner.
func NewActivityLog(sdkConfig aws.Config, questioner demotools.IQuestioner,
    helper IScenarioHelper) ActivityLog {
    scenario := ActivityLog{
        helper:      helper,
        questioner:  questioner,
        resources:   Resources{},
        cognitoActor: &actions.CognitoActions{CognitoClient:
        cognitoidentityprovider.NewFromConfig(sdkConfig)},
    }
    scenario.resources.init(scenario.cognitoActor, questioner)
    return scenario
}
```



```
// AddUserToPool selects a user from the known users table and uses administrator
credentials to add the user to the user pool.
func (runner *ActivityLog) AddUserToPool(userPoolId string, tableName string)
(string, string) {
    log.Println("To facilitate this example, let's add a user to the user pool using
administrator privileges.")
    users, err := runner.helper.GetKnownUsers(tableName)
    if err != nil {
        panic(err)
    }
    user := users.Users[0]
    log.Printf("Adding known user %v to the user pool.\n", user.UserName)
    err = runner.cognitoActor.AdminCreateUser(userPoolId, user.UserName,
user.Email)
    if err != nil {
        panic(err)
    }
    pwSet := false
    password := runner.questioner.AskPassword("\nEnter a password that has at least
eight characters, uppercase, lowercase, numbers and symbols.\n"+
"(the password will not display as you type):", 8)
    for !pwSet {
        log.Printf("\nSetting password for user '%v'.\n", user.UserName)
        err = runner.cognitoActor.AdminSetUserPassword(userPoolId, user.UserName,
password)
        if err != nil {
            var invalidPassword *types.InvalidPasswordException
            if errors.As(err, &invalidPassword) {
                password = runner.questioner.AskPassword("\nEnter another password:", 8)
            } else {
                panic(err)
            }
        } else {
            pwSet = true
        }
    }

    log.Println(strings.Repeat("-", 88))

    return user.UserName, password
}
```

```
// AddActivityLogTrigger adds a Lambda handler as an invocation target for the
PostAuthentication trigger.
func (runner *ActivityLog) AddActivityLogTrigger(userPoolId string,
activityLogArn string) {
log.Println("Let's add a Lambda function to handle the PostAuthentication
trigger from Cognito.\n" +
"This trigger happens after a user is authenticated, and lets your function
take action, such as logging\n" +
"the outcome.")
err := runner.cognitoActor.UpdateTriggers(
userPoolId,
actions.TriggerInfo{Trigger: actions.PostAuthentication, HandlerArn:
aws.String(activityLogArn)})
if err != nil {
panic(err)
}
runner.resources.triggers = append(runner.resources.triggers,
actions.PostAuthentication)
log.Printf("Lambda function %v added to user pool %v to handle
PostAuthentication Cognito trigger.\n",
activityLogArn, userPoolId)

log.Println(strings.Repeat("-", 88))
}

// SignInUser signs in as the specified user.
func (runner *ActivityLog) SignInUser(clientId string, userName string, password
string) {
log.Printf("Now we'll sign in user %v and check the results in the logs and the
DynamoDB table.", userName)
runner.questioner.Ask("Press Enter when you're ready.")
authResult, err := runner.cognitoActor.SignIn(clientId, userName, password)
if err != nil {
panic(err)
}
log.Println("Sign in successful.",
"The PostAuthentication Lambda handler writes custom information to CloudWatch
Logs.")

runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
*authResult.AccessToken)
}
```

```
// GetKnownUserLastLogin gets the login info for a user from the Amazon DynamoDB
table and displays it.
func (runner *ActivityLog) GetKnownUserLastLogin(tableName string, userName
string) {
    log.Println("The PostAuthentication handler also writes login data to the
DynamoDB table.")
    runner.questioner.Ask("Press Enter when you're ready to continue.")
    users, err := runner.helper.GetKnownUsers(tableName)
    if err != nil {
        panic(err)
    }
    for _, user := range users.Users {
        if user.UserName == userName {
            log.Println("The last login info for the user in the known users table is:")
            log.Printf("\t%+v", *user.LastLogin)
        }
    }
    log.Println(strings.Repeat("-", 88))
}

// Run runs the scenario.
func (runner *ActivityLog) Run(stackName string) {
    defer func() {
        if r := recover(); r != nil {
            log.Println("Something went wrong with the demo.")
            runner.resources.Cleanup()
        }
    }()

    log.Println(strings.Repeat("-", 88))
    log.Printf("Welcome\n")

    log.Println(strings.Repeat("-", 88))

    stackOutputs, err := runner.helper.GetStackOutputs(stackName)
    if err != nil {
        panic(err)
    }
    runner.resources.userPoolId = stackOutputs["UserPoolId"]
    runner.helper.PopulateUserTable(stackOutputs["TableName"])
    userName, password := runner.AddUserToPool(stackOutputs["UserPoolId"],
stackOutputs["TableName"])
```

```

runner.AddActivityLogTrigger(stackOutputs["UserPoolId"],
stackOutputs["ActivityLogFunctionArn"])
runner.SignInUser(stackOutputs["UserPoolClientId"], userName, password)
runner.helper.ListRecentLogEvents(stackOutputs["ActivityLogFunction"])
runner.GetKnownUserLastLogin(stackOutputs["TableName"], userName)

runner.resources.Cleanup()

log.Println(strings.Repeat("-", 88))
log.Println("Thanks for watching!")
log.Println(strings.Repeat("-", 88))
}

```

Manipulez le PostAuthentication déclencheur à l'aide d'une fonction Lambda.

```

const TABLE_NAME = "TABLE_NAME"

// LoginInfo defines structured login data that can be marshalled to a DynamoDB
// format.
type LoginInfo struct {
    UserPoolId string `dynamodbav:"UserPoolId"`
    ClientId    string `dynamodbav:"ClientId"`
    Time       string `dynamodbav:"Time"`
}

// UserInfo defines structured user data that can be marshalled to a DynamoDB
// format.
type UserInfo struct {
    UserName    string `dynamodbav:"UserName"`
    UserEmail   string `dynamodbav:"UserEmail"`
    LastLogin   LoginInfo `dynamodbav:"LastLogin"`
}

// GetKey marshals the user email value to a DynamoDB key format.
func (user UserInfo) GetKey() map[string]dynamodbtypes.AttributeValue {
    userEmail, err := attributevalue.Marshal(user.UserEmail)
    if err != nil {
        panic(err)
    }
    return map[string]dynamodbtypes.AttributeValue{"UserEmail": userEmail}
}

```

```
}

type handler struct {
    dynamoClient *dynamodb.Client
}

// HandleRequest handles the PostAuthentication event by writing custom data to
// the logs and
// to an Amazon DynamoDB table.
func (h *handler) HandleRequest(ctx context.Context,
    event events.CognitoEventUserPoolsPostAuthentication)
    (events.CognitoEventUserPoolsPostAuthentication, error) {
    log.Printf("Received post authentication trigger from %v for user '%v'",
        event.TriggerSource, event.UserName)
    tableName := os.Getenv(TABLE_NAME)
    user := UserInfo{
        UserName:    event.UserName,
        UserEmail:   event.Request.UserAttributes["email"],
        LastLogin:   LoginInfo{
            UserPoolId: event.UserPoolID,
            ClientId:   event CallerContext.ClientID,
            Time:      time.Now().Format(time.UnixDate),
        },
    }
    // Write to CloudWatch Logs.
    fmt.Printf("%#v", user)

    // Also write to an external system. This examples uses DynamoDB to demonstrate.
    userMap, err := attributevalue.MarshalMap(user)
    if err != nil {
        log.Printf("Couldn't marshal to DynamoDB map. Here's why: %v\n", err)
    } else if len(userMap) == 0 {
        log.Printf("User info marshaled to an empty map.")
    } else {
        _, err := h.dynamoClient.PutItem(ctx, &dynamodb.PutItemInput{
            Item:      userMap,
            TableName: aws.String(tableName),
        })
        if err != nil {
            log.Printf("Couldn't write to DynamoDB. Here's why: %v\n", err)
        } else {
            log.Printf("Wrote user info to DynamoDB table %v.\n", tableName)
        }
    }
}
```

```

    return event, nil
}

func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        log.Panicln(err)
    }
    h := handler{
        dynamoClient: dynamodb.NewFromConfig(sdkConfig),
    }
    lambda.Start(h.HandleRequest)
}

```

Créez une structure qui exécute des tâches courantes.

```

// IScenarioHelper defines common functions used by the workflows in this
// example.
type IScenarioHelper interface {
    Pause(secs int)
    GetStackOutputs(stackName string) (actions.StackOutputs, error)
    PopulateUserTable(tableName string)
    GetKnownUsers(tableName string) (actions.UserList, error)
    AddKnownUser(tableName string, user actions.User)
    ListRecentLogEvents(functionName string)
}

// ScenarioHelper contains AWS wrapper structs used by the workflows in this
// example.
type ScenarioHelper struct {
    questioner demotools.IQuestioner
    dynamoActor *actions.DynamoActions
    cfnActor    *actions.CloudFormationActions
    cwActor     *actions.CloudWatchLogsActions
    isTestRun  bool
}

// NewScenarioHelper constructs a new scenario helper.

```

```
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
ScenarioHelper {
scenario := ScenarioHelper{
questioner: questioner,
dynamoActor: &actions.DynamoActions{DynamoClient:
dynamodb.NewFromConfig(sdkConfig)},
cfnActor: &actions.CloudFormationActions{CfnClient:
cloudformation.NewFromConfig(sdkConfig)},
cwlActor: &actions.CloudWatchLogsActions{CwlClient:
cloudwatchlogs.NewFromConfig(sdkConfig)},
}
return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
if !helper.isTestRun {
time.Sleep(time.Duration(secs) * time.Second)
}
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
structured format.
func (helper ScenarioHelper) GetStackOutputs(stackName string)
(actions.StackOutputs, error) {
return helper.cfnActor.GetOutputs(stackName), nil
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(tableName string) {
log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
this example.\n", tableName)
err := helper.dynamoActor.PopulateTable(tableName)
if err != nil {
panic(err)
}
}

// GetKnownUsers gets the users from the known users table in a structured
format.
func (helper ScenarioHelper) GetKnownUsers(tableName string) (actions.UserList,
error) {
knownUsers, err := helper.dynamoActor.Scan(tableName)
if err != nil {
```

```
    log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
        tableName, err)
    }
    return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(tableName string, user actions.User) {
    log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
        table...\n",
        user.UserName, user.UserEmail)
    err := helper.dynamoActor.AddUser(tableName, user)
    if err != nil {
        panic(err)
    }
}

// ListRecentLogEvents gets the most recent log stream and events for the
// specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(functionName string) {
    log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
    helper.Pause(10)
    log.Println("Okay, let's check the logs to find what's happened recently with
        your Lambda function.")
    logStream, err := helper.cwlActor.GetLatestLogStream(functionName)
    if err != nil {
        panic(err)
    }
    log.Printf("Getting some recent events from log stream %v\n",
        *logStream.LogStreamName)
    events, err := helper.cwlActor.GetLogEvents(functionName,
        *logStream.LogStreamName, 10)
    if err != nil {
        panic(err)
    }
    for _, event := range events {
        log.Printf("\t%v", *event.Message)
    }
    log.Println(strings.Repeat("-", 88))
}
```


Créez une structure qui englobe les actions Amazon Cognito.

```
type CognitoActions struct {
    CognitoClient *cognitoidentityprovider.Client
}

// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
// trigger.
type Trigger int

const (
    PreSignUp Trigger = iota
    UserMigration
    PostAuthentication
)

type TriggerInfo struct {
    Trigger    Trigger
    HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
// is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(userPoolId string,
    triggers ...TriggerInfo) error {
    output, err := actor.CognitoClient.DescribeUserPool(context.TODO(),
    &cognitoidentityprovider.DescribeUserPoolInput{
        UserPoolId: aws.String(userPoolId),
    })
    if err != nil {
        log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
        userPoolId, err)
        return err
    }
    lambdaConfig := output.UserPool.LambdaConfig
    for _, trigger := range triggers {
        switch trigger.Trigger {
        case PreSignUp:
            lambdaConfig.PreSignUp = trigger.HandlerArn
        }
    }
}
```

```
case UserMigration:
    lambdaConfig.UserMigration = trigger.HandlerArn
case PostAuthentication:
    lambdaConfig.PostAuthentication = trigger.HandlerArn
}
}
_, err = actor.CognitoClient.UpdateUserPool(context.TODO(),
&cognitoidentityprovider.UpdateUserPoolInput{
    UserPoolId:    aws.String(userPoolId),
    LambdaConfig: lambdaConfig,
})
if err != nil {
    log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
}
return err
}

// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(clientId string, userName string, password
string, userEmail string) (bool, error) {
    confirmed := false
    output, err := actor.CognitoClient.SignUp(context.TODO(),
&cognitoidentityprovider.SignUpInput{
        ClientId: aws.String(clientId),
        Password: aws.String(password),
        Username: aws.String(userName),
        UserAttributes: []types.AttributeType{
            {Name: aws.String("email"), Value: aws.String(userEmail)},
        },
    })
    if err != nil {
        var invalidPassword *types.InvalidPasswordException
        if errors.As(err, &invalidPassword) {
            log.Println(*invalidPassword.Message)
        } else {
            log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
        }
    } else {
        confirmed = output.UserConfirmed
    }
    return confirmed, err
}
```

```
// SignIn signs in a user to Amazon Cognito using a username and password
authentication flow.
func (actor CognitoActions) SignIn(clientId string, userName string, password
string) (*types.AuthenticationResultType, error) {
    var authResult *types.AuthenticationResultType
    output, err := actor.CognitoClient.InitiateAuth(context.TODO(),
    &cognitoidentityprovider.InitiateAuthInput{
        AuthFlow:      "USER_PASSWORD_AUTH",
        ClientId:      aws.String(clientId),
        AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
    })
    if err != nil {
        var resetRequired *types.PasswordResetRequiredException
        if errors.As(err, &resetRequired) {
            log.Println(*resetRequired.Message)
        } else {
            log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
        }
    } else {
        authResult = output.AuthenticationResult
    }
    return authResult, err
}

// ForgotPassword starts a password recovery flow for a user. This flow typically
sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(clientId string, userName string)
(*types.CodeDeliveryDetailsType, error) {
    output, err := actor.CognitoClient.ForgotPassword(context.TODO(),
    &cognitoidentityprovider.ForgotPasswordInput{
        ClientId: aws.String(clientId),
        Username: aws.String(userName),
    })
    if err != nil {
        log.Printf("Couldn't start password reset for user '%v'. Here's why: %v\n",
        userName, err)
    }
    return output.CodeDeliveryDetails, err
}
```

```
}

// ConfirmForgotPassword confirms a user with a confirmation code and a new
password.
func (actor CognitoActions) ConfirmForgotPassword(clientId string, code string,
userName string, password string) error {
_, err := actor.CognitoClient.ConfirmForgotPassword(context.TODO(),
&cognitoidentityprovider.ConfirmForgotPasswordInput{
  ClientId:      aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:      aws.String(password),
  Username:      aws.String(userName),
})
if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
    log.Println(*invalidPassword.Message)
  } else {
    log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
}
return err
}

// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(userAccessToken string) error {
_, err := actor.CognitoClient.DeleteUser(context.TODO(),
&cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
})
if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
}
return err
}

// AdminCreateUser uses administrator credentials to add a user to a user pool.
This method leaves the user
```

```
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(userPoolId string, userName string,
userEmail string) error {
    _, err := actor.CognitoClient.AdminCreateUser(context.TODO(),
    &cognitoidentityprovider.AdminCreateUserInput{
        UserPoolId:    aws.String(userPoolId),
        Username:      aws.String(userName),
        MessageAction: types.MessageActionTypeSuppress,
        UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
aws.String(userEmail)}}},
    })
    if err != nil {
        var userExists *types.UsernameExistsException
        if errors.As(err, &userExists) {
            log.Printf("User %v already exists in the user pool.", userName)
            err = nil
        } else {
            log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
        }
    }
    return err
}

// AdminSetUserPassword uses administrator credentials to set a password for a
// user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(userPoolId string, userName
string, password string) error {
    _, err := actor.CognitoClient.AdminSetUserPassword(context.TODO(),
    &cognitoidentityprovider.AdminSetUserPasswordInput{
        Password:    aws.String(password),
        UserPoolId:  aws.String(userPoolId),
        Username:    aws.String(userName),
        Permanent:  true,
    })
    if err != nil {
        var invalidPassword *types.InvalidPasswordException
        if errors.As(err, &invalidPassword) {
            log.Println(*invalidPassword.Message)
        } else {
            log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
err)
        }
    }
}
```

```
    }  
  }  
  return err  
}
```

Créez une structure qui enveloppe les actions DynamoDB.

```
// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)  
actions  
// used in the examples.  
type DynamoActions struct {  
  DynamoClient *dynamodb.Client  
}  
  
// User defines structured user data.  
type User struct {  
  UserName string  
  UserEmail string  
  LastLogin *LoginInfo `dynamodbav:",omitempty"`  
}  
  
// LoginInfo defines structured custom login data.  
type LoginInfo struct {  
  UserPoolId string  
  ClientId string  
  Time string  
}  
  
// UserList defines a list of users.  
type UserList struct {  
  Users []User  
}  
  
// UserNameList returns the usernames contained in a UserList as a list of  
strings.  
func (users *UserList) UserNameList() []string {  
  names := make([]string, len(users.Users))  
  for i := 0; i < len(users.Users); i++ {  
    names[i] = users.Users[i].UserName  
  }  
}
```

```
    return names
}

// PopulateTable adds a set of test users to the table.
func (actor DynamoActions) PopulateTable(tableName string) error {
    var err error
    var item map[string]types.AttributeValue
    var writeReqs []types.WriteRequest
    for i := 1; i < 4; i++ {
        item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
        %v", i), userEmail: fmt.Sprintf("test_email_%v@example.com", i)})
        if err != nil {
            log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
            err)
            return err
        }
        writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
        &types.PutRequest{Item: item}})
    }
    _, err = actor.DynamoClient.BatchWriteItem(context.TODO(),
    &dynamodb.BatchWriteItemInput{
        RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
    })
    if err != nil {
        log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
        tableName, err)
    }
    return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(tableName string) (UserList, error) {
    var userList UserList
    output, err := actor.DynamoClient.Scan(context.TODO(), &dynamodb.ScanInput{
        TableName: aws.String(tableName),
    })
    if err != nil {
        log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
        err)
    } else {
        err = attributevalue.UnmarshalListOfMaps(output.Items, &userList.Users)
        if err != nil {
            log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
        }
    }
}
```

```

    }
    return userList, err
}

// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(tableName string, user User) error {
    userItem, err := attributevalue.MarshalMap(user)
    if err != nil {
        log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
    }
    _, err = actor.DynamoClient.PutItem(context.TODO(), &dynamodb.PutItemInput{
        Item:      userItem,
        TableName: aws.String(tableName),
    })
    if err != nil {
        log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
    }
    return err
}

```

Créez une structure qui englobe les actions CloudWatch Logs.

```

type CloudWatchLogsActions struct {
    CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(functionName string)
(types.LogStream, error) {
    var logStream types.LogStream
    logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
    output, err := actor.CwlClient.DescribeLogStreams(context.TODO(),
&cloudwatchlogs.DescribeLogStreamsInput{
    Descending:  aws.Bool(true),
    Limit:       aws.Int32(1),
    LogGroupName: aws.String(logGroupName),
    OrderBy:    types.OrderByLastEventTime,
})
    if err != nil {

```



```

    log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
logGroupName, err)
} else {
    logStream = output.LogStreams[0]
}
return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
stream.
func (actor CloudWatchLogsActions) GetLogEvents(functionName string,
logStreamName string, eventCount int32) (
[]types.OutputLogEvent, error) {
var events []types.OutputLogEvent
logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
output, err := actor.CwlClient.GetLogEvents(context.TODO(),
&cloudwatchlogs.GetLogEventsInput{
    LogStreamName: aws.String(logStreamName),
    Limit:         aws.Int32(eventCount),
    LogGroupName:  aws.String(logGroupName),
})
if err != nil {
    log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
logStreamName, err)
} else {
    events = output.Events
}
return events, err
}

```

Créez une structure qui englobe les actions. AWS CloudFormation

```

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
    CfnClient *cloudformation.Client
}

```

```
// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
// structured format.
func (actor CloudFormationActions) GetOutputs(stackName string) StackOutputs {
    output, err := actor.CfnClient.DescribeStacks(context.TODO(),
        &cloudformation.DescribeStacksInput{
            StackName: aws.String(stackName),
        })
    if err != nil || len(output.Stacks) == 0 {
        log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
            stackName, err)
    }
    stackOutputs := StackOutputs{}
    for _, out := range output.Stacks[0].Outputs {
        stackOutputs[*out.OutputKey] = *out.OutputValue
    }
    return stackOutputs
}
```

Nettoyez les ressources.

```
// Resources keeps track of AWS resources created during an example and handles
// cleanup when the example finishes.
type Resources struct {
    userPoolId      string
    userAccessTokens []string
    triggers        []actions.Trigger

    cognitoActor *actions.CognitoActions
    questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
    demotools.IQuestioner) {
    resources.userAccessTokens = []string{}
    resources.triggers = []actions.Trigger{}
    resources.cognitoActor = cognitoActor
    resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
```

```
func (resources *Resources) Cleanup() {
    defer func() {
        if r := recover(); r != nil {
            log.Printf("Something went wrong during cleanup.\n%v\n", r)
            log.Println("Use the AWS Management Console to remove any remaining resources\n" +
                "that were created for this scenario.")
        }
    }()

    wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
    resources that were created "+
        "during this demo (y/n)?", "y")
    if wantDelete {
        for _, accessToken := range resources.userAccessTokens {
            err := resources.cognitoActor.DeleteUser(accessToken)
            if err != nil {
                log.Println("Couldn't delete user during cleanup.")
                panic(err)
            }
            log.Println("Deleted user.")
        }
        triggerList := make([]actions.TriggerInfo, len(resources.triggers))
        for i := 0; i < len(resources.triggers); i++ {
            triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
                HandlerArn: nil}
        }
        err := resources.cognitoActor.UpdateTriggers(resources.userPoolId,
            triggerList...)
        if err != nil {
            log.Println("Couldn't update Cognito triggers during cleanup.")
            panic(err)
        }
        log.Println("Removed Cognito triggers from user pool.")
    } else {
        log.Println("Be sure to remove resources when you're done with them to avoid
        unexpected charges!")
    }
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Go .
 - [AdminCreateUser](#)
 - [AdminSetUserPassword](#)
 - [DeleteUser](#)
 - [InitiateAuth](#)
 - [UpdateUserPool](#)

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples de code pour Amazon Cognito Sync à l'aide de kits SDK AWS

Les exemples de code suivants montrent comment utiliser Amazon Cognito Sync avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés et dans des exemples interservices.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- [Actions pour Amazon Cognito Sync à l'aide des SDK AWS](#)
 - [Utilisation ListIdentityPoolUsage avec un AWS SDK ou une CLI](#)

Actions pour Amazon Cognito Sync à l'aide des SDK AWS

Les exemples de code suivants montrent comment effectuer des actions Amazon Cognito Sync individuelles avec AWS des SDK. Ces extraits appellent l'API Synchronisation Amazon Cognito et sont des extraits de code de programmes de plus grande envergure qui doivent être exécutés en contexte. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez [Amazon Cognito Sync API Reference](#) (Référence de l'API Synchronisation Amazon Cognito).

Exemples

- [Utilisation ListIdentityPoolUsage avec un AWS SDK ou une CLI](#)

Utilisation `ListIdentityPoolUsage` avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser `ListIdentityPoolUsage`.

Rust

SDK pour Rust

Note

Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
async fn show_pools(client: &Client) -> Result<(), Error> {
    let response = client
        .list_identity_pool_usage()
        .max_results(10)
        .send()
        .await?;

    let pools = response.identity_pool_usages();
    println!("Identity pools:");
}
```

```
for pool in pools {
    println!(
        " Identity pool ID:   {}",
        pool.identity_pool_id().unwrap_or_default()
    );
    println!(
        " Data storage:         {}",
        pool.data_storage().unwrap_or_default()
    );
    println!(
        " Sync sessions count: {}",
        pool.sync_sessions_count().unwrap_or_default()
    );
    println!(
        " Last modified:         {}",
        pool.last_modified_date().unwrap().to_chrono_utc()?
    );
    println!();
}

println!("Next token: {}", response.next_token().unwrap_or_default());

Ok(())
}
```

- Pour plus de détails sur l'API, voir [ListIdentityPoolUsage](#) la section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Bonnes pratiques pour les applications multilocataires

Les groupes d'utilisateurs Amazon Cognito fonctionnent avec des applications multi-locataires qui génèrent un volume de demandes qui doit rester dans les limites des quotas Amazon Cognito. Pour augmenter cette capacité lorsque votre clientèle augmente, vous pouvez [acheter des capacités de quota supplémentaires](#).

Note

Les [quotas](#) Amazon Cognito sont appliqués au fur et à mesure. Compte AWS Région AWS Ces quotas sont partagés entre tous les locataires au sein de votre application. Passez en revue les quotas du service Amazon Cognito et assurez-vous qu'ils correspondent au volume attendu et au nombre de locataires attendus dans votre application.

Cette section décrit les méthodes que vous pouvez mettre en œuvre pour séparer les locataires entre les ressources Amazon Cognito au sein d'une même région et. Compte AWS Vous pouvez également répartir vos locataires sur plusieurs Compte AWS régions et attribuer à chacun son propre quota. Les autres avantages de la mutualisation multirégionale incluent le niveau d'isolation le plus élevé possible, le temps de transit réseau le plus court pour les utilisateurs répartis dans le monde entier et le respect des modèles de distribution existants dans votre organisation.

La mutualisation dans une seule région peut également présenter des avantages pour vos clients et vos administrateurs.

La liste suivante présente certains des avantages de la mutualisation avec des ressources partagées.

Avantages de la location multiple

Répertoire d'utilisateurs commun

La mutualisation prend en charge les modèles dans lesquels les clients ont des comptes dans plusieurs applications. Vous pouvez [lier les identités de fournisseurs tiers](#) dans un profil de groupe d'utilisateurs unique et cohérent. Dans les cas où les profils utilisateur sont propres à leur locataire, toute stratégie de mutualisation avec un pool d'utilisateurs unique comporte un point d'entrée unique dans l'administration des utilisateurs.

Sécurité commune

Dans un groupe d'utilisateurs partagé, vous pouvez créer une norme de sécurité unique et appliquer les mêmes normes de [sécurité avancée](#), [d'authentification multifactorielle](#) (MFA) [AWS WAF](#) et les mêmes normes à tous les locataires. Comme une ACL AWS WAF Web doit se trouver dans la même Région AWS zone que la ressource à laquelle vous l'associez, la mutualisation offre un accès partagé à une ressource complexe. Lorsque vous souhaitez conserver une configuration de sécurité cohérente dans les applications Amazon Cognito multirégionales, vous devez appliquer des normes opérationnelles qui répliquent votre configuration entre les ressources.

Personnalisation commune

Vous pouvez personnaliser les groupes d'utilisateurs et les groupes d'identités avec AWS Lambda. La configuration des [déclencheurs Lambda](#) dans les groupes d'utilisateurs et des événements [Amazon Cognito](#) dans les groupes d'identités peut devenir complexe. Les fonctions Lambda doivent être identiques à celles Région AWS de votre groupe d'utilisateurs ou de votre groupe d'identités. Les fonctions Lambda partagées peuvent appliquer les normes relatives aux flux d'authentification personnalisés, à la migration des utilisateurs, à la génération de jetons et à d'autres fonctions au sein d'une région.

Messagerie courante

Amazon Simple Notification Service (Amazon SNS) nécessite une configuration supplémentaire dans une région avant que vous puissiez [envoyer des SMS](#) à vos utilisateurs. Vous pouvez envoyer [des e-mails avec des](#) identités vérifiées par Amazon Simple Email Service (Amazon SES) et des domaines contenus dans une région.

Avec le multitenant, vous pouvez partager cette configuration et les frais de maintenance entre tous vos locataires. Amazon SNS et Amazon SES ne étant pas tous disponibles Régions AWS, la répartition de vos ressources entre les régions nécessite une attention particulière.

Lorsque vous utilisez [des fournisseurs de messagerie personnalisés](#), vous bénéficiez de la personnalisation commune d'une seule fonction Lambda pour gérer la livraison de vos messages.

L'[interface utilisateur hébergée](#) définit un cookie de session dans le navigateur afin qu'il reconnaisse un utilisateur qui s'est déjà authentifié. Lorsque vous authentifiez des utilisateurs locaux dans un groupe d'utilisateurs, leur cookie de session les authentifie pour tous les clients d'applications du même groupe d'utilisateurs. Un utilisateur local existe exclusivement dans l'annuaire de votre groupe

d'utilisateurs sans fédération via un fournisseur d'identité externe. Le cookie de session est valide pendant une heure. Vous ne pouvez pas modifier la durée du cookie de session.

Il existe deux méthodes pour empêcher la connexion entre les clients de l'application à l'aide d'un cookie de session d'interface utilisateur hébergé.

- Séparez vos utilisateurs en groupes d'utilisateurs par locataire.
- Remplacez la connexion à l'interface utilisateur hébergée par la connexion à l'API des groupes d'utilisateurs Amazon Cognito.

Rubriques

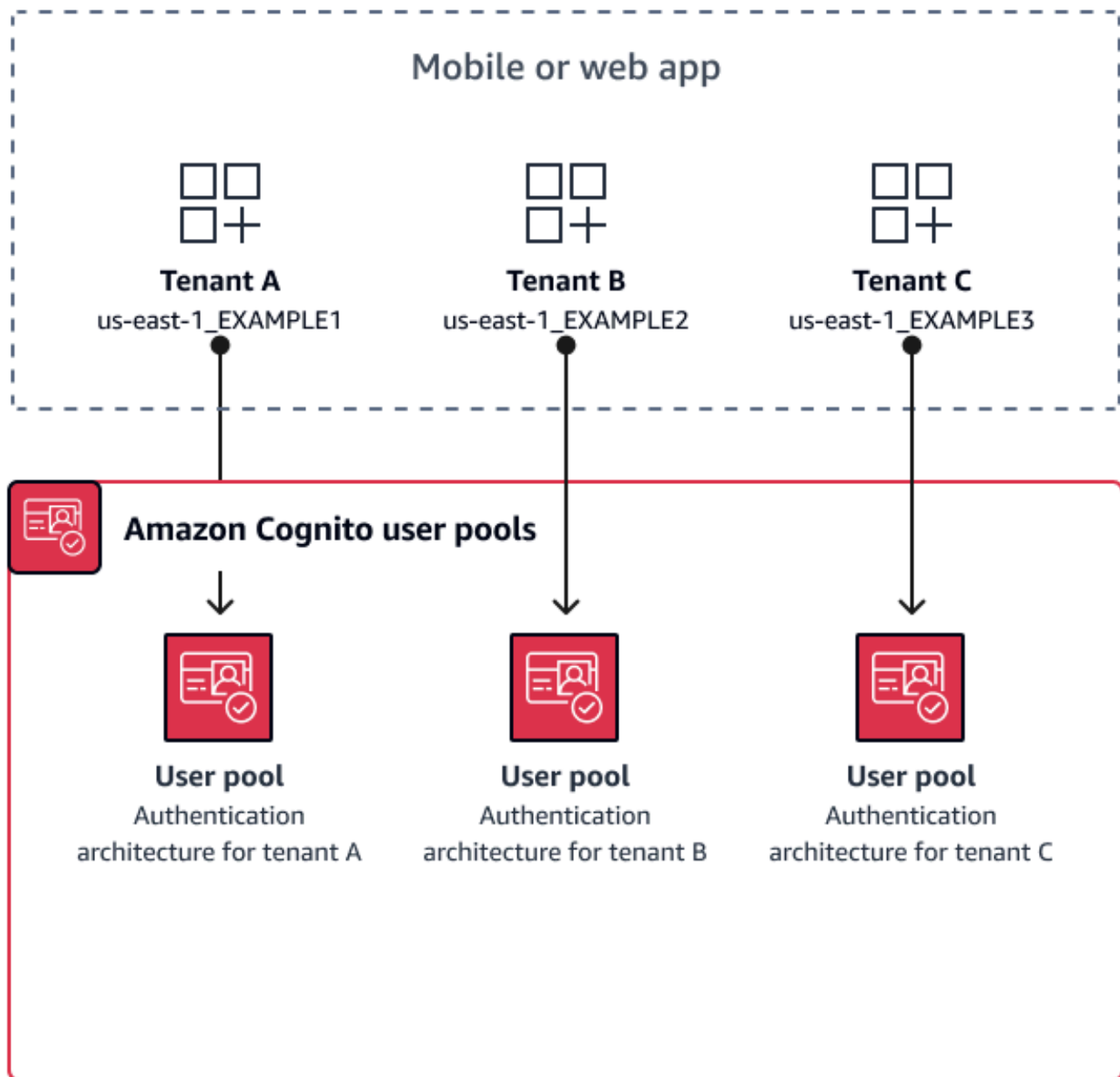
- [Meilleures pratiques en matière de mutualisation des pools d'utilisateurs](#)
- [Meilleures pratiques en matière de mutualisation entre applications et clients](#)
- [Meilleures pratiques en matière de mutualisation des groupes d'utilisateurs](#)
- [Bonnes pratiques en matière de mutualisation d'attributs personnalisés](#)
- [Recommandations en matière de sécurité multilocataire](#)

Meilleures pratiques en matière de mutualisation des pools d'utilisateurs

Créez un groupe d'utilisateurs pour chaque locataire de votre application. Cette approche offre un isolement maximal pour chaque locataire. Vous pouvez implémenter différentes configurations pour chaque locataire. L'isolation des locataires par groupe d'utilisateurs vous donne de la flexibilité dans le user-to-tenant mappage. Vous pouvez créer plusieurs profils pour un même utilisateur. Cependant, chaque utilisateur doit s'inscrire individuellement pour chaque locataire auquel il a accès.

Grâce à cette approche, vous pouvez configurer une interface utilisateur hébergée pour chaque locataire indépendamment et rediriger les utilisateurs vers l'instance de votre application spécifique au locataire. Vous pouvez également utiliser cette approche pour intégrer des services principaux tels qu'[Amazon API Gateway](#).

Le schéma suivant montre chaque locataire avec un pool d'utilisateurs dédié.



Quand mettre en œuvre la mutualisation du pool d'utilisateurs

Lorsque l'isolation et la personnalisation sont vos principales préoccupations. La relation entre les utilisateurs et les locataires peut être complexe dans une architecture comportant plusieurs groupes d'utilisateurs. Prenons un exemple où vous avez deux locataires éducatifs. Le même utilisateur peut être un étudiant à accès limité dans une application et un enseignant disposant d'un niveau élevé d'autorisations dans une autre. Il se peut que vous ayez besoin de l'authentification MFA dans une

application mais pas dans une autre, ou que vous ayez une politique de mot de passe différente. Étant donné que les utilisateurs locaux peuvent se connecter à plusieurs clients d'applications dans des groupes d'utilisateurs grâce à l'interface utilisateur hébergée, la mutualisation des groupes d'utilisateurs est également idéale lorsque vous souhaitez que plusieurs de vos locataires se connectent via l'interface utilisateur hébergée.

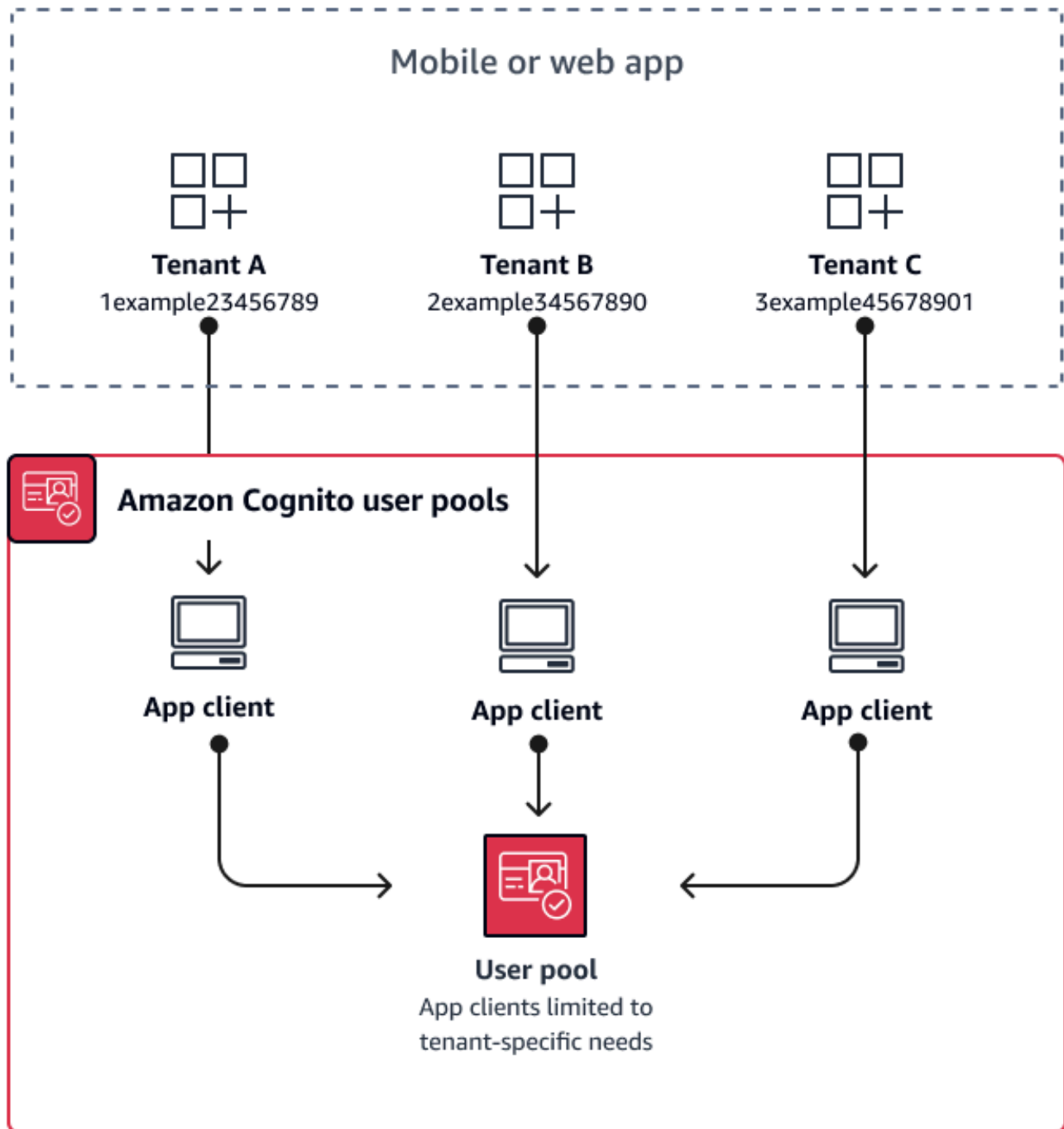
Niveau d'effort

L'effort de développement et d'exploitation lié à cette approche est élevé. Pour garantir des résultats cohérents et prévisibles pour votre famille d'applications, vous devez intégrer les ressources Amazon Cognito à vos outils d'automatisation et conserver vos bases de référence à mesure que votre architecture d'authentification devient de plus en plus complexe. Lorsque vous souhaitez créer un point de départ unique pour vos applications, vous devez créer les éléments de l'interface utilisateur (UI) pour capturer la décision initiale qui oriente les utilisateurs vers la bonne ressource.

Meilleures pratiques en matière de mutualisation entre applications et clients

Créez un [client d'application](#) pour chaque locataire de votre application. Grâce à la mutualisation entre applications et clients, vous pouvez affecter n'importe quel utilisateur à des clients d'applications liés à des locataires et conserver un profil utilisateur unique. Comme vous pouvez attribuer l'un ou l'ensemble des [fournisseurs d'identité \(IdPs\)](#) de votre groupe d'utilisateurs à un client d'application, celui-ci peut autoriser la connexion avec un IdP spécifique au locataire. Lorsque les utilisateurs existent dans plusieurs locataires, vous pouvez associer leurs profils à plusieurs IdPs pour une expérience utilisateur cohérente.

Le schéma suivant montre chaque locataire disposant d'un client d'application dédié dans un pool d'utilisateurs partagé.



Quand mettre en œuvre la mutualisation entre l'application et le client

Quand vous pouvez choisir une configuration universelle pour les paramètres au niveau du pool d'utilisateurs, tels que les déclencheurs Lambda, la politique de mot de passe, le contenu et les

méthodes de livraison des e-mails et des SMS. Étant donné que les utilisateurs d'un groupe d'utilisateurs partagé peuvent se connecter à n'importe quel client d'application, la mutualisation entre clients d'applications est idéale pour se connecter avec ou via l'API des groupes d'utilisateurs application-specific IdPs Amazon Cognito. La mutualisation client-application convient également aux one-to-many environnements dans lesquels vous souhaitez permettre aux utilisateurs de passer d'une application à une autre.

Niveau d'effort

La mutualisation entre l'application et le client nécessite un effort modéré. L'un des principaux défis de la mutualisation entre applications et clients est la possibilité pour les locataires de présenter un cookie d'interface utilisateur hébergé et de passer d'une application à l'autre. Dans une architecture mutualisée client-application, évitez de vous connecter à l'interface utilisateur hébergée lorsque l'isolation est nécessaire. Vous pouvez distribuer votre application mobile ou des liens vers votre application Web avec la logique du client d'application intégrée, ou vous pouvez créer des éléments d'interface utilisateur initiaux qui déterminent la location des utilisateurs. Le niveau d'effort est moindre car vous n'avez pas besoin de standardiser et de maintenir la configuration entre plusieurs groupes d'utilisateurs et groupes d'identités.

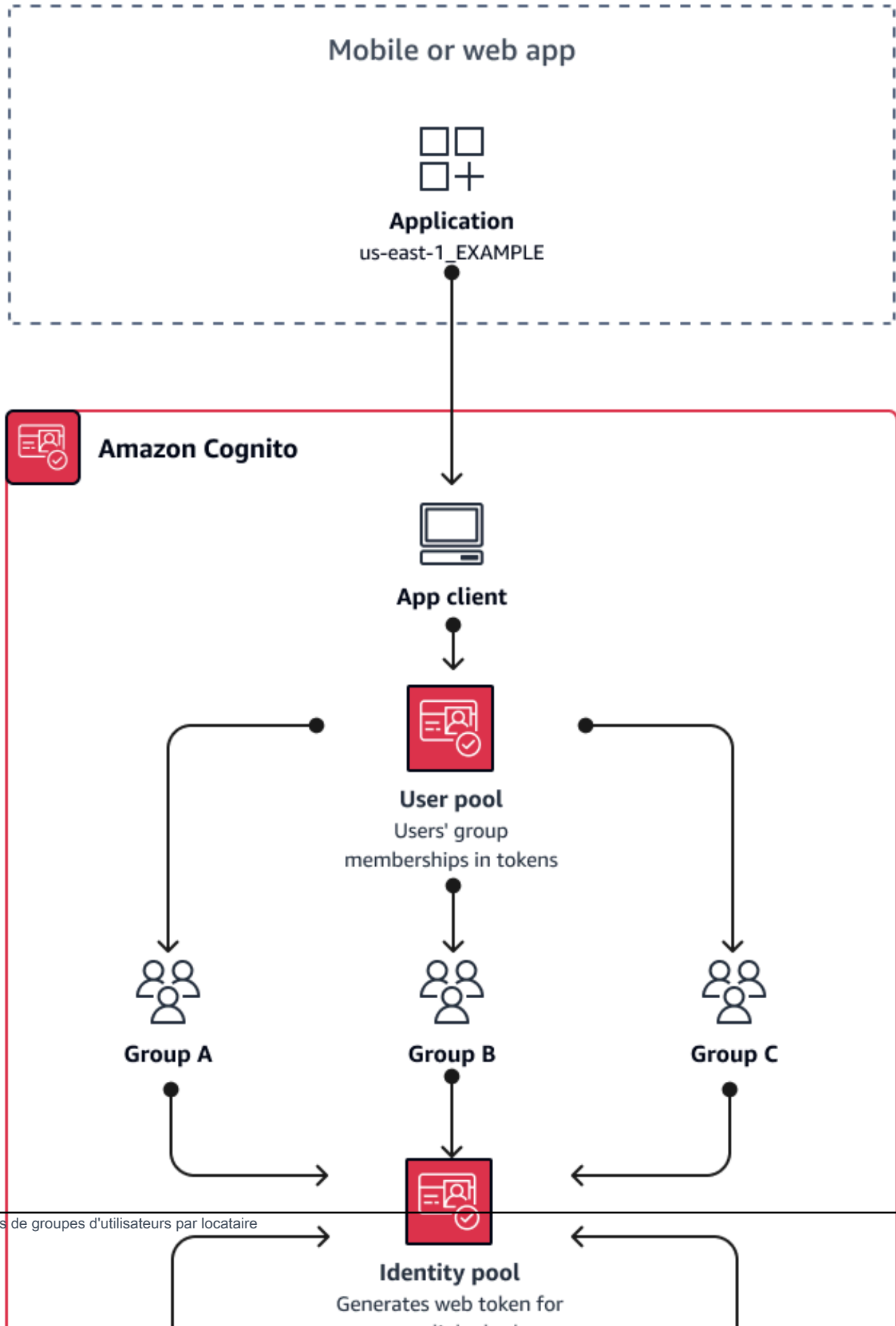
Meilleures pratiques en matière de mutualisation des groupes d'utilisateurs

La mutualisation basée sur les groupes fonctionne mieux lorsque votre architecture nécessite des groupes d'utilisateurs Amazon Cognito dotés de groupes d'identités.

L'[ID du groupe d'utilisateurs et les jetons d'accès](#) contiennent une `cognito:groups` réclamation. De plus, les jetons d'identification contiennent `cognito:roles` et `cognito:preferred_role` revendiquent. Lorsque le principal résultat de l'authentification dans votre application est des AWS informations d'identification temporaires provenant d'un pool d'identités, les appartenances aux groupes de vos utilisateurs peuvent déterminer le [rôle IAM](#) et les autorisations qu'ils reçoivent.

Prenons l'exemple de trois locataires qui stockent chacun les actifs de l'application dans leur propre compartiment Amazon S3. Affectez les utilisateurs de chaque locataire à un groupe associé, configurez un rôle préféré pour le groupe et accordez à ce rôle un accès en lecture à leur bucket.

Le schéma suivant montre les locataires partageant un client d'application et un groupe d'utilisateurs, avec des groupes dédiés dans le groupe d'utilisateurs qui déterminent leur éligibilité à un rôle IAM.



Quand mettre en œuvre la mutualisation collective

Lorsque l'accès aux AWS ressources est votre principale préoccupation. Les groupes des groupes d'utilisateurs Amazon Cognito sont un mécanisme de contrôle d'accès basé sur les rôles (RBAC). Vous pouvez configurer de nombreux groupes dans un groupe d'utilisateurs et prendre des décisions RBAC complexes avec une priorité de groupe. Les pools d'identités peuvent attribuer des informations d'identification au rôle ayant la priorité la plus élevée, à n'importe quel rôle revendiqué par le groupe ou à partir d'autres revendications contenues dans les jetons d'un utilisateur.

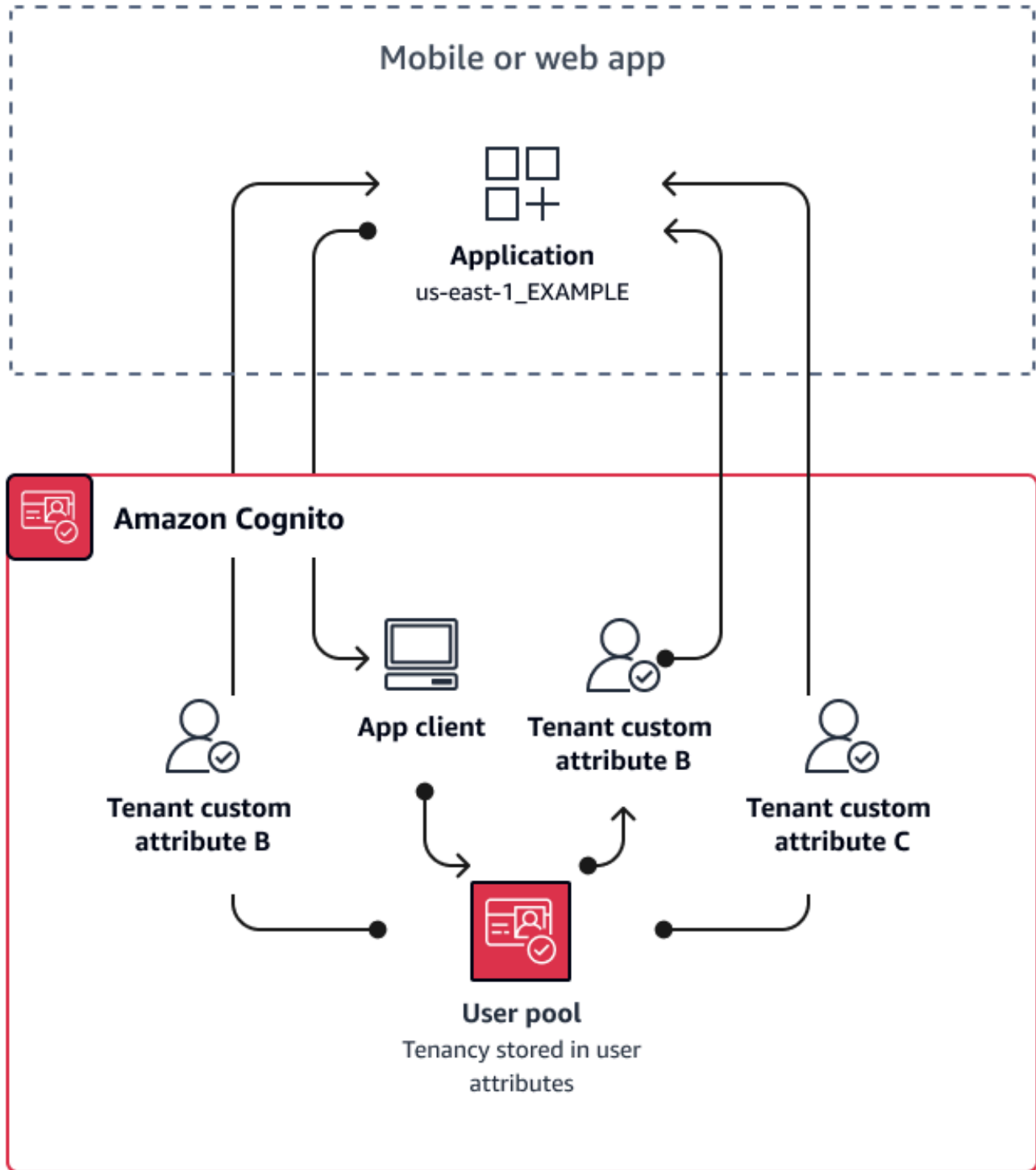
Niveau d'effort

Le niveau d'effort pour maintenir la multilocation avec la seule adhésion à un groupe est faible. Toutefois, pour étendre le rôle des groupes de groupes d'utilisateurs au-delà de la capacité intégrée de sélection des rôles IAM, vous devez créer une logique d'application qui traite l'appartenance aux groupes dans les jetons des utilisateurs et déterminer ce qu'il faut faire dans le client. Vous pouvez intégrer Amazon Verified Permissions à vos applications pour prendre des décisions d'autorisation côté client. Les identifiants de groupe ne sont actuellement pas traités dans les opérations de [IsAuthorizedWithToken](#) API Verified Permissions, mais vous pouvez [développer un code personnalisé](#) qui analyse le contenu des jetons, y compris les demandes d'adhésion à un groupe.

Bonnes pratiques en matière de mutualisation d'attributs personnalisés

Amazon Cognito prend en charge les [attributs personnalisés](#) avec des noms que vous choisissez. L'un des scénarios dans lesquels les attributs personnalisés sont utiles est lorsqu'ils distinguent la location des utilisateurs dans un groupe d'utilisateurs partagé. Lorsque vous attribuez aux utilisateurs une valeur pour un attribut tel que `custom:tenantID`, votre application peut attribuer l'accès aux ressources spécifiques au locataire en conséquence. Un attribut personnalisé qui définit un ID de locataire doit être immuable ou en lecture seule pour le client de l'application.

Le schéma suivant montre les locataires partageant un client d'application et un groupe d'utilisateurs, avec des attributs personnalisés dans le groupe d'utilisateurs qui indiquent le locataire auquel ils appartiennent.



Lorsque des attributs personnalisés déterminent la location, vous pouvez distribuer une seule application ou une seule URL de connexion. Une fois que votre utilisateur s'est connecté, votre

application peut traiter la custom:tenantID réclamation, déterminer les actifs à charger, l'image de marque à appliquer et les fonctionnalités à afficher. Pour prendre des décisions avancées en matière de contrôle d'accès à partir des attributs utilisateur, configurez votre groupe d'utilisateurs en tant que fournisseur d'identité dans Amazon Verified Permissions et générez des décisions d'accès à partir du contenu des identifiants ou des jetons d'accès.

Quand mettre en œuvre la mutualisation des attributs personnalisés

Lorsque la location se fait au niveau de la surface. Un attribut tenant peut contribuer aux résultats de marque et de mise en page. Lorsque vous souhaitez obtenir une isolation significative entre les locataires, les attributs personnalisés ne sont pas le meilleur choix. Toute différence entre les locataires qui doit être configurée au niveau du pool d'utilisateurs ou du client de l'application, comme le MFA ou l'image de marque de l'interface utilisateur hébergée, nécessite que vous créiez des distinctions entre les locataires d'une manière que les attributs personnalisés ne proposent pas. Avec les pools d'identités, vous pouvez même choisir le rôle IAM parmi vos utilisateurs à partir de l'attribut personnalisé indiqué dans leur jeton d'identification.

Niveau d'effort

Étant donné que la mutualisation des attributs personnalisés transfère le devoir de prendre les décisions d'autorisation basées sur les locataires sur votre application, le niveau d'effort a tendance à être élevé. Si vous êtes déjà familiarisé avec une configuration client qui analyse les demandes OIDC ou avec Amazon Verified Permissions, cette approche peut nécessiter le moins d'efforts.

Recommandations en matière de sécurité multilocataire

Pour vous aider à sécuriser votre application, nous vous recommandons ce qui suit :

- Validez la location dans votre application avec les autorisations vérifiées par Amazon. Élaborez des politiques qui examinent les droits relatifs au groupe d'utilisateurs, aux clients d'applications, aux groupes ou aux attributs personnalisés avant d'autoriser la demande d'un utilisateur dans votre application. AWS a créé des [sources d'identité](#) Verified Permissions en pensant aux groupes d'utilisateurs Amazon Cognito. Verified Permissions propose [des instructions supplémentaires](#) pour la gestion de l'hébergement mutualisé.
- Utilisez uniquement une adresse e-mail vérifiée pour autoriser l'accès utilisateur à un locataire sur la base d'une correspondance de domaine. Ne faites pas confiance aux adresses e-mail et aux numéros de téléphone à moins que votre application ne les ait vérifiés ou que le fournisseur d'identité externe n'ait fourni une preuve de vérification. Pour plus d'informations sur la définition de ces autorisations, consultez [Attribuer des autorisations et des périmètres](#).

- Utilisez des attributs personnalisés immuables ou en lecture seule pour les attributs de profil utilisateur qui identifient les locataires. Vous ne pouvez définir la valeur des attributs immuables que lorsque vous créez un utilisateur ou lorsqu'un utilisateur s'inscrit dans votre groupe d'utilisateurs. De plus, accordez aux clients d'application un accès en lecture seule à ces attributs.
- Utilisez un mappage 1:1 entre l'IdP externe d'un locataire et le client d'application pour empêcher tout accès non autorisé entre locataires. Un utilisateur authentifié par un fournisseur d'identité externe et doté d'un cookie de session Amazon Cognito valide peut accéder aux applications d'autres locataires qui font confiance au même fournisseur d'identité.
- Quand vous implémentez la logique d'autorisation et de correspondance de locataire dans votre application, limitez les utilisateurs afin qu'ils ne puissent pas modifier les critères utilisés pour autoriser l'accès des utilisateurs aux locataires. De plus, si un fournisseur d'identité externe est utilisé pour la fédération, limitez les administrateurs du fournisseur d'identité du locataire afin qu'ils ne puissent pas modifier l'accès utilisateur.

Scénarios Amazon Cognito courants

Cette rubrique décrit six scénarios courants d'utilisation d'Amazon Cognito.

Les deux principaux composants d'Amazon Cognito sont les groupes d'utilisateurs et les groupes d'identités. Les groupes d'utilisateurs sont des répertoires d'utilisateurs qui fournissent des options d'inscription et de connexion pour les utilisateurs de votre application web ou mobile. Les pools d'identités fournissent des AWS informations d'identification temporaires pour permettre à vos utilisateurs d'accéder à d'autres Services AWS.

Un groupe d'utilisateurs est un répertoire d'utilisateurs dans Amazon Cognito. Les utilisateurs de votre application peuvent soit se connecter directement via un groupe d'utilisateurs, soit fédérer via un fournisseur d'identité (IdP) tiers. Le groupe d'utilisateurs gère les frais généraux liés à la gestion des jetons renvoyés lors de la connexion aux réseaux sociaux via Facebook, Google, Amazon et Apple, et depuis OpenID Connect (OIDC) et SAML. IdPs Que vos utilisateurs se connectent directement ou via un tiers, tous les membres du groupe d'utilisateurs ont un profil de répertoire auquel vous pouvez accéder par le biais d'un kit SDK.

Avec un pool d'identités, vos utilisateurs peuvent obtenir des AWS informations d'identification temporaires pour accéder à AWS des services tels qu'Amazon S3 et DynamoDB. Les pools d'identités prennent en charge les utilisateurs invités anonymes, ainsi que la fédération par le biais de tiers IdPs.

Rubriques

- [S'authentifier avec un groupe d'utilisateurs](#)
- [Accès à vos ressources côté serveur avec un groupe d'utilisateurs](#)
- [Accéder à des ressources avec API Gateway et Lambda avec un groupe d'utilisateurs](#)
- [AWS Services d'accès avec un pool d'utilisateurs et un pool d'identités](#)
- [S'authentifier avec un tiers et accéder aux services AWS avec un groupe d'identités](#)
- [Accédez aux AWS AppSync ressources avec Amazon Cognito](#)

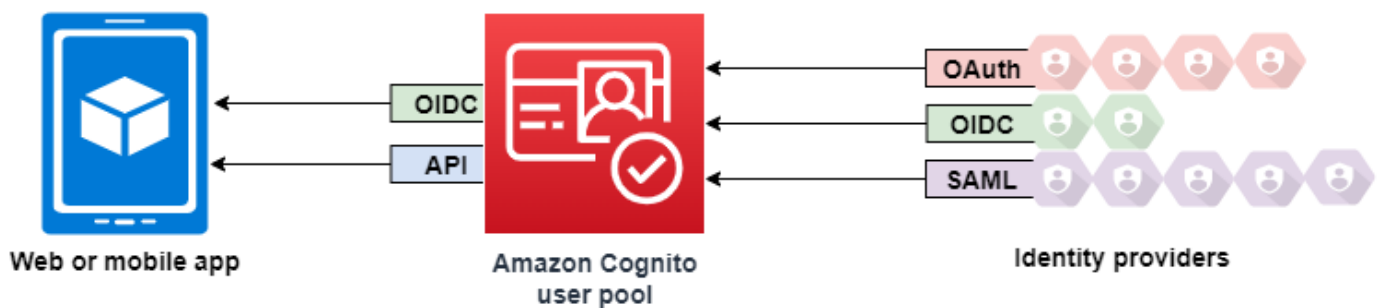
S'authentifier avec un groupe d'utilisateurs

Vous pouvez permettre à vos utilisateurs de s'authentifier à l'aide d'un groupe d'utilisateurs. Les utilisateurs de votre application peuvent soit se connecter directement via un groupe d'utilisateurs,

soit fédérer via un fournisseur d'identité (IdP) tiers. Le groupe d'utilisateurs gère les frais généraux liés à la gestion des jetons renvoyés lors de la connexion aux réseaux sociaux via Facebook, Google, Amazon et Apple, et depuis OpenID Connect (OIDC) et SAML. IdPs

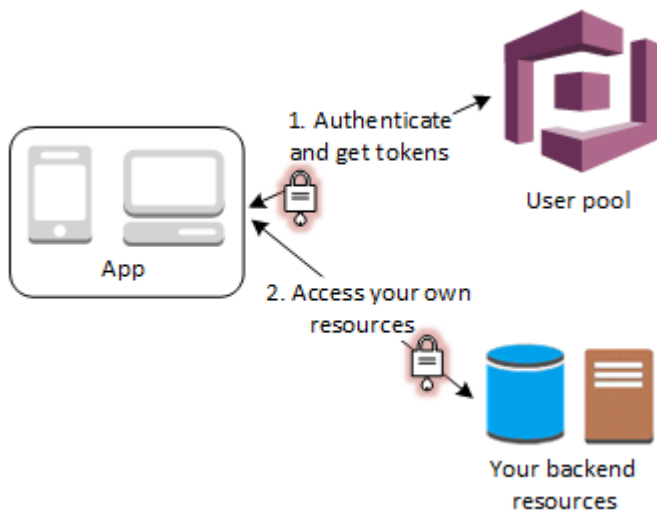
Après une authentification réussie, votre application mobile ou web recevra d'Amazon Cognito des jetons de groupe d'utilisateurs. Vous pouvez utiliser ces jetons pour récupérer les AWS informations d'identification qui permettent à votre application d'accéder à d'autres AWS services, ou vous pouvez choisir de les utiliser pour contrôler l'accès à vos ressources côté serveur ou à Amazon API Gateway.

Pour plus d'informations, consultez [Flux d'authentification de groupe d'utilisateurs](#) et [Utilisation des jetons avec des groupes d'utilisateurs](#).



Accès à vos ressources côté serveur avec un groupe d'utilisateurs

Après une connexion de groupe d'utilisateurs réussie, votre application mobile ou web recevra des jetons de groupe d'utilisateurs d'Amazon Cognito. Vous pouvez utiliser ces jetons pour contrôler l'accès à vos ressources côté serveur. Vous pouvez également créer des ensembles de groupes d'utilisateurs afin de gérer leurs autorisations et de représenter différents types d'utilisateurs. Pour plus d'informations sur l'utilisation de groupes pour contrôler l'accès à vos ressources, consultez [Ajout de groupes à un groupe d'utilisateurs](#).



Une fois que vous avez configuré un domaine pour votre groupe d'utilisateurs, Amazon Cognito met en service une interface utilisateur web hébergée qui vous permet d'ajouter des pages d'inscription et de connexion à votre application. À l'aide de cette base OAuth 2.0, vous pouvez créer votre propre serveur de ressources pour permettre à vos utilisateurs d'accéder à des ressources protégées. Pour plus d'informations, consultez [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#).

Pour plus d'informations sur l'authentification d'un groupe d'utilisateurs, consultez [Flux d'authentification de groupe d'utilisateurs](#) et [Utilisation des jetons avec des groupes d'utilisateurs](#).

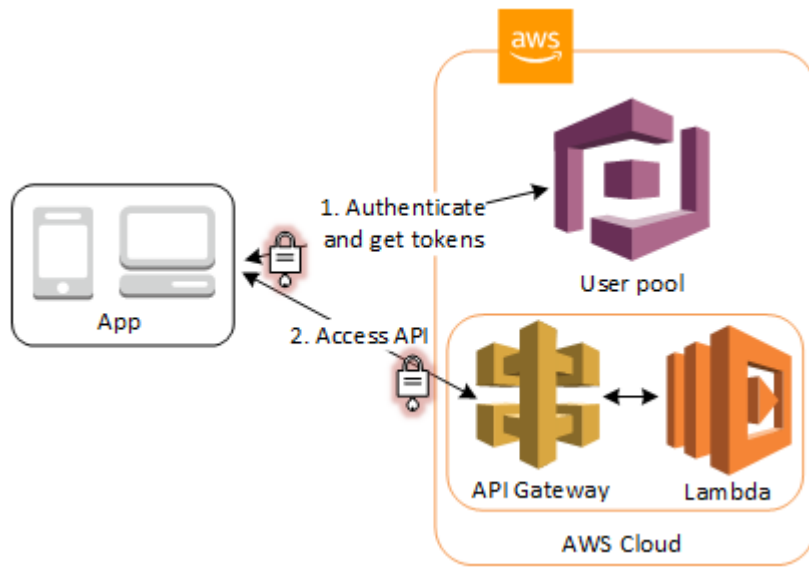
Accéder à des ressources avec API Gateway et Lambda avec un groupe d'utilisateurs

Vous pouvez permettre à vos utilisateurs d'accéder à votre API via API Gateway. API Gateway valide les jetons d'une authentification réussie d'un groupe d'utilisateurs, et les utilise pour accorder à vos utilisateurs l'accès à des ressources, dont des fonctions Lambda ou vos propres API.

Vous pouvez utiliser des groupes dans un groupe d'utilisateurs afin de contrôler les autorisations avec API Gateway en mappant l'adhésion de groupe à des rôles IAM. Les groupes dont un utilisateur est un membre sont inclus dans le jeton d'ID fourni par un groupe d'utilisateurs lorsque votre utilisateur d'application se connecte. Pour de plus amples informations sur les groupes d'utilisateurs, veuillez consulter [Ajout de groupes à un groupe d'utilisateurs](#).

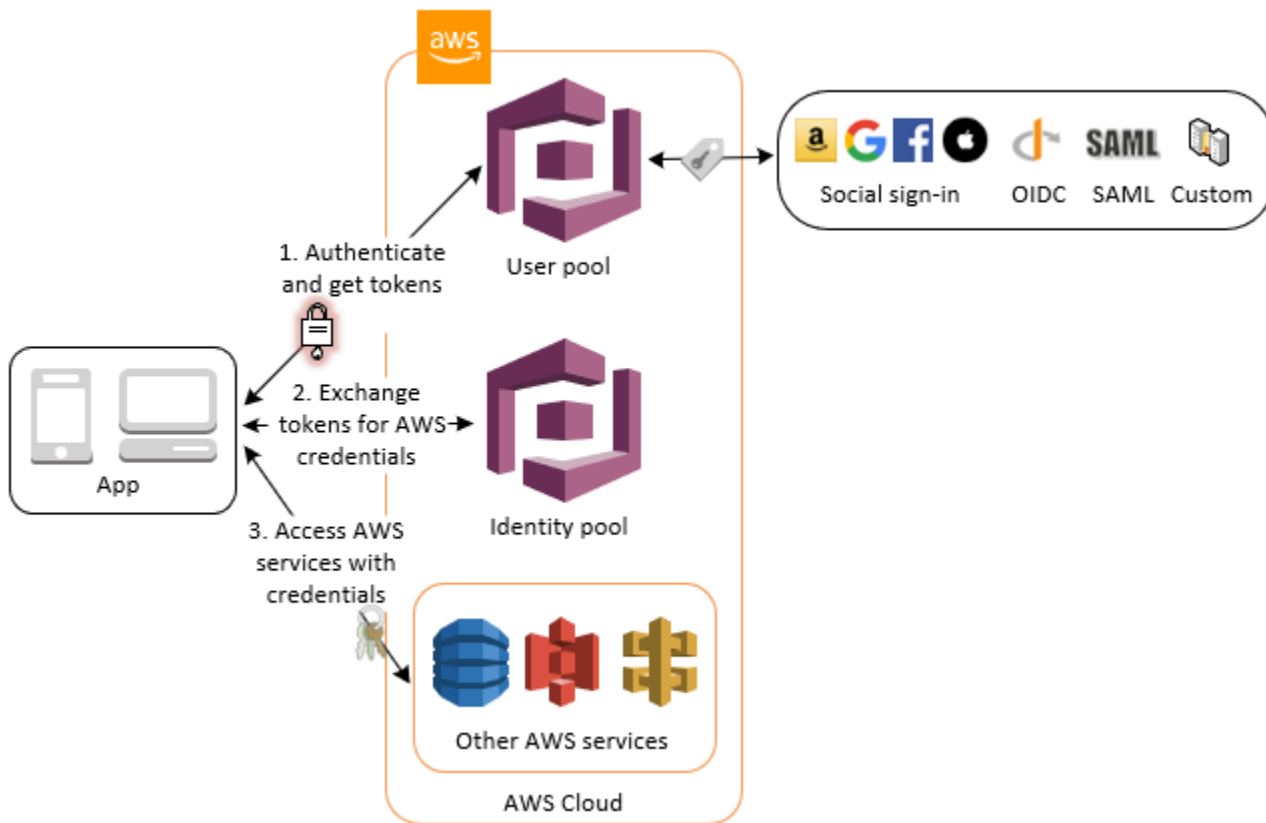
Vous pouvez envoyer vos jetons de groupe d'utilisateurs avec une demande à API Gateway pour vérification par une fonction Lambda de mécanisme d'autorisation Amazon Cognito. Pour plus

d'informations sur API Gateway, consultez [Utilisation d'API Gateway avec des groupes d'utilisateurs Amazon Cognito](#).



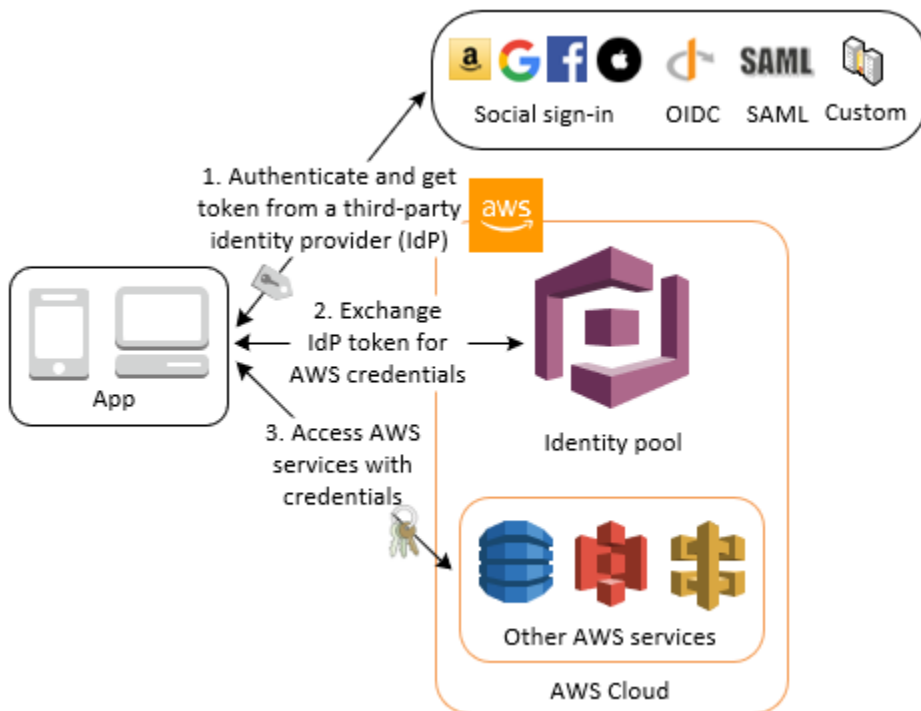
AWS Services d'accès avec un pool d'utilisateurs et un pool d'identités

Après authentification du groupe d'utilisateurs, votre application recevra d'Amazon Cognito des jetons de groupe d'utilisateurs. Vous pouvez les échanger contre un accès temporaire à d'autres AWS services dotés d'un pool d'identités. Pour plus d'informations, consultez [Accès à Services AWS l'aide d'un pool d'identités après la connexion](#) et [Commencer à utiliser les pools d'identités Amazon Cognito](#).



S'authentifier avec un tiers et accéder aux services AWS avec un groupe d'identités

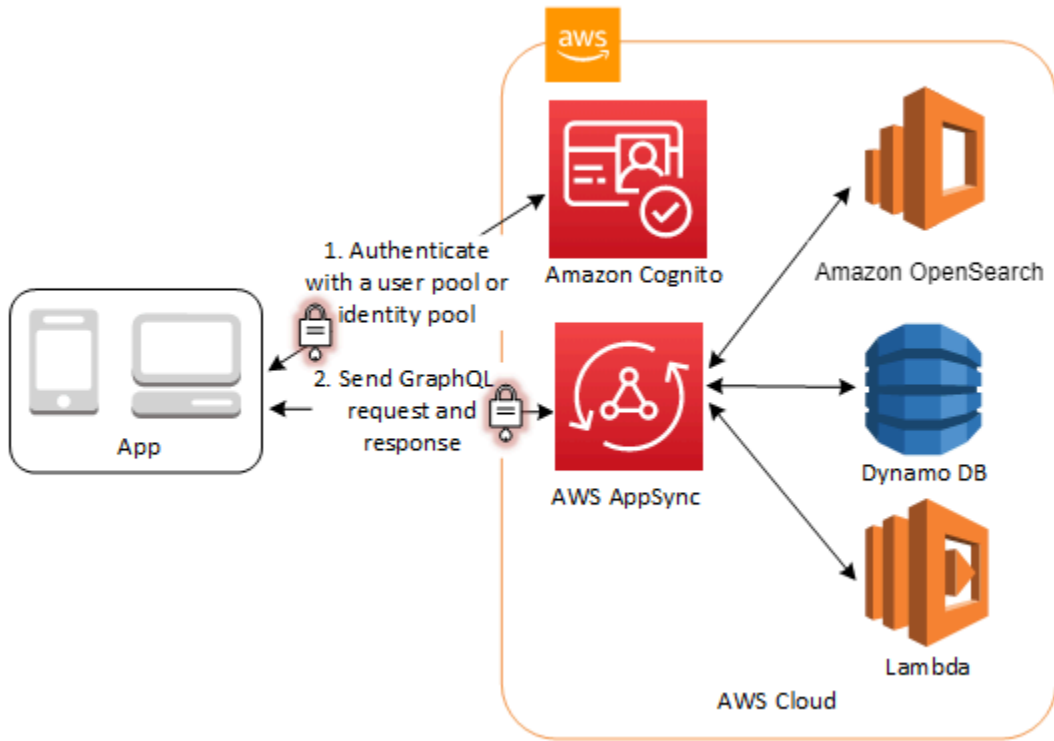
Vous pouvez permettre à vos utilisateurs d'accéder aux AWS services par le biais d'un pool d'identités. Un groupe d'identités nécessite un jeton du fournisseur d'identité de la part d'un utilisateur authentifié par un fournisseur d'identité tiers (ou rien s'il s'agit d'un invité anonyme). En échange, le pool d'identités octroie des AWS informations d'identification temporaires que vous pouvez utiliser pour accéder à d'autres AWS services. Pour plus d'informations, consultez [Commencer à utiliser les pools d'identités Amazon Cognito](#).



Accédez aux AWS AppSync ressources avec Amazon Cognito

Vous pouvez accorder à vos utilisateurs l'accès aux AWS AppSync ressources à l'aide de jetons issus d'une authentification réussie du groupe d'utilisateurs Amazon Cognito. Pour plus d'informations, consultez [Autorisation AMAZON_COGNITO_USER_POOLS](#) dans le Guide du développeur AWS AppSync .

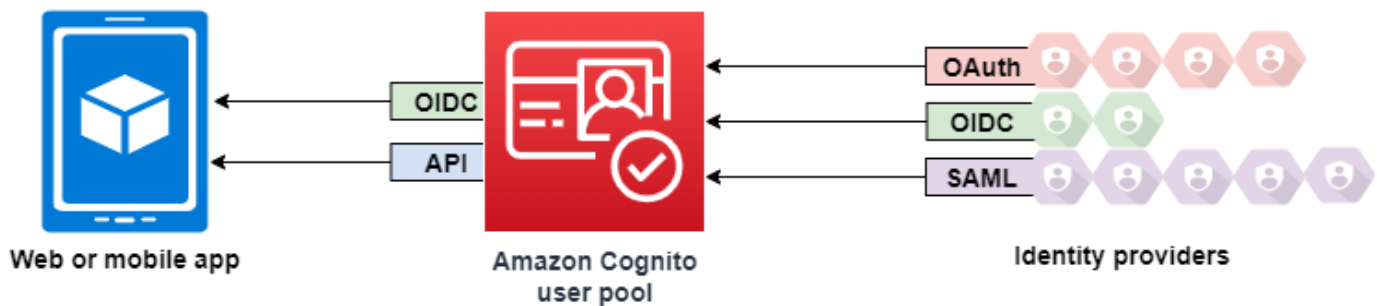
Vous pouvez également signer les demandes adressées à l'API AWS AppSync GraphQL avec les informations d'identification IAM que vous recevez d'un pool d'identités. Consultez [Autorisation AWS_IAM](#).



Groupes d'utilisateurs Amazon Cognito

Un groupe d'utilisateurs Amazon Cognito est un annuaire d'utilisateurs pour l'authentification et l'autorisation d'applications Web et mobiles. Du point de vue de votre application, un groupe d'utilisateurs Amazon Cognito est un fournisseur d'identité (IdP) OpenID Connect (OIDC). Un groupe d'utilisateurs ajoute plusieurs niveaux de fonctionnalités supplémentaires pour la sécurité, la fédération d'identité, l'intégration d'applications et la personnalisation de l'expérience utilisateur.

Vous pouvez, par exemple, vérifier que les sessions de vos utilisateurs proviennent de sources fiables. Vous pouvez combiner l'annuaire Amazon Cognito avec un fournisseur d'identité externe. Avec votre AWS SDK préféré, vous pouvez choisir le modèle d'autorisation d'API le mieux adapté à votre application. Vous pouvez également ajouter des fonctions AWS Lambda qui modifient ou révisent le comportement par défaut d'Amazon Cognito.



Rubriques

- [Fonctionnalités](#)
- [Authentification avec un groupe d'utilisateurs](#)
- [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#)
- [Mise à jour de la configuration d'un groupe d'utilisateurs](#)
- [Configuration et utilisation de l'interface utilisateur hébergée par Amazon Cognito et des points de terminaison de fédération](#)
- [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#)
- [Ajout de la connexion du groupe d'utilisateurs via un tiers](#)
- [Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda](#)
- [Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito.](#)

- [Gestion des utilisateurs dans votre groupe d'utilisateurs](#)
- [Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito](#)
- [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#)
- [Utilisation des jetons avec des groupes d'utilisateurs](#)
- [Accès aux ressources après avoir réussi l'authentification d'un groupe d'utilisateurs](#)
- [Utiliser les fonctions de sécurité des groupes d'utilisateurs Amazon Cognito](#)

Fonctionnalités

Les groupes d'utilisateurs Amazon Cognito présentent les fonctionnalités suivantes.

Inscription

Les groupes d'utilisateurs Amazon Cognito disposent de méthodes pilotées par les utilisateurs, pilotées par les administrateurs et de programmation pour ajouter des profils utilisateur à votre groupe d'utilisateurs. Les groupes d'utilisateurs Amazon Cognito prennent en charge les modèles d'inscription suivants. Vous pouvez utiliser toute combinaison de ces modèles dans votre application.

Important

Si vous activez l'inscription des utilisateurs dans votre groupe d'utilisateurs, n'importe qui sur Internet peut créer un compte et se connecter à vos applications. N'activez pas l'auto-inscription dans votre groupe d'utilisateurs, sauf si vous souhaitez ouvrir votre application à des inscriptions publiques. Pour modifier ce paramètre, mettez à jour l'inscription en libre-service dans l'onglet Expérience d'inscription de la console du pool d'utilisateurs, ou mettez à jour la valeur de [AllowAdminCreateUserOnly](#) dans une demande d'API [CreateUserPool](#). [UpdateUserPool](#)

Pour plus d'informations sur les fonctionnalités de sécurité que vous pouvez configurer dans vos groupes d'utilisateurs, consultez [Utiliser les fonctions de sécurité des groupes d'utilisateurs Amazon Cognito](#).

1. Vos utilisateurs peuvent saisir leurs informations dans votre application et créer un profil utilisateur natif de votre groupe d'utilisateurs. Vous pouvez appeler des opérations d'inscription à l'API pour enregistrer des utilisateurs dans votre groupe d'utilisateurs. Vous pouvez ouvrir ces opérations

- d'inscription à n'importe qui, ou vous pouvez les autoriser avec un secret client ou des AWS informations d'identification.
2. Vous pouvez rediriger les utilisateurs vers un fournisseur d'identité tiers qu'ils peuvent autoriser à transmettre leurs informations à Amazon Cognito. Amazon Cognito traite les jetons d'identification OIDC, les données `userInfo` OAuth 2.0 et les assertions SAML 2.0 dans les profils utilisateur de votre groupe d'utilisateurs. Vous contrôlez les attributs que vous souhaitez qu'Amazon Cognito reçoive en fonction des règles de mappage d'attributs.
 3. Vous pouvez ignorer les inscriptions publique et fédérée et créer des utilisateurs en fonction de votre propre source de données et de votre propre schéma. Ajoutez des utilisateurs directement dans l'API ou la console Amazon Cognito. Importez des utilisateurs depuis un fichier CSV. Exécutez une just-in-time AWS Lambda fonction qui recherche votre nouvel utilisateur dans un répertoire existant et remplit son profil utilisateur à partir des données existantes.

Une fois que vos utilisateurs se sont inscrits, vous pouvez les ajouter aux groupes qu'Amazon Cognito répertorie dans les jetons d'accès et d'identification. Vous pouvez également lier des ensembles de groupes d'utilisateurs à des rôles IAM lorsque vous transmettez le jeton d'identification à une réserve d'identités.

Rubriques en relation

- [Gestion des utilisateurs dans votre groupe d'utilisateurs](#)
- [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#)
- [Exemples de code pour le fournisseur d'identité Amazon Cognito utilisant des SDK AWS](#)

Connexion

Amazon Cognito peut être un annuaire d'utilisateurs autonome et un fournisseur d'identité (IdP) pour votre application. Vos utilisateurs peuvent se connecter avec une interface utilisateur hébergée par Amazon Cognito ou avec votre propre interface utilisateur via l'API des groupes d'utilisateurs Amazon Cognito. Le niveau d'application qui sous-tend votre interface utilisateur personnalisée de front-end peut autoriser les demandes sur le backend à l'aide de plusieurs méthodes pour confirmer les demandes légitimes.

Pour connecter des utilisateurs à l'aide d'un annuaire externe, éventuellement combiné à l'annuaire d'utilisateurs intégré à Amazon Cognito, vous pouvez ajouter les intégrations suivantes.

1. Connectez-vous et importez les données utilisateur des consommateurs à l'aide d'une connexion par réseau social OAuth 2.0. Amazon Cognito prend en charge la connexion avec Google, Facebook, Amazon et Apple via OAuth 2.0.
2. Connectez-vous et importez les données utilisateur de l'entreprise avec la connexion SAML et OIDC. Vous pouvez également configurer Amazon Cognito pour accepter les champs standard provenant de tout fournisseur d'identité (IdP) SAML ou OpenID Connect (OIDC).
3. Liez les profils utilisateur externes à des profils utilisateur natifs. Un utilisateur lié peut se connecter avec une identité d'utilisateur tiers et recevoir l'accès que vous attribuez à un utilisateur dans l'annuaire intégré.

Rubriques en relation

- [Ajout de la connexion du groupe d'utilisateurs via un tiers](#)
- [Liaison d'utilisateurs fédérés à un profil utilisateur existant](#)

Machine-to-machine Autorisation M2M

Certaines sessions ne sont pas des human-to-machine interactions. Vous aurez peut-être besoin d'un compte de service capable d'autoriser une demande à une API par le biais d'un processus automatisé. [Pour générer des jetons d'accès à des fins machine-to-machine d'autorisation avec des étendues OAuth 2.0, vous pouvez ajouter un client d'application qui génère des autorisations d'identification client.](#)

Rubriques en relation

- [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#)

Interface utilisateur hébergée

Si vous ne souhaitez pas créer d'interface utilisateur, vous pouvez proposer à vos utilisateurs une interface utilisateur personnalisée hébergée Amazon Cognito. L'interface utilisateur hébergée est un ensemble de pages Web pour l'inscription, la connexion, l'authentification multifactorielle (MFA) et la réinitialisation du mot de passe. Vous pouvez ajouter l'interface utilisateur hébergée à votre domaine existant ou utiliser un identifiant de préfixe dans un AWS sous-domaine.

Rubriques en relation

- [Configuration et utilisation de l'interface utilisateur hébergée par Amazon Cognito et des points de terminaison de fédération](#)
- [Configuration d'un domaine de groupe d'utilisateurs](#)

Sécurité

Vos utilisateurs locaux peuvent fournir un facteur d'authentification supplémentaire avec un code issu d'un SMS ou d'une application qui génère des codes d'authentification multifactorielle (MFA). Vous pouvez créer des mécanismes pour configurer et traiter l'authentification multifactorielle (MFA) dans votre application, ou vous pouvez laisser l'interface utilisateur hébergée s'en occuper. Les groupes d'utilisateurs Amazon Cognito peuvent contourner l'authentification multifactorielle (MFA) quand vos utilisateurs se connectent à partir d'appareils de confiance.

Si vous ne souhaitez pas exiger initialement l'authentification multifactorielle (MFA) de la part de vos utilisateurs, vous pouvez l'exiger de manière conditionnelle. Grâce à des fonctionnalités de sécurité avancées, Amazon Cognito peut détecter une activité malveillante potentielle et demander à votre utilisateur de configurer l'authentification multifactorielle (MFA) ou de bloquer la connexion.

Si le trafic réseau vers votre groupe d'utilisateurs est potentiellement malveillant, vous pouvez le surveiller et agir à l'aide des ACL AWS WAF Web.

Rubriques en relation

- [Ajout de l'authentification MFA à un groupe d'utilisateurs](#)
- [Ajout de la sécurité avancée à un groupe d'utilisateurs](#)
- [Associer une ACL AWS WAF Web à un groupe d'utilisateurs](#)

Expérience utilisateur personnalisée

À la plupart des étapes de l'inscription, de la connexion ou de la mise à jour du profil d'un utilisateur, vous pouvez personnaliser la façon dont Amazon Cognito traite la demande. Les déclencheurs Lambda vous permettent de modifier un jeton d'identification ou de rejeter une demande d'inscription en fonction de conditions personnalisées. Vous pouvez créer votre propre flux d'authentification personnalisé.

Vous pouvez charger un fichier CSS et des logos personnalisés pour donner à l'interface utilisateur hébergée un aspect familier pour vos utilisateurs.

Rubriques en relation

- [Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda](#)
- [Déclencheurs Lambda création d'une stimulation d'authentification personnalisée](#)
- [Personnalisation des pages web intégrées d'inscription et de connexion](#)

Surveillance et analytique

Les groupes d'utilisateurs Amazon Cognito consignent les demandes d'API, y compris celles destinées à l'interface utilisateur hébergée, dans AWS CloudTrail. Vous pouvez consulter les indicateurs de performance dans Amazon CloudWatch Logs, envoyer des journaux personnalisés à l' CloudWatch aide de déclencheurs Lambda et surveiller le volume de demandes d'API dans la console Service Quotas.

Vous pouvez également consigner les données des appareils et des sessions à partir de vos demandes d'API dans une campagne Amazon Pinpoint. Avec Amazon Pinpoint, vous pouvez envoyer des notifications push depuis votre application en fonction de votre analyse de l'activité des utilisateurs.

Rubriques en relation

- [Journalisation des appels d'API Amazon Cognito avec AWS CloudTrail](#)
- [Suivi des quotas, de l'utilisation CloudWatch et des Quotas de Service](#)
- [Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito.](#)

Intégration des réserves d'identités Amazon Cognito

L'autre moitié d'Amazon Cognito est constituée des réserves d'identités. Les pools d'identités fournissent des informations d'identification qui autorisent et surveillent les demandes d'API envoyées par vos utilisateurs Services AWS, par exemple à Amazon DynamoDB ou Amazon S3. Vous pouvez créer des stratégies d'accès basées sur l'identité qui protègent vos données en fonction de la manière dont vous classez les utilisateurs dans votre groupe d'utilisateurs. Les réserves d'identités peuvent également accepter des jetons et des assertions SAML 2.0 provenant de divers fournisseurs d'identité, indépendamment de l'authentification des groupes d'utilisateurs.

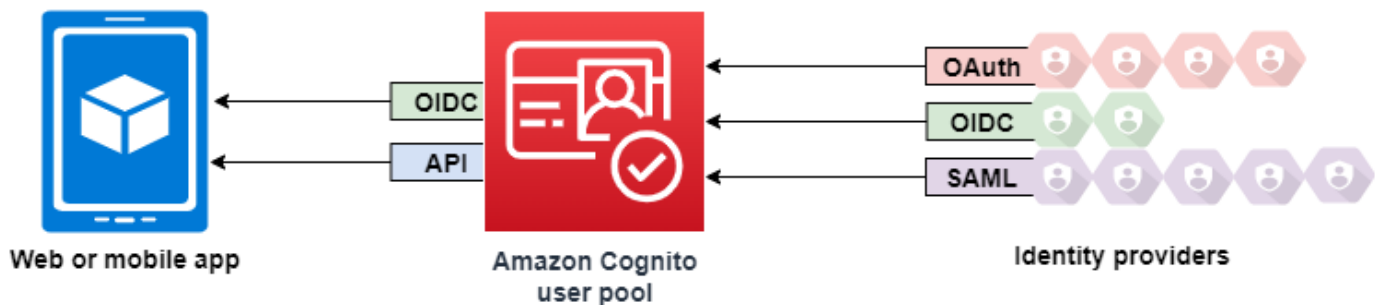
Rubriques en relation

- [Accès à Services AWS l'aide d'un pool d'identités après la connexion](#)
- [Groupes d'identités Amazon Cognito](#)

Authentification avec un groupe d'utilisateurs

Les utilisateurs de votre application peuvent soit se connecter directement via un groupe d'utilisateurs, soit fédérer via un fournisseur d'identité (IdP) tiers. Le groupe d'utilisateurs gère les frais généraux liés à la gestion des jetons renvoyés lors de la connexion aux réseaux sociaux via Facebook, Google, Amazon et Apple, et depuis OpenID Connect (OIDC) et SAML. IdPs

Après authentification, Amazon Cognito renvoie des jetons de groupe d'utilisateurs à votre application. Vous pouvez utiliser les jetons pour accorder à vos utilisateurs l'accès à vos ressources côté serveur, ou au service Amazon API Gateway. Vous pouvez également les échanger contre des AWS informations d'identification pour accéder à d'autres AWS services.



Le traitement et la gestion des jetons de groupe d'utilisateurs pour votre application web ou mobile sont assurés côté client via des kits SDK Amazon Cognito. De même, les kits SDK Mobile pour iOS et Android actualisent automatiquement vos jetons d'identification et d'accès si un jeton d'actualisation valide (non expiré) est présent, et que les jetons d'identification et d'accès ont une durée de validité restante d'au moins 5 minutes. Pour plus d'informations sur les SDK et des exemples de code pour Android et iOS JavaScript, consultez les SDK du pool [d'utilisateurs Amazon Cognito](#).

Une fois votre utilisateur d'appli connecté, Amazon Cognito crée une session et renvoie un jeton d'identification, d'accès et d'actualisation pour l'utilisateur authentifié.

JavaScript

```
// Amazon Cognito creates a session which includes the id, access, and refresh
tokens of an authenticated user.

var authenticationData = {
    Username : 'username',
    Password : 'password',
};
var authenticationDetails = new
AmazonCognitoIdentity.AuthenticationDetails(authenticationData);
var poolData = { UserPoolId : 'us-east-1_Example',
    ClientId : '1example23456789'
};
var userPool = new AmazonCognitoIdentity.CognitoUserPool(poolData);
var userData = {
    Username : 'username',
    Pool : userPool
};
var cognitoUser = new AmazonCognitoIdentity.CognitoUser(userData);
cognitoUser.authenticateUser(authenticationDetails, {
    onSuccess: function (result) {
        var accessToken = result.getAccessToken().getJwtToken();

        /* Use the idToken for Logins Map when Federating User Pools with
identity pools or when passing through an Authorization Header to an API Gateway
Authorizer */
        var idToken = result.idToken.jwtToken;
    },

    onFailure: function(err) {
        alert(err);
    },
});
```

Android

```
// Session is an object of the type CognitoUserSession, and includes the id, access,
and refresh tokens for a user.

String idToken = session.getIdToken().getJWTToken();
String accessToken = session.getAccessToken().getJWT();
```

iOS - swift

```
// AWSCognitoIdentityUserSession includes id, access, and refresh tokens for a user.  
  
- (AWSTask<AWSCognitoIdentityUserSession *> *)getSession;
```

iOS - objective-C

```
// AWSCognitoIdentityUserSession includes the id, access, and refresh tokens for a  
user.  
  
[[user getSession:@"username" password:@"password" validationData:nil scopes:nil]  
  continueWithSuccessBlock:^id _Nullable(AWSTask<AWSCognitoIdentityUserSession *> *  
    _Nonnull task) {  
    // success, task.result has user session  
    return nil;  
  }];
```

Rubriques

- [Flux d'authentification de groupe d'utilisateurs](#)
- [Clients de l'application de groupe d'utilisateurs](#)
- [Utilisation d'appareils utilisateur dans votre groupe d'utilisateurs](#)

Flux d'authentification de groupe d'utilisateurs

Amazon Cognito propose plusieurs méthodes d'authentification des utilisateurs. Tous les groupes d'utilisateurs, que vous possédiez un domaine ou non, peuvent authentifier les utilisateurs dans l'API des groupes d'utilisateurs. Si vous ajoutez un domaine à votre groupe d'utilisateurs, vous pouvez utiliser les [points de terminaison de groupe d'utilisateurs](#). L'API des groupes d'utilisateurs prend en charge divers modèles d'autorisation et flux de demandes pour les demandes d'API.

Pour vérifier l'identité des utilisateurs, Amazon Cognito prend en charge les flux d'authentification qui intègrent de nouveaux types de question de sécurité, en plus des mots de passe. L'authentification Amazon Cognito nécessite généralement que vous implémentiez deux opérations d'API dans l'ordre suivant :

Public authentication

1. [InitiateAuth](#)
2. [RespondToAuthChallenge](#)

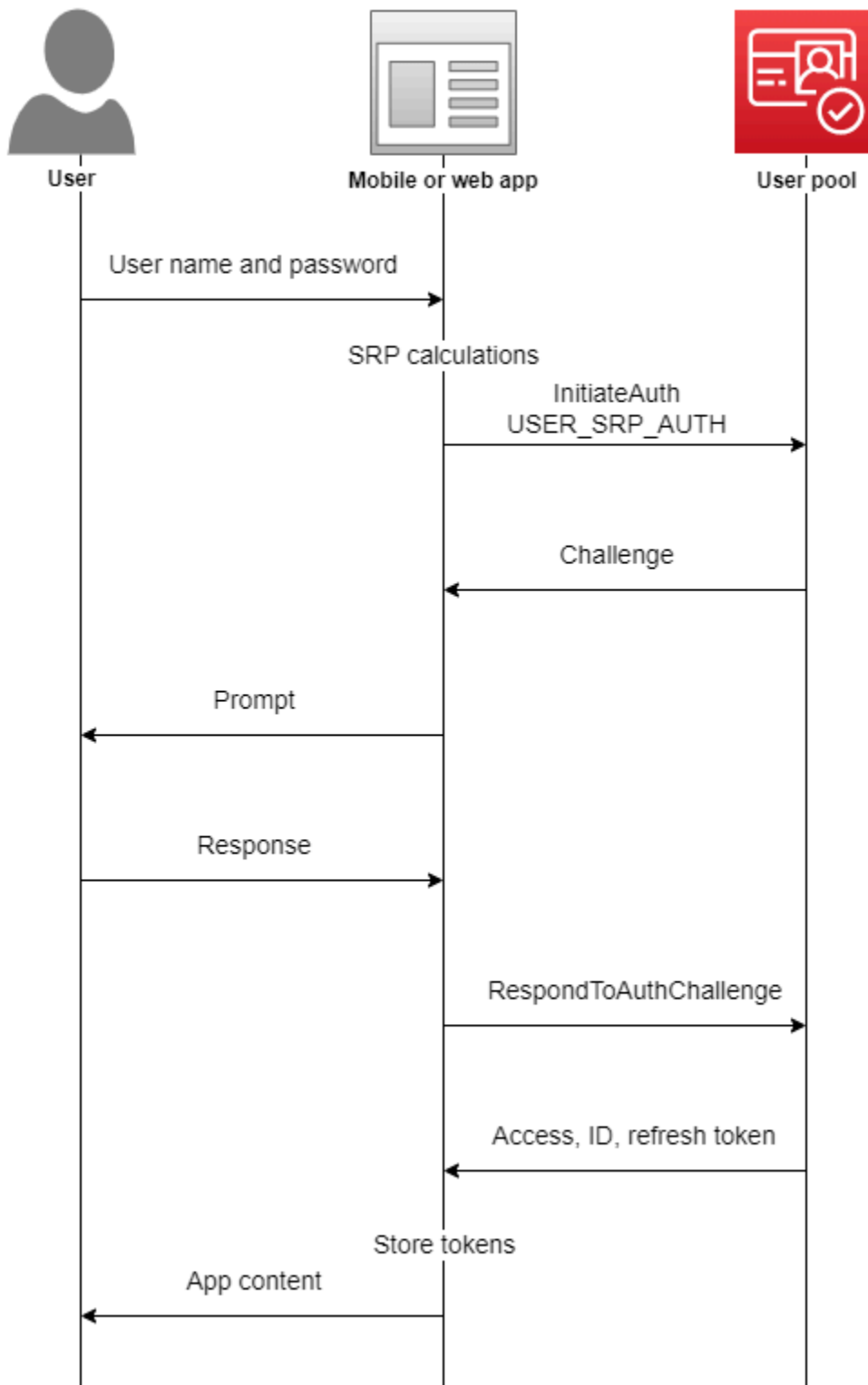
`InitiateAuth` et `RespondToAuthChallenge` sont des API non authentifiées destinées à être utilisées avec des clients d'application publique côté client.

Server-side authentication

1. [AdminInitiateAuth](#)
2. [AdminRespondToAuthChallenge](#)

`AdminInitiateAuth` et `AdminRespondToAuthChallenge` nécessitent des informations d'identification IAM et conviennent pour les clients d'application confidentielle côté serveur.

Un utilisateur s'authentifie en répondant à des demandes de vérification successives jusqu'à ce que l'authentification échoue ou qu'Amazon Cognito émette des jetons pour l'utilisateur. Vous pouvez répéter ces étapes avec Amazon Cognito, dans un processus qui inclut différentes demandes de vérification, pour prendre en charge n'importe quel flux d'authentification personnalisé.



En général, votre application génère une invite pour recueillir des informations auprès de votre utilisateur et soumet ces informations dans une demande d'API à Amazon Cognito. Envisagez un flux

`InitiateAuth` dans un groupe d'utilisateurs dans lequel vous avez configuré votre utilisateur avec l'authentification multifactorielle (MFA).

1. Votre application demande à votre utilisateur de fournir son nom d'utilisateur et son mot de passe.
2. Vous incluez le nom d'utilisateur et le mot de passe en tant que paramètres dans `InitiateAuth`.
3. Amazon Cognito renvoie un défi `SMS_MFA` et un identifiant de session.
4. Votre application invite l'utilisateur à saisir le code MFA depuis son téléphone.
5. Vous devez inclure ce code et l'identifiant de session dans la demande `RespondToAuthChallenge`.

Selon les fonctionnalités de votre groupe d'utilisateurs, vous pouvez vous retrouver face à plusieurs défis `InitiateAuth` avant que votre application ne récupère des jetons depuis Amazon Cognito. Amazon Cognito inclut une chaîne de session dans la réponse à chaque demande. Pour combiner vos requêtes d'API dans un flux d'authentification, incluez la chaîne de session de la réponse à la demande précédente dans chaque demande suivante. Par défaut, vos utilisateurs disposent de trois minutes pour terminer chaque défi avant l'expiration de la chaîne de session. Pour modifier cette période, modifiez la Durée de session d'authentification du client d'application. La procédure suivante explique comment modifier ce paramètre dans la configuration de votre client d'application.

Note

Les paramètres de durée de session du flux d'authentification s'appliquent à l'authentification avec l'API des groupes d'utilisateurs Amazon Cognito. L'interface utilisateur hébergée Amazon Cognito définit la durée de session sur 3 minutes pour l'authentification multifactorielle et sur 8 minutes pour les codes de réinitialisation de mot de passe.

Amazon Cognito console

Pour configurer la durée de session du flux d'authentification du client d'application (AWS Management Console)

1. À partir de l'onglet App integration (Intégration d'applications) dans votre groupe d'utilisateurs, sélectionnez le nom de votre client d'application à partir du conteneur App clients and analytics (Clients d'applications et analyses).
2. Choisissez Modifier dans le conteneur Informations sur le client d'application.

3. Modifiez la valeur de `Authentication flow session duration` (Durée de session du flux d'authentification) pour indiquer la durée de validité choisie (en minutes) pour les codes MFA par SMS. Cela modifie également le temps dont dispose chaque utilisateur pour réaliser un défi d'authentification dans votre client d'application.
4. Sélectionnez `Enregistrer les modifications`.

Amazon Cognito API

Pour configurer la durée de session du flux d'authentification du client d'application (API Amazon Cognito)

1. Préparez une demande `UpdateUserPoolClient` avec vos paramètres de groupe d'utilisateurs existants à partir d'une demande `DescribeUserPoolClient`. Votre demande `UpdateUserPoolClient` doit inclure toutes les propriétés du client d'application existant.
2. Modifiez la valeur de `AuthSessionValidity` et indiquez la durée de validité (en minutes) choisie pour les codes MFA par SMS. Cela modifie également le temps dont dispose chaque utilisateur pour réaliser un défi d'authentification dans votre client d'application.

Pour plus d'informations sur les clients d'application, consultez [Clients de l'application de groupe d'utilisateurs](#).

Vous pouvez utiliser des AWS Lambda déclencheurs pour personnaliser la façon dont les utilisateurs s'authentifient. Ces déclencheurs émettent et vérifient leurs propres défis dans le cadre du flux d'authentification.

Vous pouvez également utiliser le flux d'authentification de l'administration pour des serveurs backend sécurisés. Vous pouvez utiliser le flux d'authentification de migration des utilisateurs pour permettre la migration des utilisateurs sans exiger de vos utilisateurs qu'ils réinitialisent leurs mots de passe.

Comportement de verrouillage d'Amazon Cognito en cas d'échec des tentatives de connexion

Après cinq tentatives infructueuses de connexion non authentifiée ou authentifiée par IAM avec un mot de passe, Amazon Cognito verrouille votre utilisateur pendant une seconde. La durée du verrouillage double ensuite après chaque nouvelle tentative infructueuse, jusqu'à un maximum d'environ 15 minutes. Les tentatives effectuées pendant une période de verrouillage génèrent une exception `Password attempts exceeded` et n'affectent pas la durée des périodes de verrouillage suivantes. Pour un nombre cumulé de tentatives de connexion infructueuses n , à l'exclusion des

exceptions `Password attempts exceeded`, Amazon Cognito verrouille votre utilisateur pendant $2^{(n-5)}$ secondes. Pour restaurer l'état initial du verrouillage ($n=0$), votre utilisateur doit parvenir à se connecter à l'issue d'une période de verrouillage ou ne lancer à aucun moment de tentative de connexion pendant 15 minutes consécutives après le verrouillage. Ce comportement est susceptible d'être modifié. Ce comportement ne s'applique pas aux défis personnalisés à moins qu'ils n'effectuent également une authentification basée sur un mot de passe.

Rubriques

- [Flux d'authentification côté client](#)
- [Flux d'authentification côté serveur](#)
- [Flux d'authentification personnalisé](#)
- [Flux d'authentification intégré et stimulations](#)
- [Flux d'authentification personnalisé et stimulations](#)
- [Utiliser la vérification de mot de passe par protocole SRP dans le flux d'authentification personnalisé](#)
- [Flux d'authentification de l'administration](#)
- [Flux d'authentification pour la migration d'utilisateurs](#)

Flux d'authentification côté client

Le processus suivant fonctionne pour les applications côté client que vous créez avec [AWS Amplify](#) ou les [kits SDK AWS](#).

1. L'utilisateur entre le nom d'utilisateur et le mot de passe dans l'application.
2. L'application appelle l'opération `InitiateAuth` avec le nom d'utilisateur et les informations SRP (Secure Remote Password) détaillées de l'utilisateur.

Cette opération d'API renvoie les paramètres d'authentification.

Note

L'application génère des détails SRP avec les fonctionnalités Amazon Cognito SRP intégrées aux kits AWS SDK.

3. L'appli appelle l'opération `RespondToAuthChallenge`. Si l'appel aboutit, Amazon Cognito retourne les jetons de l'utilisateur et le flux d'authentification est terminé.

Si Amazon Cognito requiert une autre demande de vérification, l'appel à `RespondToAuthChallenge` ne renvoie aucun jeton. À la place, l'appel renvoie une session.

4. Si `RespondToAuthChallenge` renvoie une session, l'appli appelle `RespondToAuthChallenge` à nouveau, cette fois avec la session et la réponse à la stimulation (par exemple, le code MFA).

Flux d'authentification côté serveur

Si vous ne disposez pas d'une application d'utilisateur, mais utilisez une application côté serveur ou backend sécurisée Java, Ruby ou Node.js, vous pouvez utiliser l'API côté serveur authentifiée pour les groupes d'utilisateurs Amazon Cognito.

Pour les applications côté serveur, l'authentification d'un groupe d'utilisateurs est similaire à celle des applications côté client, à l'exception des aspects suivants :

- L'appli côté serveur appelle l'opération d'API `AdminInitiateAuth` (au lieu de `InitiateAuth`). Cette opération nécessite des AWS informations d'identification avec des autorisations telles que `cognito-idp:AdminInitiateAuth` et `cognito-idp:AdminRespondToAuthChallenge`. Cette opération renvoie les paramètres d'authentification requis.
- Une fois que l'application côté serveur possède les paramètres d'authentification, elle appelle l'opération d'API `AdminRespondToAuthChallenge` (à la place de `RespondToAuthChallenge`). L'opération `AdminRespondToAuthChallenge` d'API ne réussit que lorsque vous fournissez des AWS informations d'identification.

Pour plus d'informations sur la signature des demandes d'API Amazon Cognito avec des AWS informations d'identification, consultez le [processus de signature Signature version 4](#) dans le manuel de référence AWS général.

Les opérations `AdminInitiateAuth` et `AdminRespondToAuthChallenge` API ne peuvent pas accepter les informations `username-and-password` d'identification des utilisateurs pour la connexion des administrateurs, sauf si vous les autorisez explicitement à le faire de l'une des manières suivantes :

- Incluez `ALLOW_ADMIN_USER_PASSWORD_AUTH` (anciennement connu sous le nom `ADMIN_NO_SRP_AUTH`) dans le paramètre `ExplicitAuthFlow` lorsque vous appelez `CreateUserPoolClient` ou `UpdateUserPoolClient`.

- Ajoutez `ALLOW_ADMIN_USER_PASSWORD_AUTH` dans la liste Flux d'authentification pour votre client d'application. Configurez les clients d'application dans l'onglet App integration (Intégration d'application) de votre groupe d'utilisateurs, sous App clients and analytics (Clients d'application et analytique). Pour de plus amples informations, veuillez consulter [Clients de l'application de groupe d'utilisateurs](#).

Flux d'authentification personnalisé

Les groupes d'utilisateurs Amazon Cognito permettent également d'utiliser des flux d'authentification personnalisés, qui peuvent vous aider à créer un modèle d'authentification basé sur les défis et les réponses à l'aide de déclencheurs. AWS Lambda

Note

Vous ne pouvez pas utiliser les fonctionnalités de sécurité avancées pour des informations d'identification compromises et l'authentification adaptative avec des flux d'authentification personnalisés. Pour de plus amples informations, veuillez consulter [Ajout de la sécurité avancée à un groupe d'utilisateurs](#).

Le flux d'authentification personnalisé permet des cycles de stimulation/réponse personnalisés pour répondre à des besoins différents. Le flux commence par un appel à l'opération d'API `InitiateAuth` qui indique le type d'authentification qui sera utilisé, et fournit les paramètres d'authentification initiaux. Amazon Cognito répond à l'appel `InitiateAuth` avec l'un des types d'informations suivants :

- Une stimulation pour l'utilisateur avec une session et des paramètres.
- Une erreur si l'utilisateur ne parvient pas à s'authentifier.
- Les jetons d'identification, d'accès et d'actualisation si les paramètres fournis dans l'appel `InitiateAuth` sont suffisants pour connecter l'utilisateur. (En règle générale, l'utilisateur ou l'appli doit d'abord répondre à une stimulation, mais votre code personnalisé doit le déterminer.)

Si Amazon Cognito répond à l'appel `InitiateAuth` avec une demande de vérification, l'application recueille davantage d'informations et appelle l'opération `RespondToAuthChallenge`. Cet appel fournit les réponses à la demande de vérification et les renvoie à la session. Amazon Cognito répond à l'appel `RespondToAuthChallenge` de la même manière qu'à l'appel `InitiateAuth`. Si l'utilisateur s'est connecté, Amazon Cognito fournit des jetons ou si l'utilisateur n'est pas

connecté, Amazon Cognito fournit une autre demande de vérification ou une erreur. Si Amazon Cognito renvoie une autre demande de vérification, la séquence se reproduit et l'application appelle `RespondToAuthChallenge` jusqu'à ce que l'utilisateur se connecte avec succès ou qu'une erreur soit retournée. Pour plus d'informations sur les opérations d'API `InitiateAuth` et `RespondToAuthChallenge`, consultez la [documentation sur les API](#).

Flux d'authentification intégré et stimulations

Amazon Cognito contient des valeurs `AuthFlow` et `ChallengeName` intégrées de sorte qu'un flux d'authentification standard puisse valider un nom d'utilisateur et un mot de passe via le protocole SRP (Secure Remote Password). Les AWS SDK prennent en charge ces flux de manière intégrée avec Amazon Cognito.

Le flux commence par l'envoi de `USER_SRP_AUTH` comme `AuthFlow` pour `InitiateAuth`. Vous envoyez également les valeurs `USERNAME` et `SRP_A` dans `AuthParameters`. Si l'appel `InitiateAuth` aboutit, la réponse a inclus `PASSWORD_VERIFIER` comme `ChallengeName` et `SRP_B` dans les paramètres de stimulation. L'appli appelle ensuite `RespondToAuthChallenge` avec le `PASSWORD_VERIFIER` `ChallengeName` et les paramètres nécessaires dans `ChallengeResponses`. Si l'appel à `RespondToAuthChallenge` réussit et que l'utilisateur est connecté, Amazon Cognito émet des jetons. Si vous avez activé l'authentification multifacteur (MFA) pour l'utilisateur, Amazon Cognito renvoie `ChallengeName` de `SMS_MFA`. L'application peut fournir le code nécessaire via un autre appel à `RespondToAuthChallenge`.

Flux d'authentification personnalisé et stimulations

Pour initier un flux d'authentification personnalisé, une appli peut appeler `InitiateAuth` avec `CUSTOM_AUTH` comme paramètre `AuthFlow`. Avec un flux d'authentification personnalisé, trois déclencheurs Lambda contrôlent les demandes de vérification et la vérification des réponses.

- Le déclencheur Lambda `DefineAuthChallenge` utilise en entrée un tableau de session de demandes de vérification et de réponses précédentes. Il affiche ensuite le nom de la demande de vérification suivante et les booléens qui indiquent si l'utilisateur est authentifié et peut recevoir des jetons. Ce déclencheur Lambda est une machine d'état qui contrôle le parcours de l'utilisateur au fil des stimulations.
- Le déclencheur Lambda `CreateAuthChallenge` prend un nom de demande de vérification en entrée et génère le défi et les paramètres permettant d'évaluer la réponse. Quand `DefineAuthChallenge` retourne `CUSTOM_CHALLENGE` comme demande de vérification suivante, le flux d'authentification appelle `CreateAuthChallenge`. Le déclencheur Lambda

`CreateAuthChallenge` transmet le type de demande de vérification suivant dans le paramètre de métadonnées de demande de vérification.

- La fonction Lambda `VerifyAuthChallengeResponse` évalue la réponse et renvoie une valeur booléenne indiquant si la réponse était valide.

Un flux d'authentification personnalisé peut également utiliser une combinaison de stimulations intégrées, telles que la vérification de mot de passe via le protocole SRP et la MFA par SMS. Il peut utiliser des stimulations personnalisées, telles que CAPTCHA ou des questions secrètes.

Utiliser la vérification de mot de passe par protocole SRP dans le flux d'authentification personnalisé

Si vous souhaitez inclure le protocole SRP dans un flux d'authentification personnalisé, vous devez commencer par SRP.

- Pour lancer la vérification de mot de passe par protocole SRP dans un flux personnalisé, l'appli appelle `InitiateAuth` avec `CUSTOM_AUTH` en tant que `Authflow`. Dans le mappage `AuthParameters`, la demande de votre application inclut `SRP_A`: (la valeur SRP A) et `CHALLENGE_NAME: SRP_A`.
- Le flux `CUSTOM_AUTH` invoque le déclencheur Lambda `DefineAuthChallenge` avec une session initiale de `challengeName: SRP_A` et `challengeResult: true`. Votre fonction Lambda répond avec `challengeName: PASSWORD_VERIFIER`, `issueTokens: false` et `failAuthentication: false`.
- L'appli doit ensuite appeler `RespondToAuthChallenge` avec `challengeName: PASSWORD_VERIFIER` et les autres paramètres requis pour le protocole SRP dans la carte `challengeResponses`.
- Si Amazon Cognito vérifie le mot de passe, `RespondToAuthChallenge` appelle le déclencheur Lambda `DefineAuthChallenge` avec une deuxième session de `challengeName: PASSWORD_VERIFIER` et `challengeResult: true`. À ce stade, le déclencheur Lambda `DefineAuthChallenge` répond avec `challengeName: CUSTOM_CHALLENGE` pour démarrer la stimulation personnalisée.
- Si l'authentification MFA est activée pour un utilisateur, une fois qu'Amazon Cognito a vérifié le mot de passe, l'utilisateur est invité à configurer MFA ou à se connecter avec MFA.

Note

La page web de connexion hébergée Amazon Cognito ne peut pas activer les [Déclencheurs Lambda création d'une stimulation d'authentification personnalisée](#).

Pour plus d'informations sur les déclencheurs Lambda, ainsi qu'un exemple de code, consultez [Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda](#).

Flux d'authentification de l'administration

La meilleure pratique pour l'authentification consiste à utiliser les opérations d'API décrites dans [Flux d'authentification personnalisé](#) avec SRP pour la vérification du mot de passe. Les AWS SDK utilisent cette approche, qui les aide à utiliser le SRP. Toutefois, si vous souhaitez éviter les calculs SRP, un autre ensemble d'opérations d'API d'administration est disponible pour les serveurs backend sécurisés. Pour ces implémentations d'administration backend, utilisez `AdminInitiateAuth` au lieu de `InitiateAuth`. Utilisez également `AdminRespondToAuthChallenge` au lieu de `RespondToAuthChallenge`. Étant donné que vous pouvez soumettre le mot de passe en texte brut, vous n'avez pas à effectuer de calculs SRP lorsque vous utilisez ces opérations. Voici un exemple :

```
AdminInitiateAuth Request {
  "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
  "AuthParameters": {
    "USERNAME": "<username>",
    "PASSWORD": "<password>"
  },
  "ClientId": "<clientId>",
  "UserPoolId": "<userPoolId>"
}
```

Ces opérations d'authentification de l'administration requièrent les informations d'identification d'un développeur et utilisent le processus de signature d' AWS Signature Version 4 (SigV4). Ces opérations sont disponibles dans les kits SDK AWS standard, y compris Node.js, ce qui est pratique avec les fonctions Lambda. Pour utiliser ces opérations et leur faire accepter des mots de passe en texte brut, vous devez les activer pour l'application dans la console. Vous pouvez également transmettre `ADMIN_USER_PASSWORD_AUTH` pour le paramètre `ExplicitAuthFlow` dans les appels à `CreateUserPoolClient` ou `UpdateUserPoolClient`. Les opérations `InitiateAuth` et `RespondToAuthChallenge` n'acceptent pas `ADMIN_USER_PASSWORD_AUTH` `AuthFlow`.

Dans la réponse `AdminInitiateAuth ChallengeParameters`, l'attribut `USER_ID_FOR_SRP`, s'il est présent, contient le nom d'utilisateur réel de l'utilisateur, et non pas un alias (tel qu'une adresse e-mail ou un numéro de téléphone). Dans votre appel à `AdminRespondToAuthChallenge`, dans les `ChallengeResponses`, vous devez transmettre ce nom d'utilisateur dans le paramètre `USERNAME`.

Note

Comme les implémentations d'administrateur backend utilisent le flux d'authentification d'administrateur, ce flux ne prend pas en charge la fonctionnalité de suivi des dispositifs. Lorsque la fonctionnalité de suivi des dispositifs est activée, l'authentification d'administrateur réussit, mais tout appel visant à actualiser le jeton d'accès échoue.

Flux d'authentification pour la migration d'utilisateurs

Un déclencheur Lambda de migration d'utilisateur facilite la migration d'utilisateurs à partir d'un système de gestion des utilisateurs hérité vers votre groupe d'utilisateurs. Si vous choisissez le flux d'authentification `USER_PASSWORD_AUTH`, les utilisateurs n'ont pas à réinitialiser leurs mots de passe durant la migration des utilisateurs. Ce flux envoie les mots de passe de vos utilisateurs au service via une connexion SSL cryptée pendant l'authentification.

Lorsque vous avez migré tous vos utilisateurs, changez de flux et passez au flux SRP plus sécurisé. Le flux SRP n'envoie aucun mot de passe sur le réseau.

Pour en savoir plus sur les déclencheurs Lambda, consultez [Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda](#).

Pour plus d'informations sur la migration d'utilisateurs avec un déclencheur Lambda, consultez [Importation d'utilisateurs dans des groupes d'utilisateurs avec un déclencheur Lambda de migration d'utilisateur](#).

Clients de l'application de groupe d'utilisateurs

Un client d'application de groupe d'utilisateurs est une configuration au sein d'un groupe d'utilisateurs qui interagit avec une application mobile ou Web qui s'authentifie auprès d'Amazon Cognito. Les clients d'application peuvent appeler des opérations d'API authentifiées et non authentifiées et lire ou modifier tout ou partie des attributs de vos utilisateurs. Votre application doit s'identifier auprès du client d'application dans les opérations d'enregistrement, de connexion et de gestion des mots

de passe oubliés. Ces demandes d'API doivent inclure une auto-identification à l'aide d'un ID client d'application et une autorisation avec un secret client facultatif. Vous devez sécuriser les ID ou secrets des clients d'application afin que seules les applications client autorisées puissent appeler ces opérations non authentifiées. En outre, si vous configurez votre application pour signer des demandes d'API authentifiées à l'aide d'informations d'identification AWS, vous devez protéger vos informations d'identification contre toute inspection par les utilisateurs.

Vous pouvez créer plusieurs applications pour un groupe d'utilisateurs. Un client d'application peut être lié à la plateforme de code d'une application ou à un locataire distinct dans votre groupe d'utilisateurs. Par exemple, vous pouvez créer une application pour une application côté serveur et une application Android distincte. Chaque application a son propre ID de client d'application.

Types de client d'application

Lorsque vous créez un client d'application dans Amazon Cognito, vous pouvez préremplir les options en fonction des types de client OAuth standard client public et client confidentiel. Configurez un client confidentiel avec un secret client. Pour plus d'informations sur les types de clients, consultez [IETF RFC 6749 #2.1](#).

Client public

Un client public s'exécute dans un navigateur ou sur un appareil mobile. Étant donné qu'il ne dispose pas de ressources approuvées côté serveur, il ne possède pas de secret client.

Client confidentiel

Un client confidentiel dispose de ressources côté serveur qui peuvent être approuvées à l'aide d'un secret client pour des opérations d'API non authentifiées. L'application peut s'exécuter en tant que démon ou script shell sur votre serveur backend.

Secret client

Un secret client, ou un mot de passe client, est une chaîne fixe que votre application doit utiliser dans toutes les demandes d'API adressées au client d'application. Votre client d'application doit posséder un secret client pour effectuer des octrois `client_credentials`. Pour en savoir plus, consultez [IETF RFC 6749 #2.3.1](#).

Vous ne pouvez pas modifier les secrets après avoir créé une application. Vous pouvez créer une nouvelle application avec un nouveau secret, si vous souhaitez effectuer une rotation du secret. Vous pouvez également supprimer une application pour empêcher les applications qui utilisent cet ID de client d'application d'y accéder.

Vous pouvez utiliser un client confidentiel et un secret client avec une application publique. Utilisez un CloudFront proxy Amazon pour ajouter un fichier SECRET_HASH en transit. Pour plus d'informations, consultez la section [Protéger les clients publics pour Amazon Cognito à l'aide d'un CloudFront proxy Amazon](#) sur le AWS blog.

Jetons Web JSON

Les clients de l'application Amazon Cognito peuvent émettre des jetons Web JSON (JWT) possédant les types suivants.

Jeton d'identité (ID)

Une déclaration vérifiable indiquant que votre utilisateur est authentifié à partir de votre groupe d'utilisateurs. OpenID Connect (OIDC) a ajouté la [spécification du jeton d'identification](#) aux normes des jetons d'accès et d'actualisation définies par OAuth 2.0. Le jeton d'identification contient des informations d'identité, telles que les attributs utilisateur, que votre application peut utiliser pour créer un profil utilisateur et fournir des ressources. Pour plus d'informations, consultez [Utilisation du jeton d'identification](#).

Jeton d'accès

Une déclaration vérifiable des droits d'accès de vos utilisateurs. Le jeton d'accès contient des [portées](#), une fonctionnalité d'OIDC et d'OAuth 2.0. Votre application peut présenter des portées aux ressources principales et prouver que votre groupe d'utilisateurs a autorisé un utilisateur ou une machine à accéder aux données à partir d'une API, ou à ses propres données utilisateur. Un jeton d'accès avec des portées personnalisées, souvent issu d'une autorisation d'informations d'identification du client M2M, autorise l'accès à un serveur de ressources. Pour plus d'informations, consultez [Utilisation du jeton d'accès](#).

Jeton d'actualisation

Déclaration chiffrée d'authentification initiale que votre application peut présenter à votre groupe d'utilisateurs lorsque les jetons de votre utilisateur expirent. Une demande de jeton d'actualisation renvoie de nouveaux jetons d'accès et d'identification non expirés. Pour plus d'informations, consultez [Utilisation du jeton actualisation](#).

Vous pouvez définir l'expiration de ces jetons pour chaque client d'application depuis l'onglet Intégration des applications de votre groupe d'utilisateurs dans la console [Amazon Cognito](#).

Termes du client d'application

Les termes suivants sont les propriétés disponibles des clients d'application dans la console Amazon Cognito.

URL de rappel autorisées

Une URL de rappel indique où l'utilisateur doit être redirigé lorsque la connexion aboutit. Choisissez au moins une URL de rappel. L'URL de rappel doit :

- Être un URI absolu.
- Être pré-enregistré auprès d'un client.
- Ne pas inclure un composant de fragment.

Veillez consulter [OAuth 2.0 - Point de terminaison de redirection](#).

Amazon Cognito exige HTTPS plutôt que HTTP, sauf pour `http://localhost` à des fins de test uniquement.

Les URL de rappel d'application comme `myapp://example` sont également prises en charge.

URL de déconnexion autorisées

Une URL de déconnexion indique où l'utilisateur doit être redirigé après la déconnexion.

Autorisations de lecture et d'écriture pour les attributs

Votre groupe d'utilisateurs peut compter de nombreux clients, chacun ayant son propre client d'application et IdPs. Vous pouvez configurer le client de votre application pour qu'il dispose d'un accès en lecture et en écriture uniquement aux attributs utilisateur pertinents pour l'application. Dans des cas tels que l'autorisation machine-to-machine (M2M), vous ne pouvez accorder l'accès à aucun de vos attributs utilisateur.

Considérations relatives à la configuration des autorisations de lecture et d'écriture des attributs

- Lorsque vous créez un client d'application et que vous ne personnalisez pas les autorisations de lecture et d'écriture des attributs, Amazon Cognito accorde des autorisations de lecture et d'écriture à tous les attributs du groupe d'utilisateurs.
- Vous pouvez accorder un accès en écriture à des [attributs personnalisés](#) immuables. Votre client d'application peut écrire des valeurs dans des attributs inaltérables lorsque vous créez ou inscrivez un utilisateur. Ensuite, vous ne pouvez pas écrire de valeurs dans des attributs personnalisés immuables pour l'utilisateur.

- Les clients de l'application doivent disposer d'un accès en écriture aux attributs requis dans votre groupe d'utilisateurs. La console Amazon Cognito définit automatiquement les attributs requis comme étant accessibles en écriture.
- Vous ne pouvez pas autoriser un client d'application à accéder en écriture à `email_verified` ou `phone_number_verified`. Un administrateur de groupe d'utilisateurs peut modifier ces valeurs. Un utilisateur ne peut modifier la valeur de ces attributs que par le biais de la [vérification des attributs](#).

Flux d'authentification

Les méthodes de connexion autorisées par le client de votre application. Votre application peut prendre en charge l'authentification par nom d'utilisateur et mot de passe, le mot de passe distant sécurisé (SRP), l'authentification personnalisée avec des déclencheurs Lambda et l'actualisation des jetons. Pour des raisons de sécurité, utilisez l'authentification SRP comme principale méthode de connexion. L'interface utilisateur hébergée connecte automatiquement les utilisateurs à SRP.

Portées personnalisées

Une règle personnalisée est une règle que vous définissez pour votre propre serveur de ressources dans le Serveur de ressources. Le format est *resource-server-identifier/scope*. veuillez consulter [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#).

URI de redirection par défaut

Remplace le `redirect_uri` paramètre dans les demandes d'authentification des utilisateurs par des tiers IdPs. Configurez ce paramètre du client d'application avec le `DefaultRedirectURI` paramètre d'une demande d'[UpdateUserPoolClientAPI](#) [CreateUserPoolClient](#) ou d'une demande d'API. Cette URL doit également être membre du client `CallbackURLs` de votre application. Amazon Cognito redirige les sessions authentifiées vers cette URL lorsque :

1. Un [fournisseur d'identité](#) est attribué à votre client d'application et plusieurs [URL de rappel](#) sont définies. Votre groupe d'utilisateurs redirige les demandes d'authentification vers le [serveur d'autorisation](#) vers l'URI de redirection par défaut lorsqu'elles ne contiennent aucun `redirect_uri` paramètre.
2. Un [fournisseur d'identité](#) est attribué à votre client d'application et une [URL de rappel](#) est définie. Dans ce scénario, il n'est pas nécessaire de définir une URL de rappel par défaut. Les demandes qui n'incluent pas de `redirect_uri` paramètre sont redirigées vers la seule URL de rappel disponible.

Fournisseurs d'identité

Vous pouvez choisir certains ou tous les fournisseurs d'identité externes de votre groupe d'utilisateurs (IdPs) pour authentifier vos utilisateurs. Le client d'application peut également n'authentifier que les utilisateurs locaux dans votre groupe d'utilisateurs. Lorsque vous ajoutez un IdP à votre client d'application, vous pouvez générer des liens d'autorisation vers l'IdP et l'afficher sur votre page de connexion de l'interface utilisateur hébergée. Vous pouvez en attribuer plusieurs IdPs, mais vous devez en attribuer au moins un. Pour plus d'informations sur l'utilisation de l'interface externe IdPs, consultez [Ajout de la connexion du groupe d'utilisateurs via un tiers](#).

Portées OpenID Connect

Choisissez un ou plusieurs des paramètres scope OAuth suivants pour spécifier les privilèges d'accès qui peuvent être demandés pour les jetons d'accès.

- La portée `openid` déclare que vous souhaitez récupérer un jeton d'identification et l'identifiant unique d'un utilisateur. Il demande également la totalité ou une partie des attributs utilisateur, en fonction des portées supplémentaires de la demande. Amazon Cognito ne renvoie pas de jeton d'identification à moins que vous ne demandiez la portée `openid`. La portée `openid` autorise les demandes de jetons d'identification structurels, telles que l'expiration et l'identifiant de clé, et détermine les attributs utilisateur que vous recevez dans une réponse du [Point de terminaison UserInfo](#).
- Si `openid` est la seule portée que vous demandez, Amazon Cognito remplit le jeton d'identification avec tous les attributs utilisateur que le client d'application actuel peut lire. La réponse `userInfo` à un jeton d'accès ayant cette portée seule renvoie tous les attributs utilisateur.
- Lorsque vous demandez `openid` avec d'autres portées, telles que `phone`, `email` ou `profile`, le jeton d'identification et `userInfo` renvoient l'identifiant unique de l'utilisateur et les attributs définis par les portées supplémentaires.
- Le paramètre scope `phone` accorde l'accès aux réclamations `phone_number` et `phone_number_verified`. Cette portée peut uniquement être demandée avec le paramètre scope `openid`.
- Le paramètre scope `email` accorde l'accès aux réclamations `email` et `email_verified`. Cette portée peut uniquement être demandée avec le paramètre scope `openid`.
- Le `aws.cognito.signin.user.admin` champ d'application autorise l'accès aux [opérations de l'API des groupes d'utilisateurs Amazon Cognito](#) qui nécessitent des jetons d'accès, tels que [UpdateUserAttributes](#) et [VerifyUserAttribute](#).

- Le paramètre `scope profile` accorde l'accès à tous les attributs utilisateurs pouvant être lus par le client. Cette portée peut uniquement être demandée avec le paramètre `scope openid`.

Pour plus d'informations sur les portées, consultez la liste des [portées OIDC standard](#).

Types d'autorisations OAuth

Une autorisation OAuth est une méthode d'authentification qui récupère les jetons du groupe d'utilisateurs. Amazon Cognito prend en charge les types suivants d'autorisations. Pour intégrer ces autorisations OAuth dans votre application, vous devez ajouter un domaine à votre groupe d'utilisateurs.

Octroi de code d'autorisation

L'octroi du code d'autorisation génère un code que votre application peut échanger contre des jetons du groupe d'utilisateurs avec le [Point de terminaison de jeton](#). Lorsque vous échangez un code d'autorisation, votre application reçoit des jetons d'ID, d'accès et d'actualisation. Ce flux OAuth, comme l'autorisation implicite, se produit dans les navigateurs de vos utilisateurs. L'attribution d'un code d'autorisation est l'autorisation la plus sécurisée proposée par Amazon Cognito, car les jetons ne sont pas visibles dans les sessions de vos utilisateurs. Au lieu de cela, votre application génère la demande qui renvoie des jetons et peut les mettre en cache dans un espace de stockage protégé. Pour plus d'informations, consultez le Code d'autorisation dans la norme [IETF RFC 6749 #1.3.1](#).

Note

La bonne pratique de sécurité dans les applications public-client consiste à n'activer que le flux d'autorisation OAuth autorisation-code, et à mettre en œuvre Proof Key for Code Exchange (PKCE) pour restreindre l'échange de jetons. Avec PKCE, un client ne peut échanger un code d'autorisation que s'il a fourni au point de terminaison du jeton le même secret que celui présenté dans la demande d'authentification d'origine. Pour plus d'informations sur PKCE, consultez [IETF RFC 7636](#).

Octroi implicite

L'autorisation implicite fournit un jeton d'accès et d'ID, mais pas un jeton d'actualisation, à la session de navigation de votre utilisateur directement depuis le [Point de terminaison d'autorisation](#). Une autorisation implicite supprime l'obligation d'envoyer une demande séparée

au point de terminaison du jeton, mais n'est pas compatible avec PKCE et ne renvoie pas de jetons d'actualisation. Cette autorisation prend en charge les scénarios de test et l'architecture d'applications qui ne peuvent pas compléter les autorisations autorisation-code. Pour en savoir plus, consultez Autorisation implicite dans [IETF RFC 6749 #1.3.2](#). Vous pouvez activer à la fois l'autorisation autorisation-code et l'autorisation implicite dans un client d'application, puis utiliser chaque autorisation selon vos besoins.

Autorisation d'informations d'identification du client

Les informations d'identification du client sont accordées pour les communications machine-to-machine (M2M). Les autorisations autorisation-code et implicite émettent des jetons à des utilisateurs humains authentifiés. Les informations d'identification du client accordent une autorisation basée sur la portée pour une API à partir d'un système non interactif. Votre application peut demander les informations d'identification du client directement depuis le point de terminaison du jeton et recevoir un jeton d'accès. Pour plus d'informations, consultez Informations d'identification du client dans [IETF RFC 6749 #1.3.4](#). Vous ne pouvez activer les autorisations client-informations d'identification que dans les clients d'application qui ont un secret client et qui ne prennent pas en charge les autorisations autorisation-code ou implicites.

Note

Comme vous n'invoquez pas le flux d'informations d'identification du client en tant qu'utilisateur, cette autorisation peut uniquement ajouter des portées personnalisées aux jetons d'accès. Une règle personnalisée est une règle que vous définissez pour votre propre serveur de ressources. Les portées par défaut comme `openid` et `profile` ne s'appliquent pas aux utilisateurs non humains.

Les jetons d'ID étant une validation des attributs utilisateur, ils ne sont pas pertinents pour les communications M2M, et les autorisations d'informations d'identification du client ne les délivrent pas. veuillez consulter [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#).

Les informations d'identification des clients vous permettent d'ajouter des frais à votre AWS facture. Pour plus d'informations, consultez [Tarification d'Amazon Cognito](#).

Création d'un client d'application

AWS Management Console

Pour créer un client d'application (console)

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou créez-en un.
4. Sélectionnez l'onglet Intégration d'applications.
5. Sous Clients d'application, sélectionnez Créer un client d'application.
6. Sélectionnez un type d'application : Client public, Client confidentiel ou Autre.
7. Saisissez un nom de client d'application.
8. Choisissez Générer un secret de client pour qu'Amazon Cognito génère un secret client pour vous. Les secrets des clients sont généralement associés à des clients confidentiels.
9. Sélectionnez le flux d'authentification que vous souhaitez autoriser dans votre client d'application.
10. Configuration de Durée de session du flux d'authentification. Il s'agit du temps dont disposent vos utilisateurs pour terminer chaque défi d'authentification avant l'expiration de leur jeton de session.
11. (Facultatif) Effectuez les opérations suivantes si vous souhaitez configurer l'expiration du jeton :
 - a. Spécifiez le délai d'expiration du jeton d'actualisation de l'application. La valeur par défaut est de 30 jours. Vous pouvez le remplacer par toute valeur comprise entre 1 heure et 10 ans.
 - b. Spécifiez le délai d'expiration du jeton d'accès du client d'application. La valeur par défaut est 1 heure. Vous pouvez la remplacer par toute valeur comprise entre 5 minutes et 24 heures.
 - c. Spécifiez le délai d'expiration du jeton d'identification du client d'application. La valeur par défaut est 1 heure. Vous pouvez la remplacer par toute valeur comprise entre 5 minutes et 24 heures.

⚠ Important

Si vous utilisez l'interface utilisateur hébergée et configurez des jetons de moins d'une heure, l'utilisateur final sera en mesure d'utiliser des jetons basés sur le cookie de sa session qui est actuellement fixée à une heure.

12. Indiquez si vous voulez activer la révocation de jetons pour ce client d'application. Cela augmentera la taille des jetons émis par Amazon Cognito.
13. Indiquez si vous voulez empêcher les erreurs d'existence d'utilisateur pour ce client d'application. Amazon Cognito répondra aux demandes de connexion des utilisateurs inexistantes avec un message générique indiquant que le nom d'utilisateur ou le mot de passe était incorrect.
14. Si vous souhaitez utiliser l'interface utilisateur hébergée avec ce client d'application, configurez Paramètres de l'interface utilisateur hébergée.
 - a. Saisissez une ou plusieurs URL de rappel autorisées. Il s'agit des URL web ou d'applications vers lesquelles vous souhaitez qu'Amazon Cognito redirige vos utilisateurs une fois leur authentification terminée.
 - b. Saisissez une ou plusieurs URL de déconnexion autorisées. Il s'agit des URL que vous souhaitez que votre application accepte dans les demandes adressées à [Point de terminaison de déconnexion](#).
 - c. Choisissez un ou plusieurs Fournisseurs d'identité que vous souhaitez autoriser à inscrire des utilisateurs pour votre application. Vous pouvez choisir n'importe quelle combinaison d'éléments existants IdPs. Vous pouvez authentifier les utilisateurs uniquement auprès de votre groupe d'utilisateurs ou auprès d'un ou de plusieurs tiers IdPs que vous avez configurés dans votre groupe d'utilisateurs.
 - d. Choisissez les types d'octroi OAuth 2.0 que vous souhaitez que votre client d'application accepte.
 - Sélectionnez Octroi de code d'autorisation pour transmettre à votre application des codes qu'elle pourra échanger contre des jetons avec [Point de terminaison de jeton](#).
 - Sélectionnez Octroi implicite pour transmettre l'ID et les jetons d'accès directement à votre application. Le flux d'octroi implicite expose les jetons directement à vos utilisateurs.

- Sélectionnez Informations d'identification client pour transmettre des jetons d'accès à votre application en fonction de sa connaissance non pas des informations d'identification de l'utilisateur, mais du secret du client. Le flux d'octroi des informations d'identification du client s'exclut mutuellement du code d'autorisation et des flux d'octroi implicites.
 - e. Sélectionnez les Portées OpenID Connect que vous souhaitez autoriser pour ce client d'application. Vous pouvez générer des jetons d'accès avec l'étendue `aws.cognito.signin.user.admin` via l'API des groupes d'utilisateurs uniquement. Pour des portées supplémentaires, vous devez demander vos jetons d'accès auprès de [Point de terminaison de jeton](#).
 - f. Sélectionnez les Portées personnalisées que vous souhaitez autoriser pour ce client d'application. Les portées personnalisées sont le plus souvent utilisées pour autoriser l'accès à des API tierces.
15. Configurez Autorisations de lecture et d'écriture pour les attributs pour ce client d'application. Votre client d'application peut être autorisé à lire et à écrire tout ou un sous-ensemble limité du schéma d'attribut de votre groupe d'utilisateurs.
 16. Choisissez Créer un client d'application.
 17. Notez l'ID du client. Cela permettra d'identifier le client de l'application lors des demandes d'inscription et de connexion.

AWS CLI

```
aws cognito-idp create-user-pool-client --user-pool-id MyUserPoolID --client-name myApp
```

Note

Utilisez le format JSON pour les URL de rappel et de déconnexion afin d'empêcher la CLI de les traiter comme des fichiers de paramètre à distance :

```
--callback-urls "[\"https://example.com\"]"  
--logout-urls "[\"https://example.com\"]"
```

Consultez la référence des AWS CLI commandes pour plus d'informations : [create-user-pool-client](#)

Amazon Cognito user pools API

Générez une demande de [CreateUserPoolClient](#) API. Vous devez spécifier une valeur pour tous les paramètres auxquels vous ne voulez pas attribuer une valeur par défaut.

Mettre à jour un client d'application de groupe d'utilisateurs (AWS CLI et une AWS API)

Au niveau de AWS CLI, entrez la commande suivante :

```
aws cognito-idp update-user-pool-client --user-pool-id "MyUserPoolID" --client-id
"MyAppClientID" --allowed-o-auth-flows-user-pool-client --allowed-o-auth-flows "code"
"implicit" --allowed-o-auth-scopes "openid" --callback-urls ["https://example.com"]
--supported-identity-providers ["MySAMLIdP", "LoginWithAmazon"]
```

Si la commande aboutit, elle AWS CLI renvoie une confirmation :

```
{
  "UserPoolClient": {
    "ClientId": "MyClientID",
    "SupportedIdentityProviders": [
      "LoginWithAmazon",
      "MySAMLIdP"
    ],
    "CallbackURLs": [
      "https://example.com"
    ],
    "AllowedOAuthScopes": [
      "openid"
    ],
    "ClientName": "Example",
    "AllowedOAuthFlows": [
      "implicit",
      "code"
    ],
    "RefreshTokenValidity": 30,
    "AuthSessionValidity": 3,
    "CreationDate": 1524628110.29,
    "AllowedOAuthFlowsUserPoolClient": true,
    "UserPoolId": "MyUserPoolID",
    "LastModifiedDate": 1530055177.553
  }
}
```



```
}  
}
```

Consultez la référence des AWS CLI commandes pour plus d'informations : [update-user-pool-client](#).

AWS API : [UpdateUserPoolClient](#)

Obtenir des informations sur un client d'application de groupe d'utilisateurs (AWS CLI et une AWS API)

```
aws cognito-idp describe-user-pool-client --user-pool-id MyUserPoolID --client-id MyClientID
```

Consultez la référence des AWS CLI commandes pour plus d'informations : [describe-user-pool-client](#).

AWS API : [DescribeUserPoolClient](#)

Répertorier toutes les informations du client de l'application dans un pool d'utilisateurs (AWS CLI et une AWS API)

```
aws cognito-idp list-user-pool-clients --user-pool-id "MyUserPoolID" --max-results 3
```

Consultez la référence des AWS CLI commandes pour plus d'informations : [list-user-pool-clients](#).

AWS API : [ListUserPoolClients](#)

Suppression d'un client d'application de groupe d'utilisateurs (AWS CLI et AWS d'une API)

```
aws cognito-idp delete-user-pool-client --user-pool-id "MyUserPoolID" --client-id "MyAppClientID"
```

Consultez la référence des AWS CLI commandes pour plus d'informations : [delete-user-pool-client](#)

AWS API : [DeleteUserPoolClient](#)

Utilisation d'appareils utilisateur dans votre groupe d'utilisateurs

Lorsque vous connectez les utilisateurs du groupe d'utilisateurs local à l'aide de l'API des groupes d'utilisateurs Amazon Cognito, vous pouvez associer les journaux d'activité de vos utilisateurs à partir des [fonctionnalités de sécurité avancée](#) à chacun de leurs appareils et, éventuellement,

autoriser vos utilisateurs à passer l'authentification multifactorielle (MFA) à condition qu'ils utilisent un appareil approuvé. Amazon Cognito inclut une clé d'appareil dans la réponse à toute connexion qui ne comporte pas déjà d'informations d'appareil. La clé d'appareil est au format *Region_UUID*. Avec une clé d'appareil, une bibliothèque de mots de passe distants sécurisés (SRP, Secure Remote Password) et un pool d'utilisateurs qui permet l'authentification d'appareil, vous pouvez inviter les utilisateurs de votre application à faire confiance à l'appareil actuel et à ne plus leur demander de code MFA lors de la connexion.

Rubriques

- [Configuration des appareils mémorisés](#)
- [Obtention d'une clé d'appareil](#)
- [Connexion avec un appareil](#)
- [Affichage, mise à jour et oubli des appareils](#)

Configuration des appareils mémorisés

Avec les groupes d'utilisateurs Amazon Cognito, vous pouvez associer les appareils de chacun de vos utilisateurs à un identifiant d'appareil unique : la clé d'appareil. Lorsque vous présentez la clé d'appareil et que vous procédez une authentification d'appareil au moment de la connexion, vous pouvez profiter des avantages de deux fonctionnalités.

1. Avec les fonctionnalités de sécurité avancée, vous pouvez surveiller l'activité des utilisateurs sur des appareils spécifiques à des fins de sécurité et d'analyse. Lorsque les utilisateurs se connectent, votre application a la possibilité d'authentifier chaque utilisateur et son appareil, en ajoutant les informations d'appareil à leurs journaux d'activité.
2. La mémorisation des appareils prend également en charge un flux d'authentification d'appareil approuvé, qui permet à vos utilisateurs de se connecter sans authentification MFA sur une période adaptée aux exigences de sécurité de votre application. Lorsque vous voulez réinviter utilisateur à soumettre un code MFA, vous pouvez modifier l'état mémorisé de son appareil.

Les appareils mémorisés peuvent passer outre l'authentification MFA uniquement dans les groupes d'utilisateurs où l'authentification MFA est active.

Lorsque votre utilisateur se connecte avec un appareil mémorisé, vous devez procéder à une authentification d'appareil supplémentaire durant son flux d'authentification. Pour de plus amples informations, veuillez consulter [Connexion avec un appareil](#).

Configurez votre groupe d'utilisateurs de sorte qu'il mémorise les appareils dans l'onglet Expérience de connexion de votre groupe d'utilisateurs, sous Suivi de l'appareil. Lorsque vous configurez la fonctionnalité de mémorisation des appareils via la console Amazon Cognito, vous disposez de trois options : Toujours, Acceptation de l'utilisateur et Non.

Ne pas mémoriser

Votre groupe d'utilisateurs n'invite pas les utilisateurs à mémoriser les appareils lorsqu'ils se connectent.

Toujours mémoriser

Lorsque votre application confirme l'appareil d'un utilisateur, votre groupe d'utilisateurs mémorise toujours l'appareil et ne renvoie pas de questions de sécurité MFA lors des futures connexions réussies de l'appareil.

Consentement de l'utilisateur

Lorsque votre application confirme l'appareil d'un utilisateur, votre groupe d'utilisateurs ne supprime pas automatiquement les questions de sécurité MFA. Vous devez inviter votre utilisateur à se décider quant à la mémorisation de l'appareil.

Lorsque vous sélectionnez Toujours mémoriser ou Consentement de l'utilisateur, Amazon Cognito génère une clé et un secret d'identification d'appareil chaque fois qu'un utilisateur se connecte depuis un appareil non identifié. La clé d'appareil est l'identifiant initial que votre application envoie à votre groupe d'utilisateurs lorsque votre utilisateur procède à l'authentification de l'appareil.

À chaque appareil utilisateur confirmé, que sa mémorisation soit automatique ou soumise au consentement de l'utilisateur, vous pouvez utiliser la clé et le secret d'identification d'appareil pour authentifier un appareil à chaque connexion utilisateur.

Vous pouvez également configurer les paramètres des appareils mémorisés pour votre groupe d'utilisateurs dans une demande d'API [CreateUserPool](#) ou [UpdateUserPool](#). Pour en savoir plus, consultez la propriété [DeviceConfiguration](#).

L'API des groupes d'utilisateurs Amazon Cognito propose d'autres opérations pour les appareils mémorisés.

1. [ListDevices](#) et [AdminListDevices](#) renvoient la liste des clés d'appareils et leurs métadonnées pour un utilisateur.

2. [GetDevice](#) et [AdminGetDevice](#) renvoie la clé et les métadonnées d'un seul appareil.
3. [UpdateDeviceStatus](#) et [AdminUpdateDeviceStatus](#) définissent l'appareil d'un utilisateur comme étant mémorisé ou non mémorisé.
4. [ForgetDevice](#) et [AdminForgetDevice](#) suppriment l'appareil confirmé d'un utilisateur de son profil.

Les opérations d'API dont le nom commence par Admin sont destinées à être utilisées dans des applications côté serveur et doivent être autorisées avec des informations d'identification IAM. Pour de plus amples informations, veuillez consulter [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#).

Obtention d'une clé d'appareil

Chaque fois que votre utilisateur se connecte avec l'API des groupes d'utilisateurs et n'inclut pas de clé d'appareil dans les paramètres d'authentification en tant que DEVICE_KEY, Amazon Cognito renvoie une nouvelle clé d'appareil dans la réponse. Dans votre application publique côté client, placez la clé d'appareil dans le stockage de l'application afin de pouvoir l'inclure dans les futures demandes. Dans votre application confidentielle côté serveur, définissez un cookie de navigateur ou un autre jeton côté client avec la clé d'appareil de votre utilisateur.

Pour que votre utilisateur puisse se connecter avec son appareil approuvé, votre application doit d'abord confirmer la clé de l'appareil et fournir des informations supplémentaires. Générez une demande [ConfirmDevice](#) à destination d'Amazon Cognito qui confirme l'appareil de votre utilisateur avec la clé de l'appareil, un nom convivial, un vérificateur de mot de passe et une valeur salt. Si vous avez configuré votre groupe d'utilisateurs pour l'authentification d'appareil par consentement, Amazon Cognito répond à votre demande [ConfirmDevice](#) par un message qui invite votre utilisateur à décider si l'appareil actuel doit être mémorisé. Répondez par la sélection de votre utilisateur dans une demande [UpdateDeviceStatus](#).

Lorsque vous confirmez l'appareil de votre utilisateur mais que vous ne le définissez pas comme étant mémorisé, Amazon Cognito enregistre l'association mais procède à une connexion non basée sur l'appareil lorsque vous fournissez la clé de l'appareil. Les appareils peuvent générer des journaux utiles pour la sécurité des utilisateurs et la résolution des problèmes. Un appareil confirmé mais non mémorisé ne bénéficie pas de la fonctionnalité de connexion, mais de la fonctionnalité de journaux de surveillance de la sécurité. Lorsque vous activez les fonctionnalités de sécurité avancée pour votre client d'application et que vous encodez l'empreinte d'un appareil dans votre demande, Amazon Cognito associe les événements utilisateur à l'appareil confirmé.

Pour obtenir une nouvelle clé d'appareil

1. Démarrez la session de connexion de votre utilisateur avec une demande d'API [InitiateAuth](#).
2. Répondez à toutes les questions de sécurité d'authentification avec [RespondToAuthChallenge](#) jusqu'à ce que vous receviez des jetons web JSON (JWT) qui marquent la fin de la session de connexion de votre utilisateur.
3. Dans votre application, enregistrez les valeurs renvoyées par Amazon Cognito dans `NewDeviceMetadata` dans sa réponse `RespondToAuthChallenge` ou `InitiateAuth` : `DeviceGroupKey` et `DeviceKey`.
4. Générez un nouveau secret SRP pour votre utilisateur : une valeur salt et un vérificateur de mot de passe. Cette fonction est disponible dans les kits SDK qui fournissent des bibliothèques SRP.
5. Invitez votre utilisateur à saisir un nom d'appareil, ou générez-en un à partir des caractéristiques de l'appareil de votre utilisateur.
6. Fournissez le jeton d'accès, la clé de l'appareil, le nom de l'appareil et le secret SRP de votre utilisateur dans une demande d'API [ConfirmDevice](#). Si votre groupe d'utilisateurs est configuré pour Toujours mémoriser les appareils, l'enregistrement de votre utilisateur est terminé.
7. Si Amazon Cognito a répondu à `ConfirmDevice` par `"UserConfirmationNecessary": true`, invitez votre utilisateur à décider si l'appareil doit être mémorisé. S'il affirme vouloir mémoriser l'appareil, générez une demande d'API [UpdateDeviceStatus](#) avec le jeton d'accès de votre utilisateur, la clé de l'appareil et `"DeviceRememberedStatus": "remembered"`.
8. Si vous avez donné instruction à Amazon Cognito de mémoriser l'appareil, à la prochaine connexion, une question de sécurité `DEVICE_SRP_AUTH` est présentée à l'utilisateur au lieu d'une question de sécurité MFA.

Connexion avec un appareil

Une fois que vous avez configuré la mémorisation de l'appareil d'un utilisateur, Amazon Cognito ne lui demande plus de soumettre de code MFA lorsqu'il se connecte avec la même clé d'appareil. L'authentification d'appareil remplace uniquement la question de sécurité d'authentification MFA par une question de sécurité d'authentification d'appareil. Vous ne pouvez pas connecter les utilisateurs uniquement avec l'authentification d'appareil. Votre utilisateur doit d'abord s'authentifier avec son mot de passe ou une question de sécurité personnalisée. Voici le processus d'authentification pour un utilisateur doté d'un appareil mémorisé.

Pour effectuer une authentification d'appareil dans un flux qui utilise des [déclencheurs Lambda de question de sécurité d'authentification personnalisée](#), transmettez un paramètre `DEVICE_KEY` dans

vosre demande d'API [InitiateAuth](#). Une fois que votre utilisateur a bien répondu à toutes les questions de sécurité et que CUSTOM_CHALLENGE renvoie un valeur true pour issueTokens, Amazon Cognito renvoie une dernière question de sécurité DEVICE_SRP_AUTH.

Pour vous connecter avec un appareil

1. Récupérez la clé de l'appareil de votre utilisateur depuis le stockage client.
2. Démarrez la session de connexion de votre utilisateur avec une demande d'API [InitiateAuth](#). Attribuez à AuthFlow la valeur USER_SRP_AUTH, REFRESH_TOKEN_AUTH, USER_PASSWORD_AUTH ou CUSTOM_AUTH. Dans AuthParameters, ajoutez la clé d'appareil de votre utilisateur au paramètre DEVICE_KEY et incluez les autres paramètres obligatoires au flux de connexion que vous avez sélectionné.
 - a. Vous pouvez également transmettre DEVICE_KEY dans les paramètres d'une réponse PASSWORD_VERIFIER à une question de sécurité d'authentification.
3. Répondez aux questions de sécurité jusqu'à ce que vous receviez une question de sécurité DEVICE_SRP_AUTH dans la réponse.
4. Dans une demande d'API [RespondToAuthChallenge](#), envoyez un ChallengeName avec la valeur DEVICE_SRP_AUTH et des paramètres pour USERNAME, DEVICE_KEY et SRP_A.
5. Amazon Cognito répond par une question de sécurité DEVICE_PASSWORD_VERIFIER. La réponse à cette question de sécurité comporte des valeurs pour SECRET_BLOCK et SRP_B.
6. Avec votre bibliothèque SRP, générez et soumettez des paramètres PASSWORD_CLAIM_SIGNATURE, PASSWORD_CLAIM_SECRET_BLOCK, TIMESTAMP, USERNAME et DEVICE_KEY. Soumettez-les dans une demande RespondToAuthChallenge supplémentaire.
7. Complétez les autres questions de sécurité jusqu'à ce que vous receviez les jetons JWT de l'utilisateur.

Le pseudocode suivant montre comment calculer les valeurs de votre réponse à la question de sécurité DEVICE_PASSWORD_VERIFIER.

```
PASSWORD_CLAIM_SECRET_BLOCK = SECRET_BLOCK
TIMESTAMP = Tue Sep 25 00:09:40 UTC 2018
PASSWORD_CLAIM_SIGNATURE = Base64(SHA256_HMAC(K_USER, DeviceGroupKey + DeviceKey +
  PASSWORD_CLAIM_SECRET_BLOCK + TIMESTAMP))
K_USER = SHA256_HASH(S_USER)
S_USER = (SRP_B - k * gx)(a + ux)
x = SHA256_HASH(salt + FULL_PASSWORD)
u = SHA256_HASH(SRP_A + SRP_B)
```

```
k = SHA256_HASH(N + g)
```

Affichage, mise à jour et oubli des appareils

Vous pouvez implémenter les fonctionnalités suivantes dans votre application avec l'API Amazon Cognito.

1. Afficher les informations sur l'appareil actuel d'un utilisateur.
2. Afficher la liste de tous les appareils de votre utilisateur.
3. Oublier un appareil.
4. Mettre à jour l'état mémorisé d'un appareil.

Les jetons d'accès qui autorisent les demandes d'API incluses dans les descriptions suivantes doivent inclure la portée `aws.cognito.signin.user.admin`. Amazon Cognito ajoute une demande pour cette portée à tous les jetons d'accès que vous générez avec l'API des groupes d'utilisateurs Amazon Cognito. Les IdP tiers doivent gérer séparément les appareils et l'authentification MFA pour leurs utilisateurs qui s'authentifient auprès d'Amazon Cognito. Dans l'interface utilisateur hébergée, vous pouvez demander la portée `aws.cognito.signin.user.admin`, mais l'interface utilisateur hébergée ajoute automatiquement les informations d'appareils aux journaux utilisateur de sécurité avancée et ne propose pas de mémoriser les appareils.

Affichage les informations sur un appareil

Vous pouvez demander les informations sur l'appareil d'un utilisateur pour déterminer s'il est toujours utilisé. Par exemple, vous pouvez souhaiter désactiver les appareils mémorisés qui n'ont pas été connectés depuis plus de 90 jours.

- Pour afficher les informations de l'appareil de votre utilisateur dans une application cliente publique, soumettez la clé d'accès et la clé de l'appareil de votre utilisateur dans une demande d'API [GetDevice](#).
- Pour afficher les informations de l'appareil de votre utilisateur dans une application cliente confidentielle, signez une demande d'API [AdminGetDevice](#) avec des informations d'identification AWS et soumettez le nom d'utilisateur, la clé de l'appareil et le groupe d'utilisateurs de votre utilisateur.

Affichage de la liste de tous les appareils de votre utilisateur

Vous pouvez afficher la liste de tous les appareils de votre utilisateur ainsi que leurs propriétés. Par exemple, vous pouvez souhaiter vérifier que l'appareil actuel correspond à un appareil mémorisé.

- Dans une application cliente publique, soumettez le jeton d'accès de votre utilisateur dans une demande d'API [ListDevices](#).
- Dans une application cliente confidentielle, signez une demande d'API [AdminGetDevice](#) avec des informations d'identification AWS et soumettez le nom d'utilisateur et le groupe d'utilisateurs de votre utilisateur.

Oubli d'un appareil

Vous pouvez supprimer la clé de l'appareil d'un utilisateur. Vous pouvez souhaiter le faire si vous constatez que votre utilisateur n'utilise plus un appareil ou si vous détectez une activité inhabituelle et souhaitez inviter un utilisateur à procéder à une nouvelle authentification MFA. Pour réenregistrer l'appareil ultérieurement, vous devez générer et stocker une nouvelle clé d'appareil.


- Dans une application cliente publique, soumettez la clé de l'appareil et le jeton d'accès de votre utilisateur dans la demande d'API [ForgetDevice](#).
- Dans une application cliente confidentielle, soumettez la clé de l'appareil et le jeton d'accès de votre utilisateur dans la demande d'API [AdminForgetDevice](#).

Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito

Si vous voulez inscrire, connecter et gérer les utilisateurs de votre groupe d'utilisateurs, deux options s'offrent à vous.

1. Les points de terminaison d'un groupe d'utilisateurs incluent l'[interface utilisateur hébergée](#) et les [points de terminaison de fédération](#). Ils constituent un ensemble de pages Web publiques qu'Amazon Cognito active lorsque vous [choisissez un domaine](#) pour votre groupe d'utilisateurs. Pour démarrer rapidement avec les fonctionnalités d'authentification et d'autorisation des groupes d'utilisateurs Amazon Cognito, notamment les pages d'inscription, de connexion, de gestion des mots de passe et d'authentification multifactorielle (MFA), utilisez l'interface utilisateur intégrée de l'interface utilisateur hébergée. Les autres points de terminaison d'un groupe d'utilisateurs facilitent l'authentification auprès de fournisseurs d'identité (IdPs) tiers. Les services qu'ils fournissent incluent ce qui suit.

- a. Des points de terminaison de rappel des fournisseurs de services pour les champs standard authentifiés provenant de vos fournisseurs d'identité, tels que `saml2/idpresponse` et `oauth2/idpresponse`. Lorsqu'Amazon Cognito est un fournisseur de services intermédiaire (SP) entre votre application et votre fournisseur d'identité, les points de terminaison de rappel représentent le service.
 - b. Des points de terminaison qui fournissent des informations sur votre environnement, comme `oauth2/userInfo` et `jwtks.json`. Votre application utilise ces points de terminaison lorsqu'elle vérifie les jetons ou récupère les données de profil utilisateur à l'aide de kits SDK AWS et des bibliothèques OAuth 2.0.
2. L'[API des groupes d'utilisateurs Amazon Cognito](#) est un ensemble d'outils qui permet à votre application Web ou mobile de collecter des informations de connexion sur votre propre front-end personnalisé pour authentifier les utilisateurs. L'authentification par API des groupes d'utilisateurs produit les jetons Web JSON suivants.
- a. Un jeton d'identité avec des champs standard d'attribut vérifiables provenant de votre utilisateur.
 - b. Un jeton d'accès qui autorise votre utilisateur à créer des demandes d'API autorisées par jeton sur un [point de terminaison de service AWS](#).

 Note

Par défaut, les jetons d'accès issus de l'authentification par API des groupes d'utilisateurs contiennent uniquement la portée `aws.cognito.signin.user.admin`. Pour générer un jeton d'accès avec des portées supplémentaires, par exemple pour autoriser une demande à une API tierce, demandez des portées pendant l'authentification via les points de terminaison de votre groupe d'utilisateurs ou ajoutez des portées personnalisées dans un [Déclencheur Lambda avant génération de jeton](#). La personnalisation des jetons d'accès augmente les coûts sur votre facture AWS.

Vous pouvez lier un utilisateur fédéré, qui se connecterait normalement via les points de terminaison des groupes d'utilisateurs, à un utilisateur dont le profil est local à votre groupe d'utilisateurs. Un utilisateur local existe exclusivement dans l'annuaire de votre groupe d'utilisateurs sans fédération via un fournisseur d'identité externe. Si vous liez son identité fédérée à un utilisateur local dans une demande d'API [AdminLinkProviderForUser](#), il peut se connecter avec l'API des groupes d'utilisateurs. Pour de plus amples informations, veuillez consulter [Liaison d'utilisateurs fédérés à un profil utilisateur existant](#).

L'API des groupes d'utilisateurs Amazon Cognito est à double usage. D'une part, elle permet de créer et de configurer les ressources de vos groupes d'utilisateurs Amazon Cognito. Par exemple, vous pouvez créer des groupes d'utilisateurs, ajouter des déclencheurs AWS Lambda et configurer le domaine de votre interface utilisateur hébergée. D'autre part, l'API des groupes d'utilisateurs assure l'inscription, la connexion et d'autres opérations utilisateur pour les utilisateurs locaux et liés.

Exemple de scénario avec l'API des groupes d'utilisateurs Amazon Cognito

1. Votre utilisateur sélectionne un bouton « Create an account » (Créer un compte) que vous avez créé dans votre application. Il saisit une adresse e-mail et un mot de passe.
2. Votre application envoie une demande d'API [SignUp](#) et crée un nouvel utilisateur dans votre groupe d'utilisateurs.
3. Votre application demande à votre utilisateur un code de confirmation par e-mail. L'utilisateur saisit le code qu'il a reçu dans un e-mail.
4. Votre application envoie une demande d'API [ConfirmSignup](#) avec le code de confirmation de l'utilisateur.
5. Votre application demande à votre utilisateur son nom d'utilisateur et son mot de passe et saisit ces informations.
6. Votre application envoie une demande d'API [InitiateAuth](#) et stocke un jeton d'identification, un jeton d'accès et un jeton d'actualisation. Votre application appelle les bibliothèques OIDC pour gérer les jetons de l'utilisateur et faire persister la session de cet utilisateur.

Dans l'API des groupes d'utilisateurs Amazon Cognito, vous ne pouvez pas connecter les utilisateurs fédérés via un fournisseur d'identité. Vous devez authentifier ces utilisateurs via les points de terminaison de votre groupe d'utilisateurs. Pour en savoir plus sur les points de terminaison d'un groupe d'utilisateurs qui incluent l'interface utilisateur hébergée, consultez [Référence de l'interface utilisateur hébergée et des points de terminaison de la fédération des groupes d'utilisateurs](#). Vos utilisateurs fédérés peuvent partir de l'interface utilisateur hébergée et sélectionner leur fournisseur d'identité. Vous pouvez également ignorer l'interface utilisateur hébergée et diriger vos utilisateurs directement vers votre fournisseur d'identité pour qu'ils se connectent. Lorsque votre demande d'API à destination du [Point de terminaison d'autorisation](#) comporte un paramètre de fournisseur d'identité, Amazon Cognito redirige discrètement votre utilisateur vers la page de connexion du fournisseur d'identité.

Exemple de scénario avec les points de terminaison d'un groupe d'utilisateurs

1. Votre utilisateur sélectionne un bouton « Create an account » (Créer un compte) que vous avez créé dans votre application.
2. Vous présentez à votre utilisateur une liste des fournisseurs d'identité sociaux auprès desquels vous avez enregistré des informations d'identification de développeur. Votre utilisateur choisit Apple.
3. Votre application adresse une demande au [Point de terminaison d'autorisation](#) avec le nom de fournisseur SignInWithApple.
4. Le navigateur de votre utilisateur ouvre la page d'autorisation OAuth Apple. Votre utilisateur choisit d'autoriser Amazon Cognito à lire les informations de son profil.
5. Amazon Cognito confirme le jeton d'accès Apple et interroge le profil Apple de votre utilisateur.
6. Votre utilisateur présente un code d'autorisation Amazon Cognito à votre application.
7. Votre application échange le code d'autorisation avec le [Point de terminaison de jeton](#) et stocke un jeton d'identification, un jeton d'accès et un jeton d'actualisation. Votre application appelle les bibliothèques OIDC pour gérer les jetons de l'utilisateur et faire persister la session de cet utilisateur.

L'API des groupes d'utilisateurs et les points de terminaison d'un groupe d'utilisateurs prennent en charge divers scénarios, décrits dans ce guide. Les sections suivantes examinent la façon dont l'API des groupes d'utilisateurs se divise en différentes classes qui répondent à vos exigences en matière d'inscription, de connexion et de gestion des ressources.

Opérations d'API authentifiées et non authentifiées des groupes d'utilisateurs Amazon Cognito

L'API des groupes d'utilisateurs Amazon Cognito, qui est à la fois une interface de gestion des ressources et une interface d'authentification et d'autorisation destinée à l'utilisateur, combine les modèles d'autorisation suivants dans ses opérations. En fonction de l'opération de l'API, vous serez peut-être amené à fournir une autorisation avec des informations d'identification IAM, un jeton d'accès, un jeton de session, un secret client ou une combinaison de ces éléments. Pour bon nombre d'opérations d'authentification et d'autorisation des utilisateurs, vous avez le choix entre différentes versions authentifiées et non authentifiées de la demande. Les opérations non authentifiées constituent une bonne pratique de sécurité pour les applications que vous distribuez à vos utilisateurs, comme les applications mobiles, car vous n'avez pas besoin d'inclure de secrets dans votre code.

Vous ne pouvez attribuer des autorisations de politiques IAM que pour les [Opérations de gestion authentifiées par IAM](#) et les [Opérations utilisateur authentifiées par IAM](#).

Opérations de gestion authentifiées par IAM

Les opérations de gestion authentifiées par IAM permettent de modifier et d'afficher la configuration de votre groupe d'utilisateurs et de votre client d'application, comme vous le feriez dans la AWS Management Console.

Par exemple, pour modifier votre groupe d'utilisateurs dans une demande d'API [UpdateUserPool](#), vous devez présenter des informations d'identification AWS et des autorisations IAM pour mettre à jour la ressource.

Pour autoriser ces demandes dans l'AWS Command Line Interface (AWS CLI) ou un kit SDK AWS, configurez votre environnement avec des variables d'environnement ou une configuration client qui ajoute des informations d'identification IAM à votre demande. Pour en savoir plus, consultez [Accès à AWS à l'aide de vos informations d'identification AWS](#) dans le document Références générales AWS. Vous pouvez également envoyer des demandes directement aux [points de terminaison de service](#) pour l'API des groupes d'utilisateurs Amazon Cognito. Vous devez autoriser, c'est-à-dire signer, ces demandes à l'aide des informations d'identification AWS que vous incorporez dans l'en-tête des demandes. Pour en savoir plus, consultez [Signature des demandes d'API AWS](#).

Opérations de gestion authentifiées par IAM

AddCustomAttributes

CreateGroup

CreateIdentityProvider

CreateResourceServer

CreateUserImportJob

CreateUserPool

CreateUserPoolClient

CreateUserPoolDomain

DeleteGroup

Opérations de gestion authentifiées par IAM

DeleteIdentityProvider

DeleteResourceServer

DeleteUserPool

DeleteUserPoolClient

DeleteUserPoolDomain

DescribeIdentityProvider

DescribeResourceServer

DescribeRiskConfiguration

DescribeUserImportJob

DescribeUserPool

DescribeUserPoolClient

DescribeUserPoolDomain

GetCSVHeader

GetGroup

GetIdentityProviderByIdentifier

GetSigningCertificate

GetUICustomization

GetUserPoolMfaConfig

ListGroups

ListIdentityProviders

ListResourceServers

Opérations de gestion authentifiées par IAM

ListTagsForResource

ListUserImportJobs

ListUserPoolClients

ListUserPools

ListUsers

ListUsersInGroup

SetRiskConfiguration

SetUICustomization

SetUserPoolMfaConfig

StartUserImportJob

StopUserImportJob

TagResource

UntagResource

UpdateGroup

UpdateIdentityProvider

UpdateResourceServer

UpdateUserPool

UpdateUserPoolClient

UpdateUserPoolDomain

Opérations utilisateur authentifiées par IAM

Les opérations utilisateur authentifiées par IAM permettent d'inscrire, connecter, modifier et afficher vos utilisateurs, ainsi que de gérer leurs informations d'identifications.

Par exemple, vous pouvez avoir un niveau d'application côté serveur qui repose sur un front-end web. Votre application côté serveur est un client confidentiel OAuth de confiance qui est doté d'un accès privilégié à vos ressources Amazon Cognito. Pour inscrire un utilisateur dans l'application, votre serveur peut inclure des informations d'identification AWS dans une demande d'API [AdminCreateUser](#). Pour en savoir plus sur les types de client OAuth, consultez [Client Types](#) (Types de clients) dans le document The OAuth 2.0 Authorization Framework.

Pour autoriser ces demandes dans l'AWS CLI ou un kit SDK AWS, configurez l'environnement de votre application côté serveur avec des variables d'environnement ou une configuration client qui ajoute les informations d'identification IAM à votre demande. Pour en savoir plus, consultez [Accès à AWS à l'aide de vos informations d'identification AWS](#) dans le document Références générales AWS. Vous pouvez également envoyer des demandes directement aux [points de terminaison de service](#) pour l'API des groupes d'utilisateurs Amazon Cognito. Vous devez autoriser, c'est-à-dire signer, ces demandes à l'aide des informations d'identification AWS que vous incorporez dans l'en-tête des demandes. Pour en savoir plus, consultez [Signature des demandes d'API AWS](#).

Si votre client d'application dispose d'un secret client, vous devez fournir à la fois vos informations d'identification IAM et, selon l'opération, le paramètre SecretHash ou la valeur SECRET_HASH dans AuthParameters. Pour de plus amples informations, veuillez consulter [Calcul des valeurs de hachage secret](#).

Opérations utilisateur authentifiées par IAM

AdminAddUserToGroup

AdminConfirmSignUp

AdminCreateUser

AdminDeleteUser

AdminDeleteUserAttributes

AdminDisableProviderForUser

Opérations utilisateur authentifiées par IAM

AdminDisableUser

AdminEnableUser

AdminForgetDevice

AdminGetDevice

AdminGetUser

AdminInitiateAuth

AdminLinkProviderForUser

AdminListDevices

AdminListGroupsWithUser

AdminListUserAuthEvents

AdminRemoveUserFromGroup

AdminResetUserPassword

AdminRespondToAuthChallenge

AdminSetUserMFAPreference

AdminSetUserPassword

AdminSetUserSettings

AdminUpdateAuthEventFeedback

AdminUpdateDeviceStatus

AdminUpdateUserAttributes

AdminUserGlobalSignOut

Opérations utilisateur non authentifiées

Les opérations utilisateur non authentifiées permettent d'inscrire, de connecter les utilisateurs et de leur lancer le processus de réinitialisation de mot de passe. Les opérations d'API non authentifiées, c'est-à-dire publiques, s'avèrent utiles lorsque vous voulez permettre à quiconque sur Internet de s'inscrire et de se connecter à votre application.

Par exemple, pour inscrire un utilisateur dans votre application, vous pouvez distribuer un client public OAuth qui n'offre pas d'accès privilégié aux secrets. Vous pouvez inscrire cet utilisateur à l'aide de l'opération d'API non authentifiée [SignUp](#).

Pour envoyer ces demandes dans un client public que vous avez développé à l'aide d'un kit SDK AWS, vous n'avez pas besoin de configurer d'informations d'identification. Vous pouvez également envoyer des demandes directement aux [points de terminaison de service](#) pour l'API des groupes d'utilisateurs Amazon Cognito sans aucune autorisation supplémentaire.

Si votre client d'application dispose d'un secret client, selon l'opération, vous devez fournir le paramètre `SecretHash` ou la valeur `SECRET_HASH` dans `AuthParameters`. Pour de plus amples informations, veuillez consulter [Calcul des valeurs de hachage secret](#).

Opérations utilisateur non authentifiées

`SignUp`

`ConfirmSignUp`

`ResendConfirmationCode`

`ForgotPassword`

`ConfirmForgotPassword`

`InitiateAuth`

Opérations des utilisateurs autorisés par jeton

Les opérations utilisateur authentifiées par jeton permettent de déconnecter les utilisateurs et de gérer leurs informations d'identification. Elles permettent également de modifier et d'afficher les utilisateurs après qu'ils se soient connectés ou qu'ils aient entamé le processus de connexion. Les opérations d'API authentifiées par jeton s'avèrent utiles lorsque vous ne voulez pas distribuer de

secrets dans votre application et que vous souhaitez autoriser les demandes avec les informations d'identification propres à vos utilisateurs. Si votre utilisateur a mené à bien le processus de connexion, vous devez autoriser sa demande d'API authentifiée par jeton avec un jeton d'accès. Si votre utilisateur se trouve au milieu d'un processus de connexion, vous devez autoriser sa demande d'API authentifiée par jeton avec le jeton de session qu'a renvoyé Amazon Cognito en réponse à la demande précédente.

Par exemple, dans un client public, vous pouvez souhaiter mettre à jour le profil d'un utilisateur de façon à limiter l'accès en écriture uniquement à son profil. Pour effectuer cette mise à jour, votre client peut inclure le jeton d'accès de l'utilisateur dans une demande d'API [UpdateUserAttributes](#).

Pour envoyer ces demandes dans un client public que vous avez développé à l'aide d'un kit SDK AWS, vous n'avez pas besoin de configurer d'informations d'identification. Incluez un paramètre `AccessToken` ou `Session` dans votre demande. Vous pouvez également envoyer des demandes directement aux [points de terminaison de service](#) pour l'API des groupes d'utilisateurs Amazon Cognito. Pour autoriser une demande à destination d'un point de terminaison de service, incluez le jeton d'accès ou de session dans le corps POST de votre demande.

Pour signer une demande d'API pour une opération authentifiée par jeton, incluez le jeton d'accès en tant qu'en-tête `Authorization` dans votre demande, au format `Bearer <Base64-encoded access token>`.

Opérations des utilisateurs autorisés par jeton	AccessTok en	Session
RespondTo AuthChallenge		✓
ChangePassword	✓	
GetUser	✓	
UpdateUserAttributes	✓	
DeleteUserAttributes	✓	

Opérations des utilisateurs autorisés par jeton	AccessTok en	Session
DeleteUser	✓	
ConfirmDevice	✓	
ForgetDevice	✓	
GetDevice	✓	
ListDevices	✓	
UpdateDeviceStatus	✓	
GetUserAttributeVerificationCode	✓	
VerifyUserAttribute	✓	
SetUserSettings	✓	
SetUserMFAPreference	✓	
GlobalSignOut	✓	
AssociateSoftwareToken	✓	✓
UpdateAuthEventFeedback		✓
VerifySoftwareToken	✓	✓
RevokeToken ¹		

¹ RevokeToken accepte un jeton d'actualisation en guise de paramètre. Le jeton d'actualisation sert de jeton d'autorisation et de ressource cible.

Mise à jour de la configuration d'un groupe d'utilisateurs

Pour modifier les paramètres des groupes d'utilisateurs Amazon Cognito dans le AWS Management Console, parcourez les onglets basés sur les fonctionnalités dans les paramètres de votre groupe d'utilisateurs et mettez à jour les champs comme décrit dans d'autres sections de ce guide. Vous ne pouvez pas modifier certains paramètres après avoir créé un groupe d'utilisateurs. Si vous souhaitez modifier les paramètres suivants, vous devez créer un nouveau groupe d'utilisateurs ou client d'application.

Nom de groupe d'utilisateurs

Nom du paramètre d'API : [PoolName](#)

Nom convivial que vous avez attribué à votre groupe d'utilisateurs. Pour modifier le nom d'un groupe d'utilisateurs, créez un nouveau groupe d'utilisateurs.

Options de connexion de groupe d'utilisateurs Amazon Cognito

noms des paramètres d'API : [AliasAttributes](#) et [UsernameAttributes](#)

Attributs que vos utilisateurs peuvent transmettre en tant que nom d'utilisateur lorsqu'ils se connectent. Quand vous créez un groupe d'utilisateurs, vous pouvez choisir d'autoriser la connexion avec un nom d'utilisateur, une adresse e-mail, un numéro de téléphone ou un nom d'utilisateur préféré. Pour modifier les options de connexion de groupe d'utilisateurs, créez un nouveau groupe d'utilisateurs.

Make user name case sensitive (Rendre le nom d'utilisateur sensible à la casse)

Nom du paramètre d'API : [UsernameConfiguration](#)

Lorsque vous créez un nom d'utilisateur correspondant à un autre nom d'utilisateur à l'exception de la casse, Amazon Cognito peut les traiter comme un même utilisateur ou comme des utilisateurs différents. Pour plus d'informations, consultez [Sensibilité à la casse du groupe d'utilisateurs](#). Pour modifier la sensibilité à la casse, créez un nouveau groupe d'utilisateurs.

Secret client

Nom du paramètre d'API : [GenerateSecret](#)

Lorsque vous créez un client d'application, vous pouvez générer un secret client afin que seules les sources autorisées puissent adresser des demandes à votre groupe d'utilisateurs. Pour plus d'informations, consultez [Clients de l'application de groupe d'utilisateurs](#). Pour modifier un secret client, créez un nouveau client d'application dans le même groupe d'utilisateurs.

Attributs requis

Nom du paramètre d'API : [Schema](#)

Attributs pour lesquels vos utilisateurs doivent fournir des valeurs lors de leur inscription ou lorsque vous les créez. Pour plus d'informations, consultez [Attributs de groupe d'utilisateurs](#). Pour modifier les attributs requis, créez un nouveau groupe d'utilisateurs.

Attributs personnalisés

Nom du paramètre d'API : [Schema](#)

Attributs incluant des noms personnalisés. Vous pouvez modifier la valeur d'un attribut personnalisé d'un utilisateur, mais vous ne pouvez pas supprimer un attribut personnalisé de votre groupe d'utilisateurs. Pour plus d'informations, consultez [Attributs de groupe d'utilisateurs](#). Si vous atteignez le nombre maximal d'attributs personnalisés et que vous souhaitez modifier la liste, créez un nouveau groupe d'utilisateurs.

Configuration des SMS

Une fois que vous avez activé les SMS dans votre groupe d'utilisateurs, vous ne pouvez pas les désactiver.

- Si vous choisissez de configurer les SMS lorsque vous créez un groupe d'utilisateurs, vous ne pouvez pas désactiver les SMS une fois la configuration terminée.
- Vous pouvez activer les SMS dans un groupe d'utilisateurs que vous avez créé, mais vous ne pourrez plus les désactiver par la suite.
- Amazon Cognito peut utiliser les SMS pour inviter et récupérer le compte utilisateur, vérifier les attributs et procéder à l'authentification multifactorielle (MFA). Après avoir activé les messages SMS, vous pouvez activer ou désactiver ces fonctions à tout moment.
- La configuration des messages SMS inclut un rôle IAM que vous déléguez à Amazon Cognito pour envoyer des messages via Amazon SNS. Vous pouvez modifier le rôle attribué à tout moment.

Mettre à jour un groupe d'utilisateurs à l'aide d'un AWS SDK ou d'une API REST AWS CDK

Dans la console Amazon Cognito, vous pouvez modifier les paramètres de votre groupe d'utilisateurs un paramètre à la fois. Par exemple, pour ajouter un déclencheur Lambda, choisissez Ajouter un déclencheur Lambda, puis choisissez la fonction et le type de déclencheur. L'API des groupes d'utilisateurs Amazon Cognito est structurée de telle sorte que les opérations de mise à jour pour les groupes d'utilisateurs et les clients d'applications nécessitent l'ensemble complet des paramètres du groupe d'utilisateurs. Cependant, la console automatise de manière transparente cette opération de mise à jour avec les autres paramètres de votre groupe d'utilisateurs.

Vous constaterez peut-être qu'une modification apportée ailleurs dans le vôtre Compte AWS peut entraîner la génération d'une erreur lors des mises à jour lorsqu'elles ne sont pas liées au paramètre que vous souhaitez modifier. Une identité Amazon SES supprimée ou une modification d'une autorisation IAM pour AWS WAF, par exemple. Si l'un des paramètres actuels n'est plus valide, vous ne pouvez pas le mettre à jour tant que vous ne l'avez pas corrigé. Lorsque vous rencontrez une telle erreur, examinez la réponse d'erreur et validez le paramètre qu'elle mentionne.

[L'AWS Cloud Development Kit \(AWS CDK\) API REST et les AWS SDK des groupes d'utilisateurs Amazon Cognito](#) sont des outils d'automatisation et de configuration programmatique des ressources Amazon Cognito. Les demandes utilisant ces outils doivent également, comme la console Amazon Cognito, mettre à jour un paramètre avec une configuration complète des ressources dans le corps de la demande. À un niveau élevé, vous devez exécuter le processus suivant.

1. Capturez le résultat d'une opération qui décrit la configuration de votre ressource existante.
2. Modifiez la sortie en fonction de vos paramètres.
3. Envoyez la configuration modifiée dans le cadre d'une opération qui met à jour votre ressource.

La procédure suivante met à jour votre configuration avec l'opération [UpdateUserPool](#) d'API. La même approche, avec des champs de saisie différents, s'applique à [UpdateUserPoolClient](#).

Important

Si vous ne fournissez pas de valeurs pour les paramètres existants, Amazon Cognito leur affecte les valeurs par défaut. Par exemple, lorsque vous avez un paramètre `LambdaConfig` existant et que vous soumettez un paramètre `UpdateUserPool` avec un paramètre `LambdaConfig` vide, vous supprimez l'affectation de toutes les fonctions Lambda aux

déclencheurs de groupes d'utilisateurs. Planifiez correctement l'automatisation des changements dans votre groupe d'utilisateurs.

1. Capturez l'état actuel de votre groupe d'utilisateurs avec [DescribeUserPool](#).
2. Formatez la sortie de `DescribeUserPool` pour qu'elle corresponde aux [paramètres de demande](#) de `UpdateUserPool`. Supprimez les champs de niveau supérieur suivants et leurs objets enfants de la sortie de code JSON.
 - `Arn`
 - `CreationDate`
 - `CustomDomain`
 - Mettez à jour ce champ avec l'opération [UpdateUserPoolDomainAPI](#).
 - `Domain`
 - Mettez à jour ce champ avec l'opération [UpdateUserPoolDomainAPI](#).
 - `EmailConfigurationFailure`
 - `EstimatedNumberOfUsers`
 - `Id`
 - `LastModifiedDate`
 - `Name`
 - `SchemaAttributes`
 - `SmsConfigurationFailure`
 - `Status`
3. Confirmez que le code JSON résultant correspond aux [paramètres de demande](#) de `UpdateUserPool`.
4. Modifiez tous les paramètres que vous souhaitez modifier dans le code JSON résultant.
5. Soumettez une demande d'API `UpdateUserPool` avec votre code JSON modifié comme entrée de demande.

Vous pouvez également utiliser cette sortie `DescribeUserPool` modifiée dans le paramètre `--cli-input-json` de `update-user-pool` dans l'interface AWS CLI.

Vous pouvez également exécuter la AWS CLI commande suivante pour générer du JSON avec des valeurs vides pour les champs de saisie acceptés pour `update-user-pool`. Vous pouvez ensuite remplir ces champs avec les valeurs existantes de votre groupe d'utilisateurs.

```
aws cognito-idp update-user-pool --generate-cli-skeleton --output json
```

Utilisez la commande suivante pour générer le même objet JSON pour un client d'application.

```
aws cognito-idp update-user-pool-client --generate-cli-skeleton --output json
```

Configuration et utilisation de l'interface utilisateur hébergée par Amazon Cognito et des points de terminaison de fédération

Un groupe d'utilisateurs Amazon Cognito avec un domaine est un serveur d'autorisation conforme à OAuth-2.0 et une interface utilisateur ready-to-use hébergée (UI) pour l'authentification. Le serveur d'autorisation route les demandes d'authentification, émet et gère les jetons Web JSON (JWT) et fournit des informations sur les attributs utilisateur. L'interface utilisateur hébergée est une collection d'interfaces Web pour les activités de base d'inscription, de connexion, d'authentification multifactorielle et de réinitialisation de mot de passe dans votre groupe d'utilisateurs. Il s'agit également d'un hub central pour l'authentification auprès des fournisseurs d'identité tiers (IdPs) que vous associez à votre application. Votre application peut invoquer l'interface utilisateur hébergée et les points de terminaison d'autorisation lorsque vous souhaitez authentifier et autoriser les utilisateurs. Vous pouvez adapter l'expérience utilisateur de l'interface utilisateur hébergée à votre marque grâce à votre propre logo et à la personnalisation CSS. Pour plus d'informations sur les composants de l'interface utilisateur hébergée et du serveur d'autorisation, consultez [Référence de l'interface utilisateur hébergée et des points de terminaison de la fédération des groupes d'utilisateurs](#).

Note

L'interface utilisateur hébergée d'Amazon Cognito ne prend pas en charge l'authentification personnalisée avec les [déclencheurs Lambda de stimulation d'authentification personnalisée](#).

Rubriques

- [Configuration de l'interface utilisateur hébergée avec AWS Amplify](#)

- [Configuration de l'interface utilisateur hébergée avec la console Amazon Cognito](#)
- [Affichage de votre page de connexion](#)
- [Informations utiles concernant l'interface utilisateur hébergée des groupes d'utilisateurs Amazon Cognito](#)
- [Configuration d'un domaine de groupe d'utilisateurs](#)
- [Personnalisation des pages web intégrées d'inscription et de connexion](#)
- [Inscription et connexion à l'aide de l'interface utilisateur hébergée](#)

Configuration de l'interface utilisateur hébergée avec AWS Amplify

Si vous ajoutez l'authentification AWS Amplify à votre application Web ou mobile, vous pouvez configurer votre interface utilisateur hébergée à l'aide de l'interface de ligne de commande (CLI) et des bibliothèques du AWS Amplify framework. Pour ajouter l'authentification à votre application, utilisez l' AWS Amplify CLI pour ajouter la catégorie Auth à votre projet. Ensuite, dans votre code client, vous utilisez les AWS Amplify bibliothèques pour authentifier les utilisateurs auprès de votre groupe d'utilisateurs Amazon Cognito.

Vous pouvez afficher une interface utilisateur hébergée prédéfinie ou fédérer des utilisateurs via un point de terminaison OAuth 2.0 qui redirige vers un fournisseur de connexion par réseaux sociaux, tel que Facebook, Google, Amazon ou Apple. Lorsqu'un utilisateur s'est authentifié auprès du fournisseur social, AWS Amplify crée un nouvel utilisateur dans votre groupe d'utilisateurs, si nécessaire, puis fournit le jeton OIDC de l'utilisateur à votre application.

Les exemples suivants montrent comment AWS Amplify configurer l'interface utilisateur hébergée avec les fournisseurs sociaux dans votre application.

- [AWS Amplify authentification pour JavaScript.](#)
- [AWS Amplify authentification pour Swift.](#)
- [AWS Amplify authentification pour Flutter.](#)
- [AWS Amplify authentification pour Android.](#)

Configuration de l'interface utilisateur hébergée avec la console Amazon Cognito

Créer un client d'application

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
4. Sélectionnez l'onglet Intégration d'applications.
5. Sous Clients d'application, sélectionnez Créer un client d'application.
6. Sélectionnez un type d'application : Client public, Client confidentiel ou Autre. Un client public fonctionne généralement à partir des appareils de vos utilisateurs et utilise des API non authentifiées et authentifiées par jeton. Un client confidentiel fonctionne généralement à partir d'une application installée sur un serveur central à laquelle vous confiez les secrets du client et les informations d'identification de l'API, et utilise des en-têtes d'autorisation et des AWS Identity and Access Management informations d'identification pour signer les demandes. Si votre cas d'utilisation est différent des paramètres préconfigurés du client d'application pour un client public ou un client confidentiel, sélectionnez Autre.
7. Saisissez un nom de client d'application.
8. Sélectionnez le flux d'authentification que vous souhaitez autoriser dans votre client d'application.
9. Configuration de Durée de session du flux d'authentification. Il s'agit du temps dont disposent vos utilisateurs pour terminer chaque défi d'authentification avant l'expiration de leur jeton de session.
10. (Facultatif) Configurez l'expiration du jeton.
 - a. Spécifiez le délai d'expiration du jeton d'actualisation du client d'application. La valeur par défaut est de 30 jours. Vous pouvez le remplacer par toute valeur comprise entre 1 heure et 10 ans.
 - b. Spécifiez le délai d'expiration du jeton d'accès du client d'application. La valeur par défaut est 1 heure. Vous pouvez la remplacer par toute valeur comprise entre 5 minutes et 24 heures.

- c. Spécifiez le délai d'expiration du jeton d'identification du client d'application. La valeur par défaut est 1 heure. Vous pouvez la remplacer par toute valeur comprise entre 5 minutes et 24 heures.

 Important

Si vous utilisez l'interface utilisateur hébergée et configurez des jetons de moins d'une heure, l'utilisateur final sera en mesure d'utiliser des jetons basés sur le cookie de sa session qui est actuellement fixée à une heure.

11. Choisissez Générer un secret client pour qu'Amazon Cognito génère un code secret client pour vous. Les secrets des clients sont généralement associés à des clients confidentiels.
12. Indiquez si vous voulez activer la révocation de jetons pour ce client d'application. Cela augmentera la taille des jetons. Pour plus d'informations, consultez [Révocation des jetons](#).
13. Indiquez si vous voulez Empêcher les messages d'erreur révélant l'existence de l'utilisateur ce client d'application. Amazon Cognito répondra aux demandes de connexion des utilisateurs inexistantes avec un message générique indiquant que le nom d'utilisateur ou le mot de passe était incorrect.
14. (Facultatif) Configurez des autorisations de lecture et d'écriture d'attributs pour ce client d'application. Votre client d'application peut être autorisé à lire et à écrire un sous-ensemble limité du schéma d'attribut de votre groupe d'utilisateurs.
15. Choisissez Créer.
16. Notez l'ID du client. Cela permettra d'identifier le client de l'application dans les demandes d'inscription et de connexion.

Configurer l'application.

1. Dans Intégration d'applications, sélectionnez votre client d'application sous Clients d'application. Vérifiez vos informations d'interface hébergée actuelles.
2. Ajoutez une URL de rappel sous Autoriser les URL de rappel. Une URL de rappel est où l'utilisateur est redirigé après l'aboutissement de la connexion.
3. Ajoutez une URL de déconnexion sous URL de déconnexion autorisées. Une URL de déconnexion est où l'utilisateur est redirigé après la déconnexion.
4. Ajoutez au moins une des options répertoriées dans la liste des fournisseurs d'identité.

5. Sous Types d'octroi OAuth 2.0, sélectionnez Octroi de code d'autorisation pour renvoyer un code d'autorisation qui est ensuite remplacé par des jetons de groupe d'utilisateurs. Étant donné que les jetons ne sont jamais exposés directement à un utilisateur final, ils sont moins susceptibles d'être compromis. Toutefois, une application personnalisée est requise au niveau du backend pour échanger le code d'autorisation contre des jetons de groupe d'utilisateurs. Pour des raisons de sécurité, nous vous recommandons d'utiliser ce flux d'octroi de code d'autorisation avec une [PKCE \(Proof Key for Code Exchange\)](#) pour les applications mobiles.
6. Sous Types d'octrois OAuth 2.0, sélectionnez Octroi implicite pour récupérer les jetons Web JSON (JWT) du groupe d'utilisateurs à partir d'Amazon Cognito. Vous pouvez utiliser ce flux lorsqu'il n'y a pas de backend disponible pour échanger un code d'autorisation contre des jetons. C'est également utile pour déboguer les jetons.
7. Vous pouvez activer les octrois Code d'autorisation et Code implicite, puis utiliser chaque octroi selon vos besoins. Si vous ne sélectionnez pas l'octroi Code d'autorisation ou l'octroi Code implicite et que votre client d'application possède un secret client, vous pouvez activer les octrois Informations d'identification client. Sélectionnez Informations d'identification du client uniquement si votre application doit demander des jetons d'accès pour elle-même et non pas pour un utilisateur.
8. Sélectionnez les périmètres OpenID Connect que vous souhaitez autoriser pour ce client d'application.
9. Sélectionnez Enregistrer les modifications.

Configurer un domaine

1. Accédez à l'onglet Intégration d'application pour votre groupe d'utilisateurs.
2. En regard de Domaine, choisissez Actions et sélectionnez Créer un nom de domaine personnalisé ou Créer un nom de domaine Cognito. Si vous avez déjà configuré un domaine de groupe d'utilisateurs, choisissez Delete Cognito domain (Supprimer un nom de domaine Cognito) ou Delete custom domain (Supprimer un nom de domaine personnalisé) avant de créer un domaine personnalisé.
3. Saisissez un préfixe de domaine disponible à utiliser avec un domaine Cognito. Pour plus d'informations sur la configuration d'un domaine personnalisée, consultez [Utilisation de votre propre domaine pour l'interface utilisateur hébergée](#).
4. Choisissez Créer.

Affichage de votre page de connexion

Dans la console Amazon Cognito, sélectionnez le bouton View Hosted UI (Afficher l'interface utilisateur hébergée) dans la configuration de votre client d'application, sous App clients and analytics (Clients d'application et analyses) dans l'onglet App integration (Intégration d'applications). Ce bouton vous renvoie vers une page de connexion dans votre interface utilisateur hébergée avec les paramètres de base suivants.

- ID du client d'application.
- Demande d'octroi de code d'autorisation
- Demande pour tous les portées que vous avez activées pour le client d'application actuel
- Première URL de rappel de la liste pour le client d'application actuel

Le bouton View hosted UI (Afficher l'interface utilisateur hébergée) est utile lorsque vous souhaitez tester les fonctions de base de votre interface utilisateur hébergée. Vous pouvez personnaliser votre URL de connexion avec des paramètres supplémentaires et modifiés. Dans la plupart des cas, les paramètres générés automatiquement du lien View hosted UI (Afficher l'interface utilisateur hébergée) ne correspondent pas entièrement aux besoins de votre application. Vous devez alors personnaliser l'URL que votre application appelle lorsqu'elle se connecte à vos utilisateurs. Pour de plus amples informations sur les clés et les valeurs de paramètre, veuillez consulter [Référence de l'interface utilisateur hébergée et des points de terminaison de la fédération des groupes d'utilisateurs](#).

La page Web de connexion de l'interface utilisateur hébergée utilise le format d'URL suivant. Cet exemple demande l'octroi d'un code d'autorisation avec le paramètre `response_type=code`.

```
https://<your domain>/oauth2/authorize?response_type=code&client_id=<your app client id>&redirect_uri=<your callback url>
```

Vous pouvez récupérer la chaîne de domaine de votre groupe d'utilisateurs à partir de l'onglet Intégration d'application. Dans le même onglet, vous pouvez identifier les identifiants des clients de l'application, leurs URL de rappel, leurs portées autorisées et d'autres configurations sous Clients d'application et analytique.

Lorsque vous accédez au point de terminaison `/oauth2/authorize` avec vos paramètres personnalisés, Amazon Cognito vous redirige vers le point de terminaison `/oauth2/login` ou, si vous avez un paramètre `identity_provider` ou `idp_identifieur`, vous redirige en mode

silencieux vers la page de connexion de votre IdP. Pour accéder à un exemple d'URL qui contourne l'interface utilisateur hébergée, veuillez consulter [Lancement de séance SAML dans les groupes d'utilisateurs Amazon Cognito](#).

Exemple de demande d'interface utilisateur hébergée pour une autorisation implicite

Vous pouvez afficher la page Web d'interface utilisateur de connexion hébergée à l'URL suivante pour l'octroi de code implicite où `response_type=token`. Après une connexion réussie, Amazon Cognito renvoie des jetons de groupe d'utilisateurs à la barre d'adresse de votre navigateur web.

```
https://mydomain.us-east-1.amazoncognito.com/authorize?
response_type=token&client_id=1example23456789&redirect_uri=https://
mydomain.example.com
```

Les jetons d'identité et d'accès apparaissent sous forme de paramètres ajoutés à votre URL de redirection.

Voici un exemple de réponse à une demande d'octroi implicite.

```
https://mydomain.example.com/
#id_token=eyJraaBcDeF1234567890&access_token=eyJraGhIjKlM1112131415&expires_in=3600&token_type=
```

Informations utiles concernant l'interface utilisateur hébergée des groupes d'utilisateurs Amazon Cognito

L'interface utilisateur hébergée et confirmer les utilisateurs en tant qu'administrateurs

Pour les utilisateurs locaux du groupe d'utilisateurs, l'interface utilisateur hébergée est plus efficace lorsque vous configurez votre groupe d'utilisateurs sur Autoriser Cognito à envoyer automatiquement des messages pour vérifier et confirmer. Lorsque vous activez ce paramètre, Amazon Cognito envoie un message contenant un code de confirmation aux utilisateurs qui s'inscrivent. Lorsque vous confirmez plutôt les utilisateurs en tant qu'administrateurs de groupes d'utilisateurs, l'interface utilisateur hébergée affiche un message d'erreur après leur inscription. Dans cet état, Amazon Cognito a créé le nouvel utilisateur, mais n'a pas été en mesure d'envoyer de

message de vérification. Vous pouvez toujours confirmer les utilisateurs en tant qu'administrateurs, mais ils peuvent contacter votre service d'assistance après avoir rencontré une erreur. Pour plus d'informations sur la confirmation administrative, consultez [Autorisation des utilisateurs à s'inscrire dans votre application, mais en les confirmant en tant qu'administrateur du groupe d'utilisateurs](#).

Consulter les modifications apportées à la configuration de l'interface utilisateur hébergée

Si les modifications de vos pages d'interface utilisateur hébergées n'apparaissent pas immédiatement, patientez quelques minutes, puis actualisez la page.

Décoder les jetons du groupe d'utilisateurs

Les jetons du groupe d'utilisateurs Amazon Cognito sont signés à l'aide d'un algorithme RS256. Vous pouvez décoder et vérifier les jetons du groupe d'utilisateurs à l'aide AWS Lambda de la section [Décoder et vérifier les jetons Amazon Cognito JWT activés](#). GitHub

L'interface utilisateur hébergée et la version TLS

L'interface utilisateur hébergée nécessite un chiffrement pendant le transit. Les domaines du groupe d'utilisateurs fournis par Amazon Cognito nécessitent une version TLS minimale de 1.2. Les domaines personnalisés sont pris en charge mais ne nécessitent pas la version 1.2 du protocole TLS. Amazon Cognito gérant la configuration de l'interface utilisateur hébergée et des points de terminaison du serveur d'autorisation, vous ne pouvez pas modifier les exigences TLS du domaine de votre groupe d'utilisateurs.

L'interface utilisateur hébergée et les politiques CORS

L'interface utilisateur hébergée d'Amazon Cognito ne prend pas en charge les politiques de partage de ressources d'origine croisée (CORS). Une politique CORS dans l'interface utilisateur hébergée empêcherait les utilisateurs de transmettre des paramètres d'authentification dans leurs demandes. Implémentez plutôt une politique CORS dans l'interface Web frontale de votre application. Amazon Cognito renvoie un en-tête de réponse Access-Control-Allow-Origin: * aux demandes adressées aux points de terminaison OAuth suivants.

1. [Point de terminaison de jeton](#)
2. [Point de terminaison de révocation](#)
3. [Point de terminaison UserInfo](#)

Cookies de l'interface utilisateur hébergée et du serveur d'autorisation

Les points de terminaison du groupe d'utilisateurs Amazon Cognito installent des cookies dans les navigateurs des utilisateurs. Les cookies sont conformes aux exigences de certains navigateurs selon lesquelles les sites ne placent pas de cookies tiers. Ils s'appliquent uniquement aux points de terminaison de votre groupe d'utilisateurs et incluent les éléments suivants :

- Un XSRF-TOKEN cookie pour chaque demande.
- `csrf-stateCookie` destiné à assurer la cohérence de la session lorsqu'un utilisateur est redirigé.
- Cookie de `cognito session` qui conserve les tentatives de connexion réussies pendant une heure.

Configuration d'un domaine de groupe d'utilisateurs

Après avoir configuré un client d'application, vous pouvez configurer l'adresse de vos pages web d'inscription et de connexion. Vous pouvez utiliser un domaine hébergé Amazon Cognito et choisir un préfixe de domaine disponible, ou utiliser votre propre adresse web comme domaine personnalisé.

Pour ajouter un client d'application et un domaine hébergé Amazon Cognito avec la AWS Management Console, consultez [Ajout d'une application pour activer l'interface utilisateur web hébergée](#).

Note

Vous ne pouvez pas utiliser les expressions `aws`, `amazon` ni `cognito` dans le préfixe de domaine.

Rubriques

- [Utilisation du domaine Amazon Cognito pour l'interface utilisateur hébergée](#)
- [Utilisation de votre propre domaine pour l'interface utilisateur hébergée](#)

Utilisation du domaine Amazon Cognito pour l'interface utilisateur hébergée

Après avoir configuré un client d'application, vous pouvez configurer l'adresse de vos pages web d'inscription et de connexion. Vous pouvez utiliser le domaine Amazon Cognito hébergé avec votre propre préfixe de domaine.

Note

Pour renforcer la sécurité de vos applications Amazon Cognito, les domaines parents des points de terminaison du groupe d'utilisateurs sont enregistrés dans la [liste des suffixes publics \(PSL\)](#). La PSL aide les navigateurs Web de vos utilisateurs à comprendre de manière cohérente les points de terminaison de votre groupe d'utilisateurs et les cookies qu'ils installent.

Les domaines parents du point de terminaison du groupe d'utilisateurs prennent les formats suivants.

```
auth.Region.amazoncognito.com  
auth-fips.Region.amazoncognito.com
```

Pour ajouter un client d'application et un domaine hébergé par Amazon Cognito avec le AWS Management Console, consultez [Création d'un client d'application](#)

Rubriques

- [Prérequis](#)
- [Étape 1 : configuration d'un domaine de groupe d'utilisateurs hébergé](#)
- [Étape 2 : Vérification de votre page de connexion](#)

Prérequis

Avant de commencer, vous avez besoin des éléments suivants :

- Un groupe d'utilisateurs avec un client d'appli. Pour plus d'informations, consultez [Démarrage avec les groupes d'utilisateurs](#).

Étape 1 : configuration d'un domaine de groupe d'utilisateurs hébergé

Pour configurer un domaine de groupe d'utilisateurs hébergé

Vous pouvez utiliser l'API AWS Management Console ou l'API AWS CLI or pour configurer un domaine de groupe d'utilisateurs.

Amazon Cognito console

Configurer un domaine

1. Accédez à l'onglet Intégration d'application pour votre groupe d'utilisateurs.
2. En regard de Domaine, choisissez Actions et sélectionnez Créer un nom de domaine personnalisé ou Créer un nom de domaine Amazon Cognito. Si vous avez déjà configuré un domaine de groupe d'utilisateurs, choisissez Supprimer le domaine Amazon Cognito ou Supprimer le domaine personnalisé avant de créer votre nouveau domaine personnalisé.
3. Saisissez un préfixe de domaine disponible à utiliser avec un domaine Amazon Cognito. Pour plus d'informations sur la configuration d'un domaine personnalisée, consultez [Utilisation de votre propre domaine pour l'interface utilisateur hébergée](#).
4. Choisissez Créer.

CLI/API

Utilisez les commandes suivantes pour créer un préfixe de domaine et l'attribuer à votre groupe d'utilisateurs.

Pour configurer un domaine de groupe d'utilisateurs

- AWS CLI: `aws cognito-idp create-user-pool-domain`

Exemple : `aws cognito-idp create-user-pool-domain --user-pool-id <user_pool_id> --domain <domain_name>`

- AWS API : [CreateUserPoolDomain](#)

Pour obtenir des informations sur un domaine

- AWS CLI: `aws cognito-idp describe-user-pool-domain`

Exemple : `aws cognito-idp describe-user-pool-domain --domain <domain_name>`

- AWS API : [DescribeUserPoolDomain](#)

Pour supprimer un domaine

- AWS CLI: `aws cognito-idp delete-user-pool-domain`

```
Exemple : aws cognito-idp delete-user-pool-domain --domain <domain_name>
```

- AWS API : [DeleteUserPoolDomain](#)

Étape 2 : Vérification de votre page de connexion

- Vérifiez que la page de connexion est disponible à partir du domaine hébergé par Amazon Cognito.

```
https://<your_domain>/login?  
response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callback_url>
```

Votre domaine est affiché sur la page Domain name (Nom de domaine) de la console Amazon Cognito. Votre ID de client d'application et votre URL de rappel s'affichent à la page App client settings (Paramètres de client d'application).

Utilisation de votre propre domaine pour l'interface utilisateur hébergée

Après avoir configuré un client d'application, vous pouvez configurer votre groupe d'utilisateurs avec un domaine personnalisé pour l'interface utilisateur hébergée d'Amazon Cognito et les points de terminaison de l'[API d'authentification](#). Avec un domaine personnalisé, vous permettez à vos utilisateurs de se connecter à votre application à l'aide de votre propre adresse web.

Rubriques

- [Ajout d'un domaine personnalisé à un groupe d'utilisateurs](#)
- [Modification du certificat SSL de votre domaine personnalisé](#)


Ajout d'un domaine personnalisé à un groupe d'utilisateurs

Pour ajouter un domaine personnalisé à votre groupe d'utilisateurs, vous spécifiez le nom de domaine dans la console Amazon Cognito et fournissez un certificat que vous gérez avec [AWS Certificate Manager](#) (ACM). Une fois que vous avez ajouté votre domaine, Amazon Cognito fournit une cible d'alias que vous ajoutez à votre configuration DNS.

Prérequis


Avant de commencer, vous avez besoin des éléments suivants :

- Un groupe d'utilisateurs avec un client d'appli. Pour plus d'informations, consultez [Démarrage avec les groupes d'utilisateurs](#).
- Un domaine Web que vous possédez. Son domaine parent doit avoir un enregistrement A DNS valide. Vous pouvez attribuer n'importe quelle valeur à cet enregistrement. Le parent peut être la racine du domaine ou un domaine enfant qui se trouve à un niveau supérieur dans la hiérarchie du domaine. Par exemple, si votre domaine personnalisé est `auth.xyz.example.com`, Amazon Cognito doit être en mesure de résoudre `xyz.example.com` à une adresse IP. Pour éviter tout impact accidentel sur l'infrastructure du client, Amazon Cognito ne prend pas en charge l'utilisation de domaines de premier niveau (TLD) pour les domaines personnalisés. Pour plus d'informations, consultez [Noms de domaine](#).
- Possibilité de créer un sous-domaine pour votre domaine personnalisé. Nous vous recommandons d'utiliser `auth` comme sous-domaine. Par exemple : `auth.exemple.com`.

 Note

Vous pouvez avoir besoin d'obtenir un nouveau certificat pour le sous-domaine de votre domaine personnalisé si vous ne disposez pas d'un [certificat générique](#).

- Certificat SSL (Secure Sockets Layer) géré par ACM.

 Note

Vous devez changer la AWS région en USA Est (Virginie du Nord) dans la console ACM avant de demander ou d'importer un certificat.

- Application qui permet à votre serveur d'autorisation de groupe d'utilisateurs d'ajouter des cookies aux sessions utilisateur. Amazon Cognito définit plusieurs cookies obligatoires pour l'interface utilisateur hébergée. Cela inclut `cognito`, `cognito-fl` et `XSRF-TOKEN`. Bien que chaque cookie individuel soit conforme aux limites de taille du navigateur, les modifications apportées à la configuration de votre groupe d'utilisateurs peuvent entraîner une augmentation de la taille des cookies d'interface utilisateur hébergés. Un service intermédiaire tel qu'un Application Load Balancer (ALB) placé devant votre domaine personnalisé peut imposer une taille d'en-tête maximale ou une taille totale de cookie. Si votre application définit également ses propres cookies, les sessions de vos utilisateurs peuvent dépasser ces limites. Pour éviter les conflits de taille, nous recommandons à votre application de ne pas définir de cookies sur le sous-domaine de l'interface utilisateur hébergé.

- Autorisation de mettre à jour CloudFront les distributions Amazon. Vous pouvez le faire en attachant la déclaration de politique IAM suivante à un utilisateur de votre compte Compte AWS :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontUpdateDistribution",
      "Effect": "Allow",
      "Action": [
        "cloudfront:updateDistribution"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Pour plus d'informations sur l'autorisation d'actions dans CloudFront, consultez la section [Utilisation de politiques basées sur l'identité \(stratégies IAM\)](#) pour CloudFront

Amazon Cognito utilise initialement vos autorisations IAM pour configurer la CloudFront distribution, mais celle-ci est gérée par AWS. Vous ne pouvez pas modifier la configuration de la CloudFront distribution qu'Amazon Cognito a associée à votre groupe d'utilisateurs. Par exemple, vous ne pouvez pas mettre à jour les versions TLS prises en charge dans la politique de sécurité.

Étape 1 : saisissez votre nom de domaine personnalisé.


Vous pouvez ajouter votre domaine à votre groupe d'utilisateurs à l'aide de la console Amazon Cognito ou de l'API.

Amazon Cognito console

Pour ajouter votre domaine à votre groupe d'utilisateurs à partir de la console Amazon Cognito :

1. Connectez-vous à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos informations d'identification AWS .
2. Choisissez Groupes d'utilisateurs.
3. Choisissez le groupe d'utilisateurs auquel vous souhaitez ajouter votre domaine.

4. Choisissez l'onglet App integration (Intégration d'applications).
5. En regard de Domaine, choisissez Actions, puis Créer un domaine personnalisé.

 Note

Si vous avez déjà configuré un domaine de groupe d'utilisateurs, choisissez Delete Cognito domain (Supprimer le domaine Cognito) ou Delete custom domain (Supprimer un domaine personnalisé) pour supprimer le domaine existant avant de créer votre domaine personnalisé.

6. Pour Domaine personnalisé, saisissez l'URL du domaine que vous souhaitez utiliser avec Amazon Cognito. Votre nom de domaine peut uniquement contenir des lettres minuscules, des chiffres et des traits d'union. N'utilisez pas de tiret comme premier ou dernier caractère. Utilisez des points pour séparer les noms des sous-domaines.
7. Pour Certificat ACM, choisissez le certificat SSL que vous souhaitez utiliser pour votre domaine. Seuls les certificats ACM de l'est des États-Unis (Virginie du Nord) peuvent être utilisés avec un domaine personnalisé Amazon Cognito, quel que soit votre groupe d'utilisateurs. Région AWS

Si vous n'avez pas de certificat disponible, vous pouvez utiliser ACM pour allouer un dans la région USA Est (Virginie du Nord). Pour plus d'informations, consultez [Démarrer](#) dans le Guide de l'utilisateur AWS Certificate Manager .

8. Choisissez Créer.
9. Amazon Cognito vous renvoie à l'onglet Intégration d'applications. Le message Créez un enregistrement d'alias dans le DNS de votre domaines'affiche. Notez le domaine et la cible d'alias affichée dans la console. Ils seront utilisés dans l'étape suivante pour diriger le trafic vers votre domaine personnalisé.

API

Pour ajouter votre domaine à votre groupe d'utilisateurs avec l'API Amazon Cognito) :

- Utilisez l'action [CreateUserPoolDomain](#).

Étape 2 : Ajout d'une cible d'alias et d'un sous-domaine

Au cours de cette étape, vous allez configurer un alias via votre fournisseur de service DNS (Domain Name Server) qui renvoie vers la cible de l'alias de l'étape précédente. Si vous utilisez Amazon Route 53 pour la résolution d'adresse DNS, choisissez la section Pour ajouter une cible d'alias et un sous-domaine à l'aide de Route 53.

Pour ajouter une cible d'alias et un sous-domaine à votre configuration DNS actuelle

- Si vous n'utilisez pas Route 53 pour la résolution d'adresse DNS, vous devez utiliser les outils de configuration de votre fournisseur de services DNS pour ajouter la cible d'alias de l'étape précédente à l'enregistrement DNS de votre domaine. Votre fournisseur DNS doit également configurer le sous-domaine pour votre domaine personnalisé.

Pour ajouter une cible d'alias et un sous-domaine à l'aide de Route 53

1. Connectez-vous à la [console Route 53](#). Si vous y êtes invité, saisissez vos Informations d'identification AWS .
2. Si vous n'avez pas de zone hébergée dans Route 53, créez-en une avec une racine parent de votre domaine personnalisé. Pour plus d'informations, veuillez consulter la rubrique
 - a. Choisissez Créer une zone hébergée.
 - b. Saisissez le domaine parent, par exemple, *auth.example.com*, de votre domaine personnalisé, par exemple, *myapp.auth.example.com*, à partir de la liste Nom de domaine.
 - c. Saisissez une description pour votre zone hébergée.
 - d. Choisissez Zone hébergée publique comme type de zone hébergée pour permettre aux clients publics de résoudre votre domaine personnalisé. Le choix Zone hébergée privée n'est pas pris en charge.
 - e. Appliquez des identifications à votre convenance.
 - f. Choisissez Créer une zone hébergée.

Note

Vous pouvez également créer une zone hébergée pour votre domaine personnalisé, et vous pouvez créer un jeu de délégations dans la zone hébergée parent qui dirige les requêtes vers la zone hébergée du sous-domaine. Sinon, créez un

enregistrement A. Cette méthode offre plus de flexibilité et de sécurité avec vos zones hébergées. Pour plus d'informations, consultez [Création d'un sous-domaine pour un domaine hébergé via Amazon Route 53](#).

3. Sur la page Hosted Zones (Zones hébergées), choisissez le nom de votre zone hébergée.
4. Ajoutez un enregistrement DNS pour le domaine parent de votre domaine personnalisé, si vous n'en avez pas déjà un. Ajoutez un A enregistrement DNS pour le domaine parent et choisissez Create records. Voici un exemple d'enregistrement pour le domaine *auth.example.com*.

```
auth.example.com. 60 IN A 198.51.100.1
```

Note

Amazon Cognito vérifie qu'il existe un enregistrement DNS pour le domaine parent de votre domaine personnalisé, afin de créer une protection contre le piratage accidentel de domaines de production. Si vous n'avez pas d'enregistrement DNS pour le domaine parent, Amazon Cognito renvoie une erreur lorsque vous tentez de définir le domaine personnalisé. Un enregistrement SOA (Start of Authority) n'est pas un enregistrement DNS suffisant aux fins de la vérification du domaine parent.

5. Ajoutez un enregistrement DNS pour votre domaine personnalisé. Votre enregistrement doit pointer vers la cible Alias de domaine personnalisée, par exemple *123example.cloudfront.net*. Choisissez Créer un enregistrement de nouveau.
6. Saisissez un nom de l'enregistrement qui correspond à votre domaine personnalisé, par exemple, *monapp* pour créer un enregistrement pour *monapp.auth.exemple.com*.
7. Activer l'option Alias.
8. Choisissez d'Acheminer le trafic vers un Alias vers distribution CloudFront. Saisissez la cible d'alias fournie par Amazon Cognito lorsque vous avez créé votre domaine personnalisé.
9. Choisissez Créer des enregistrements.

Note

Vos nouveaux enregistrements peuvent prendre environ 60 secondes pour se propager à tous les serveurs DNS Route 53. Vous pouvez utiliser la méthode de [l'GetChangeAPI](#) Route 53 pour vérifier que vos modifications se sont propagées.

Étape 3 : Vérification de votre page de connexion

- Vérifiez que la page de connexion est disponible à partir de votre domaine personnalisé.

Connectez-vous avec votre domaine personnalisé et votre sous-domaine en saisissant cette adresse dans votre navigateur. Voici un exemple d'URL de domaine personnalisé *exemple.com* avec le sous-domaine *auth* :

```
https://myapp.auth.example.com/login?  
response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callback_url>
```

Modification du certificat SSL de votre domaine personnalisé

Si nécessaire, vous pouvez utiliser Amazon Cognito pour modifier le certificat que vous avez appliqué à votre nom de domaine personnalisé.

Généralement, cela est inutile après un renouvellement de certificat avec ACM. Lorsque vous renouvelez votre certificat existant dans ACM, l'ARN de votre certificat demeure le même et votre nom de domaine personnalisé utilise le nouveau certificat automatiquement.

Toutefois, si vous remplacez votre certificat existant par un nouveau certificat, ACM attribue au nouveau certificat un nouvel ARN. Pour appliquer le nouveau certificat à votre domaine personnalisé, vous devez fournir cet ARN à Amazon Cognito.

Une fois que vous avez fourni votre nouveau certificat, Amazon Cognito nécessite jusqu'à 1 heure pour le distribuer à votre domaine personnalisé.

Avant de commencer

Avant de pouvoir modifier votre certificat dans Amazon Cognito, vous devez ajouter votre certificat à ACM. Pour plus d'informations, consultez [Mise en route](#) dans le Guide de l'utilisateur AWS Certificate Manager .

Lorsque vous ajoutez votre certificat à ACM, vous devez choisir USA Est (Virginie du Nord) comme région AWS .

Vous pouvez modifier votre certificat à l'aide de la console Amazon Cognito ou de l'API.

AWS Management Console

Pour renouveler un certificat à partir de la console Amazon Cognito :

1. Connectez-vous à la console Amazon Cognito AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/cognito/home>
2. Choisissez Groupes d'utilisateurs.
3. Choisissez le groupe d'utilisateurs pour lequel vous souhaitez mettre à jour le certificat.
4. Choisissez l'onglet App integration (Intégration d'applications).
5. Choisissez Actions, Modifier le certificat ACM.
6. Sélectionnez le nouveau certificat que vous souhaitez associer à votre domaine personnalisé.
7. Sélectionnez Enregistrer les modifications.

API

Pour renouveler un certificat (API Amazon Cognito)

- Utilisez l'action [UpdateUserPoolDomain](#).

Personnalisation des pages web intégrées d'inscription et de connexion

Vous pouvez utiliser AWS Management Console, l'AWS CLI ou l'API pour spécifier les paramètres de personnalisation pour l'expérience de l'interface utilisateur intégrée à l'application. Vous pouvez télécharger une image de logo personnalisé à afficher dans l'application. Vous pouvez également utiliser des feuilles de style en cascade (CSS) pour personnaliser l'apparence de l'interface utilisateur.

Vous pouvez spécifier les paramètres de personnalisation de l'IU de l'application pour un seul client (avec un `clientId` spécifique) ou pour tous les clients (en définissant les `clientId` sur ALL). Si vous spécifiez ALL, la configuration par défaut sera utilisée pour chaque client qui n'ayant aucune personnalisation d'IU définie au préalable. Si vous spécifiez des paramètres de personnalisation d'IU pour un client en particulier, il ne reviendra plus à la configuration ALL.

La taille de la demande qui définit votre personnalisation d'IU ne doit pas dépasser 135 Ko. Dans de rares cas, la somme des en-têtes de la demande, de votre fichier CSS et de votre logo peut dépasser 135 Ko. Amazon Cognito code le fichier image en Base64. Cela augmente la taille d'une image de 100 Ko à 130 Ko, ce qui laisse 5 Ko pour les en-têtes de demande et votre fichier CSS. Si la demande est trop volumineuse, la AWS Management Console ou votre demande d'API

`SetUICustomization` renvoie une erreur `request parameters too large`. Faites en sorte que l'image de votre logo ne dépasse pas 100 Ko et que votre fichier CSS ne dépasse pas 3 Ko. Vous ne pouvez pas définir la personnalisation du CSS et du logo séparément.

Note

Pour personnaliser votre IU, vous devez configurer un domaine pour votre groupe d'utilisateurs.

Spécification d'un logo personnalisé pour l'application

Amazon Cognito centre votre logo personnalisé au-dessus des champs de saisie du [Point de terminaison de connexion](#).

Choisissez un fichier PNG, JPG ou JPEG qui puisse être redimensionné à 350 x 178 pixels pour le logo personnalisé de votre interface utilisateur hébergée. La taille de votre fichier de logo ne doit pas dépasser 100 Ko, ou 130 Ko une fois qu'Amazon Cognito l'a codé au format Base64. Pour définir un élément `ImageFile` dans [SetUICustomization](#), dans l'API, convertissez votre fichier en chaîne de texte codée en Base64 ou, dans l'interface AWS CLI, fournissez un chemin de fichier et laissez Amazon Cognito le coder à votre place.

Spécification de personnalisations CSS pour l'application

Vous pouvez personnaliser le CSS pour les pages d'application hébergée, avec les restrictions suivantes :

- Vous pouvez utiliser l'un des noms de classe CSS suivants :
 - `background-customizable`
 - `banner-customizable`
 - `errorMessage-customizable`
 - `idpButton-customizable`
 - `idpButton-customizable:hover`
 - `idpDescription-customizable`
 - `inputField-customizable`
 - `inputField-customizable:focus`
 - `label-customizable`

- `legalText-customizable`
 - `logo-customizable`
 - `passwordCheck-valid-customizable`
 - `passwordCheck-notValid-customizable`
 - `redirect-customizable`
 - `socialButton-customizable`
 - `submitButton-customizable`
 - `submitButton-customizable: hover`
 - `textDescription-customizable`
- Les valeurs de propriétés peuvent contenir du code HTML, à l'exception des valeurs suivantes : `@import`, `@supports`, `@page`, ou `@media` ou Javascript.

Vous pouvez personnaliser les propriétés CSS suivantes.

Étiquettes

- épaisseur de police est un multiple de 100, de 100 à 900.

Champs de saisie

- largeur représente la largeur du bloc contenant en pourcentage.
- hauteur est la hauteur du champ d'entrée en pixels (px).
- couleur est la couleur du texte. Il peut s'agir de toute valeur de couleur CSS standard.
- couleur d'arrière-plan est la couleur d'arrière-plan du champ d'entrée. Il peut s'agir de toute valeur de couleur CSS standard.
- bordure est une valeur de bordure CSS standard qui spécifie la largeur, la transparence et la couleur de la bordure de la fenêtre de votre application. La valeur de la largeur peut être comprise entre 1px et 100px. La transparence peut être solide ou inexistante. La couleur peut être toute valeur de couleur standard.

Descriptions texte

- marge supérieure est la quantité de remplissage au-dessus de la description du texte.
- marge inférieure est la quantité de remplissage au-dessous de la description du texte.
- affichage peut être `block` ou `inline`.
- taille de police est la taille de la police pour les descriptions de texte.

Bouton de soumission

- `taille de police` est la taille de la police pour le texte du bouton.
- `épaisseur de la police` est l'épaisseur de la police du texte du bouton : `bold`, `italic` ou `normal`.
- `marge` est une chaîne de 4 valeurs indiquant la taille des marges en haut, à droite, en bas et à gauche pour le bouton.
- `taille de police` est la taille de la police pour les descriptions de texte.
- `largeur` est la largeur du bouton en pourcentage du texte du bloc contenant.
- `hauteur` est la hauteur du bouton en pixels (px).
- `couleur` est la couleur du texte du bouton. Il peut s'agir de toute valeur de couleur CSS standard.
- `couleur d'arrière-plan` est la couleur d'arrière-plan du bouton. Il peut s'agir de toute valeur de couleur standard.

Bannière

- `remplissage` est une chaîne de 4 valeurs indiquant la taille du remplissage en haut, à droite, en bas et à gauche pour la bannière.
- `couleur d'arrière-plan` est la couleur d'arrière-plan de la bannière. Il peut s'agir de toute valeur de couleur CSS standard.

Info-bulle de bouton de soumission

- `couleur` est la couleur de premier plan du bouton lorsque vous le survolez. Il peut s'agir de toute valeur de couleur CSS standard.
- `couleur d'arrière-plan` est la couleur d'arrière-plan du bouton lorsque vous le survolez. Il peut s'agir de toute valeur de couleur CSS standard.

Info-bulle de bouton de fournisseur d'identité

- `couleur` est la couleur de premier plan du bouton lorsque vous le survolez. Il peut s'agir de toute valeur de couleur CSS standard.
- `couleur d'arrière-plan` est la couleur d'arrière-plan du bouton lorsque vous le survolez. Il peut s'agir de toute valeur de couleur CSS standard.

Vérification de mot de passe non valide

- `couleur` est la couleur du texte du message "Password check not valid". Il peut s'agir de toute valeur de couleur CSS standard.

Contexte

- couleur d'arrière-plan est la couleur d'arrière-plan de la fenêtre de l'application. Il peut s'agir de toute valeur de couleur CSS standard.

Messages d'erreur

- marge est une chaîne de 4 valeurs indiquant la taille des marges en haut, à droite, en bas et à gauche.
- remplissage est la taille de remplissage.
- taille de police est la taille de la police.
- largeur est la largeur du message d'erreur sous forme de pourcentage du bloc contenant.
- arrière-plan est la couleur d'arrière-plan du message d'erreur. Il peut s'agir de toute valeur de couleur CSS standard.
- bordure est une chaîne de 3 valeurs spécifiant la largeur, la transparence et la couleur de la bordure.
- couleur est la couleur du texte du message. Il peut s'agir de toute valeur de couleur CSS standard.
- Dimension de la boîte est utilisé pour indiquer au navigateur les propriétés de dimensions (largeur et hauteur).

Boutons de fournisseur d'identité

- hauteur est la hauteur du bouton en pixels (px).
- largeur est la largeur du texte du bouton sous forme de pourcentage du bloc contenant.
- alignement du texte est le paramètre d'alignement du texte. Il peut être `left`, `right` ou `center`.
- marge inférieure est le paramètre de la marge inférieure.
- couleur est la couleur du texte du bouton. Il peut s'agir de toute valeur de couleur CSS standard.
- couleur d'arrière-plan est la couleur d'arrière-plan du bouton. Il peut s'agir de toute valeur de couleur CSS standard.
- couleur de bordure est la couleur de la bordure du bouton. Il peut s'agir de toute valeur de couleur CSS standard.

Descriptions de fournisseur d'identité

- marge supérieure est la quantité de remplissage au-dessus de la description.
- marge inférieure est la quantité de remplissage au-dessous de la description.

- affichage peut être `block` ou `inline`.
- taille de police est la taille de la police pour les descriptions.

Texte légal

- couleur est la couleur du texte. Il peut s'agir de toute valeur de couleur CSS standard.
- taille de police est la taille de la police.

Note

Lorsque vous personnalisez le Texte juridique, vous personnalisez la messagerie. Nous ne publierons aucun de vos comptes sans demander au préalable ce qui s'affiche sous les fournisseurs d'identité sociale sur la page de connexion.

Logo

- largeur max est la largeur maximum sous forme de pourcentage du bloc contenant.
- hauteur max est la hauteur maximum sous forme de pourcentage du bloc contenant.

Focus sur le champ de saisie

- couleur de bordure est la couleur du champ de saisie. Il peut s'agir de toute valeur de couleur CSS standard.
- contour est la largeur de la bordure du champ de saisie en pixels (px).

Bouton de réseaux sociaux

- hauteur est la hauteur du bouton en pixels (px).
- alignement du texte est le paramètre d'alignement du texte. Il peut être `left`, `right` ou `center`.
- largeur est la largeur du texte du bouton sous forme de pourcentage du bloc contenant.
- marge inférieure est le paramètre de la marge inférieure.

Vérification de mot de passe valide

- couleur est la couleur du texte du message "Password check valid". Il peut s'agir de toute valeur de couleur CSS standard.

Spécification des paramètres de personnalisation de l'interface utilisateur de l'application pour un groupe d'utilisateurs (AWS Management Console)

Vous pouvez utiliser AWS Management Console pour spécifier les paramètres de personnalisation de l'IU pour votre application.

Note

Vous pouvez afficher l'IU hébergée avec vos personnalisations en construisant

l'URL suivante avec les spécificités de votre groupe d'utilisateurs, puis en

la tapant dans votre navigateur : `https://<your_domain>/login?`

`response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callback`

Une minute d'attente sera probablement nécessaire pour actualiser votre navigateur avant que les modifications appliquées à votre console n'apparaissent.

Votre domaine est affiché sur le site Intégration d'applications onglet sous Domaine. Votre ID de client d'application et votre URL de rappel s'affichent dans l'onglet Clients d'application.

Pour spécifier les paramètres de personnalisation de l'interface utilisateur de l'application

1. Connectez-vous à la [console Amazon Cognito](#).
2. Dans le volet de navigation, choisissez Groupe d'utilisateurs, puis choisissez le groupe d'utilisateurs que vous souhaitez modifier.
3. Choisissez l'onglet App integration (Intégration d'applications).
4. Pour personnaliser les paramètres de l'interface utilisateur de tous les clients d'applications, recherchez Personnalisation de l'interface utilisateur hébergée et sélectionnez Modifier.
5. Pour personnaliser les paramètres de l'interface utilisateur d'un client d'application, localisez Clients d'application et sélectionnez le client d'application que vous souhaitez modifier, puis localisez Personnalisation de l'interface utilisateur hébergée et sélectionnez Modifier. Pour passer d'un client d'application de la personnalisation par défaut du groupe d'utilisateurs à la personnalisation spécifique au client, sélectionnez Utiliser les paramètres au niveau du client.
6. Pour télécharger votre propre fichier image logo, choisissez Choisir un fichier ou Remplacer le fichier actuel.
7. Pour personnaliser le fichier CSS d'interface utilisateur hébergée, téléchargez CSS template.css et modifiez le modèle avec les valeurs que vous souhaitez personnaliser. Seules les clés incluses dans le modèle peuvent être utilisées avec l'interface utilisateur hébergée. Les clés

CSS ajoutées ne seront pas reflétées dans votre interface utilisateur. Après avoir personnalisé le fichier CSS, choisissez Choisir un fichier ou Remplacer le fichier actuel pour télécharger votre fichier CSS personnalisé.

Spécification des paramètres de personnalisation de l'interface utilisateur de l'application pour un groupe d'utilisateurs (AWS CLI et API AWS)

Utilisez les commandes suivantes pour spécifier les paramètres de personnalisation de l'IU pour votre groupe d'utilisateurs.

Pour obtenir les paramètres de personnalisation de l'IU de l'application intégrée d'un groupe d'utilisateurs, utilisez les opérations d'API suivantes.

- AWS CLI: `aws cognito-idp get-ui-customization`
- AWS API : [GetUICustomization](#)

Pour définir les paramètres de personnalisation de l'IU de l'application intégrée d'un groupe d'utilisateurs, utilisez les opérations d'API suivantes.

- AWS CLI à partir du fichier image : `aws cognito-idp set-ui-customization --user-pool-id <your-user-pool-id> --client-id <your-app-client-id> --image-file fileb://<path-to-logo-image-file> --css ".label-customizable{ color: <color>;}"`
- AWS CLI avec l'image codée sous forme de texte binaire Base64 : `aws cognito-idp set-ui-customization --user-pool-id <your-user-pool-id> --client-id <your-app-client-id> --image-file <base64-encoded-image-file> --css ".label-customizable{ color: <color>;}"`
- AWS API : [SetUICustomization](#)

Inscription et connexion à l'aide de l'interface utilisateur hébergée

Une fois que vous avez configuré et personnalisé l'interface utilisateur hébergée par Amazon Cognito pour votre groupe d'utilisateurs et vos clients d'application, votre application peut la présenter à vos utilisateurs. L'interface utilisateur hébergée prend en charge plusieurs opérations d'authentification Amazon Cognito, notamment les exemples suivants :

- Se connecter en tant que nouvel utilisateur dans votre application

- Vérifier une adresse e-mail ou un numéro de téléphone
- Configurer une authentification multifactorielle (MFA)
- Se connecter à l'aide d'un nom d'utilisateur et d'un mot de passe locaux
- Se connecter avec un fournisseur d'identité (IdP) tiers
- Réinitialiser un mot de passe

L'interface utilisateur hébergée Amazon Cognito commence au [Point de terminaison de connexion](#).

L'URL vers votre page de connexion est une combinaison du domaine que vous avez choisi pour votre groupe d'utilisateurs et des paramètres qui reflètent les autorisations OAuth 2.0 que vous souhaitez émettre, votre client d'application, le chemin d'accès à votre application et les portées OpenID Connect (OIDC) que vous souhaitez demander.

```
https://<your user pool domain>/authorize?client_id=<your app client ID>&response_type=<code/token>&scope=<scopes to request>&redirect_uri=<your callback URL>
```

L'URL suivante remplace les champs d'espace réservé ci-dessus par des exemples de valeur.

```
https://auth.example.com/authorize? /
client_id=1example23456789 /
&response_type=code /
&scope=aws.cognito.signin.user.admin+email+openid+profile /
&redirect_uri=https%3A%2F%2Faws.amazon.com
```

La page de connexion de l'interface utilisateur hébergée par Amazon Cognito propose des options permettant de se connecter via le groupe d'utilisateurs ou tout fournisseur d'identité (IdP) que vous avez attribué au client d'application demandé par votre utilisateur. Elle contient également des liens pour créer un nouveau compte utilisateur dans le groupe d'utilisateurs ou pour réinitialiser un mot de passe oublié.


The image shows a user interface for signing in. On the left, there are three sections: 'Sign in with your corporate ID' with a blue button labeled 'MYSSO'; 'Sign In with your social account' with four buttons: 'Continue with Apple' (black), 'Continue with Login with Amazon' (yellow), 'Continue with Google' (blue), and 'Continue with Facebook' (dark blue); and a note 'We won't post to any of your accounts without asking first'. On the right, there is a 'Sign in with your username and password' section with 'Username' and 'Password' input fields, an 'OR' separator, and a 'Forgot your password?' link. At the bottom right, there is a blue 'Sign in' button and a link 'Need an account? Sign up'.

Rubriques

- [Comment créer un nouveau compte dans l'interface utilisateur hébergée par Amazon Cognito](#)
- [Comment se connecter à l'interface utilisateur hébergée par Amazon Cognito](#)
- [Comment réinitialiser un mot de passe avec l'interface utilisateur hébergée par Amazon Cognito](#)

Comment créer un nouveau compte dans l'interface utilisateur hébergée par Amazon Cognito

Ce guide explique comment créer un compte utilisateur dans les applications qui utilisent Amazon Cognito.

 Note

Lorsque vous vous connectez à une application qui utilise l'interface utilisateur hébergée par Amazon Cognito, vous pouvez voir une page que le propriétaire de l'application a personnalisée au-delà de la configuration de base présentée dans ce guide.

1. Choisissez Sign up (S'inscrire) depuis la page de connexion si vous avez l'intention de vous connecter via Amazon Cognito à l'aide d'un nom d'utilisateur et d'un mot de passe, plutôt que par l'intermédiaire d'un des fournisseurs de connexion tiers répertoriés par le propriétaire de l'application.

Si votre fournisseur de connexion n'est pas Amazon Cognito, votre inscription est terminée une fois que vous avez cliqué sur le bouton correspondant à votre fournisseur tiers. En fonction des options choisies par le propriétaire de l'application, vous pouvez choisir entre différents fournisseurs pour vous connecter ou uniquement voir s'afficher un message demandant un nom d'utilisateur et un mot de passe.

With multiple sign-in providers

The image shows a user interface for signing in with multiple providers. It is divided into two main sections: social/corporate ID sign-in and username/password sign-in.

Sign in with your corporate ID

- MYSSO

Sign In with your social account

- Continue with Apple
- Continue with Login with Amazon
- Continue with Google
- Continue with Facebook

We won't post to any of your accounts without asking first

Sign in with your username and password

Username

Password

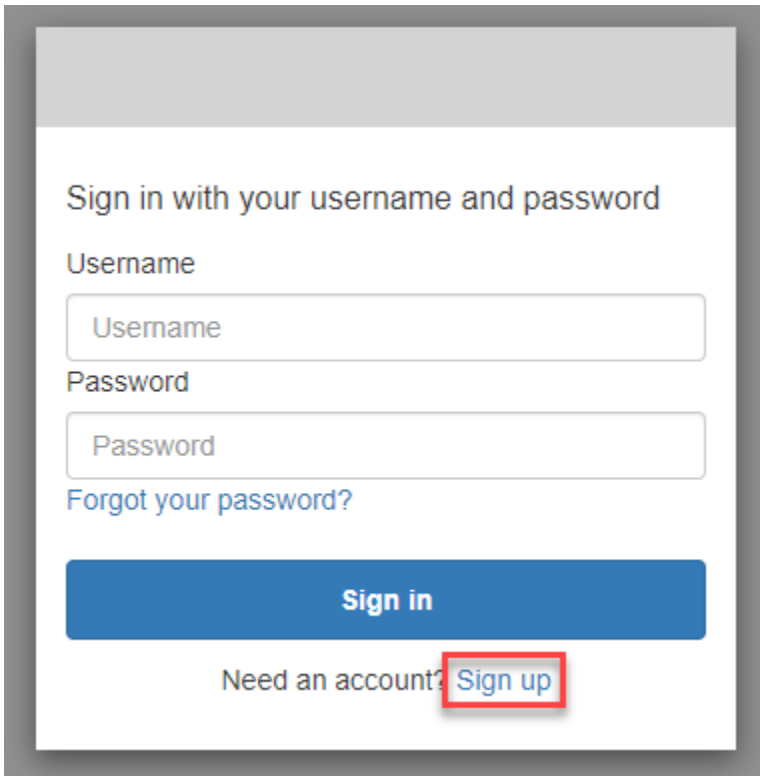
or

[Forgot your password?](#)

Sign in

Need an account? [Sign up](#)

With only Amazon Cognito as a sign-in provider



Sign in with your username and password

Username

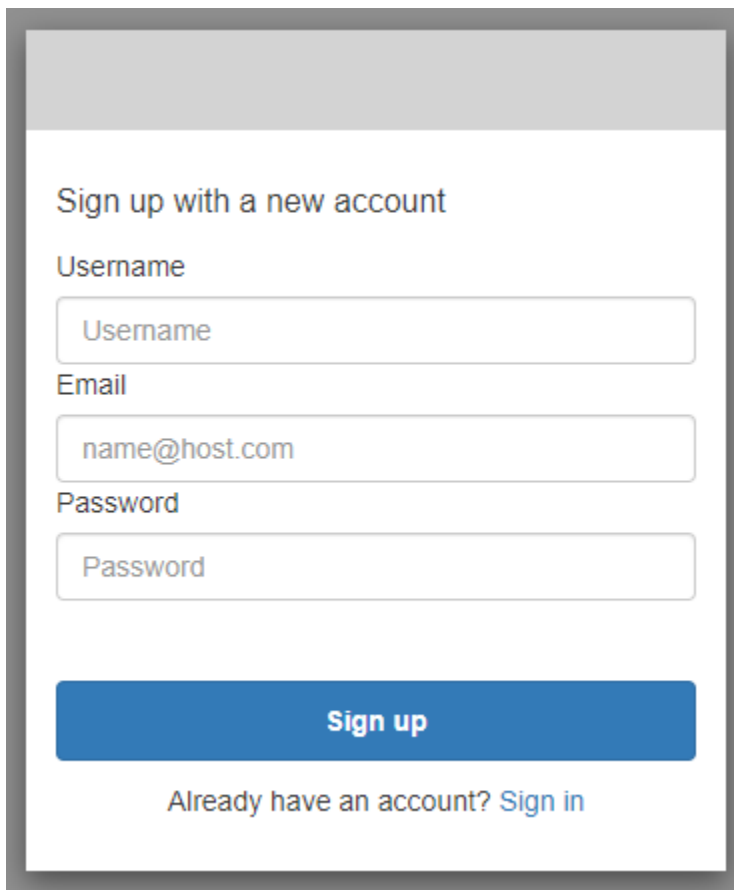
Password

[Forgot your password?](#)

Sign in

Need an account? [Sign up](#)

2. Sur la page Sign up with a new account (S'inscrire avec un nouveau compte), le propriétaire de l'application demande les informations dont il a besoin pour l'inscription. Il peut vous demander un nom d'utilisateur, une adresse e-mail ou un numéro de téléphone. Entrez les informations requises et choisissez un mot de passe.



Sign up with a new account

Username

Email

Password

Sign up

Already have an account? [Sign in](#)

3. Sur la page Confirm your account (Confirmer votre compte), le propriétaire de l'application peut vous demander de confirmer votre compte afin de vérifier que vous pouvez recevoir des messages à l'adresse e-mail ou au numéro de téléphone que vous avez fournis.

Vous recevrez un code par e-mail ou par SMS. Entrez le code dans le formulaire pour confirmer que vous avez saisi les bonnes coordonnées.

Confirm your account

We have sent a code by email to [redacted]@[redacted]. Enter it below to confirm your account.

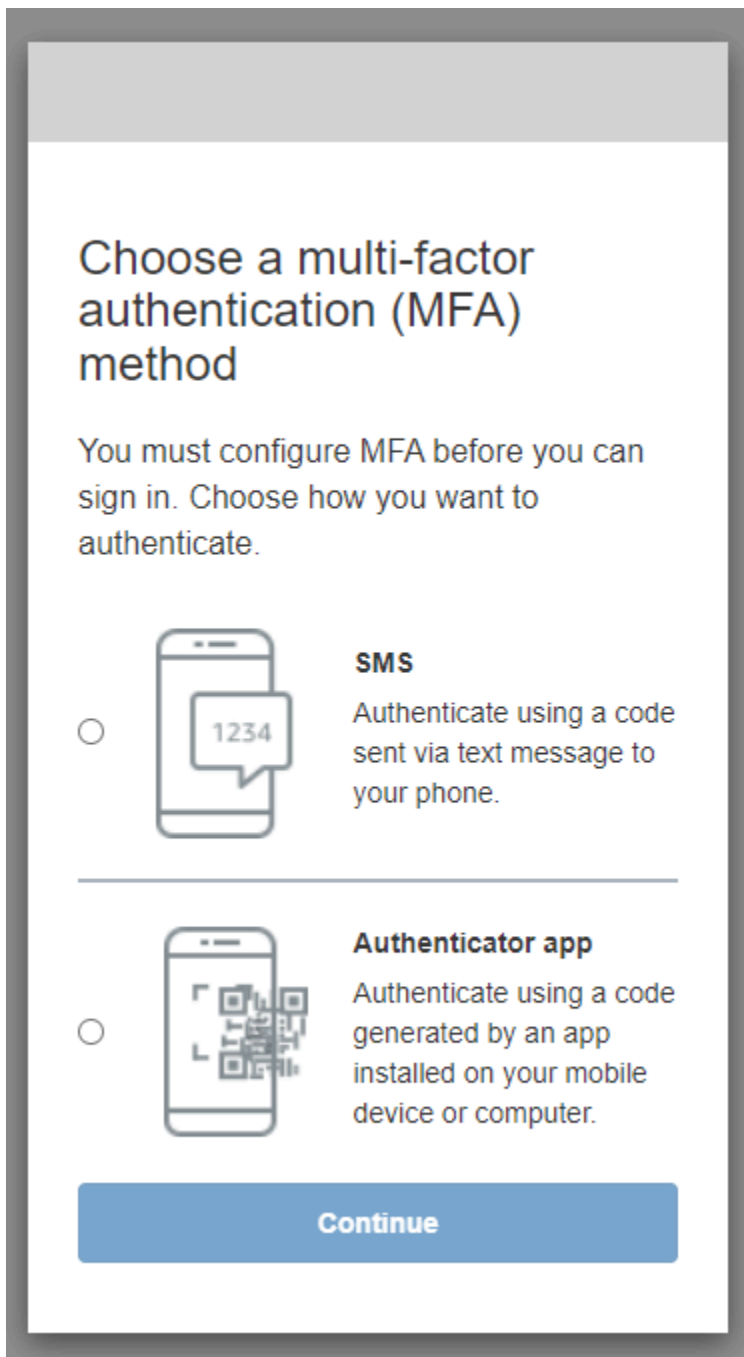
Verification code

Confirm account

Didn't receive a code? [Send a new code](#)

4. Le propriétaire de l'application peut vous demander de configurer l'authentification multifactorielle (MFA). Il se peut que vous soyez invité à choisir votre méthode MFA ou que votre application passe à l'étape suivante.

Sur la page Choose a multi-factor authentication (MFA) method (Choisir une méthode d'authentification multifactorielle [MFA]), choisissez une méthode MFA. Si vous choisissez SMS, vous recevrez des codes d'accès MFA sous forme de SMS. Si vous choisissez Authenticator app (Application d'authentification), vous devez installer une application sur votre appareil pour générer des codes d'accès MFA à durée limitée. Vous devez faire votre choix dans les 3 minutes.



5. Amazon Cognito vous demande un code depuis votre application d'authentification ou par SMS. Saisissez le code que vous avez reçu dans les 3 minutes.


Authenticator app

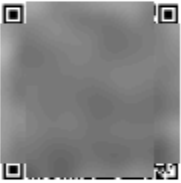
1. Ouvrez l'application d'authentification que vous avez téléchargée.
2. Scannez le code QR sur la page avec votre appareil photo. Il se peut que vous deviez autoriser l'application à utiliser votre appareil photo.

Si vous ne parvenez pas à scanner le code QR, choisissez Show secret key (Afficher la clé secrète) pour afficher un code que vous pouvez saisir manuellement dans votre application d'authentification.

3. Votre application d'authentification affiche des codes qui changent au bout de quelques secondes. Entrez un code actuel depuis l'application.
4. (Facultatif) Sur la page Set up authenticator app MFA (Configurer l'application d'authentification MFA), choisissez un nom pour votre appareil. Lorsque vous vous connectez, Amazon Cognito vous demande de saisir un code sur l'appareil dont vous avez indiqué le nom ici.

Set up authenticator app MFA

- 

1 Install an authenticator app on your mobile device.
- 

2 Scan this QR code with your authenticator app. Alternatively, you can manually enter a secret key in your authenticator app.

[Show secret key](#)
- 3 Enter a code from your authenticator app

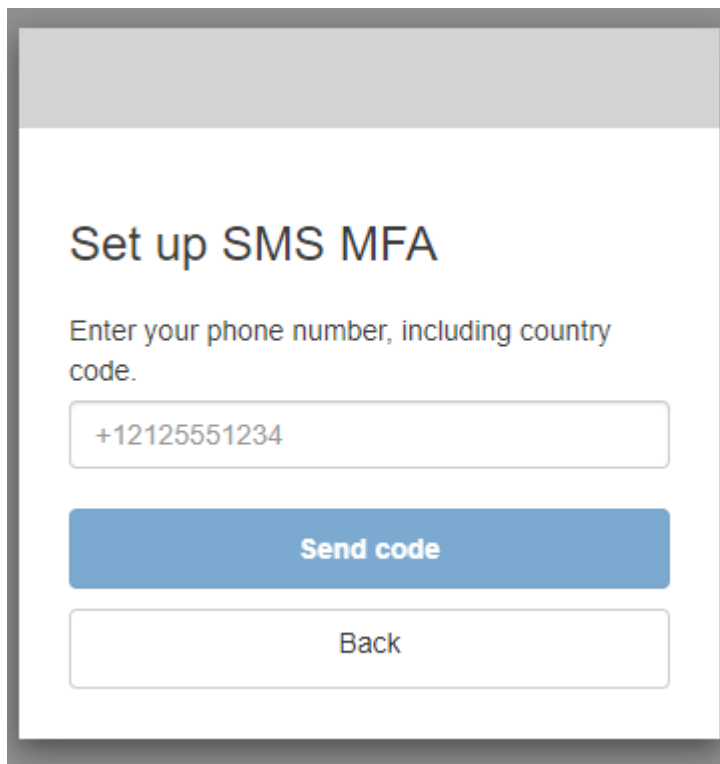
Enter a friendly device name - optional

Sign in

SMS text message

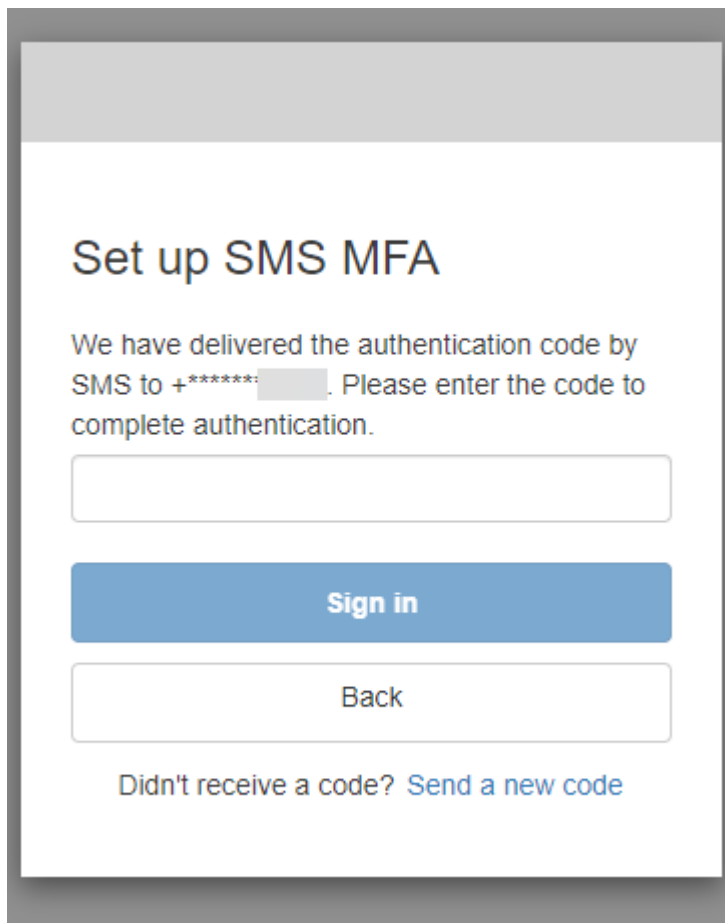
1. Si le propriétaire de l'application n'a pas encore collecté votre numéro de téléphone, Amazon Cognito demande votre numéro de téléphone.

Sur la page Set up SMS MFA (Configurer l'authentification MFA par SMS), entrez un numéro de téléphone comprenant un signe + et un code de pays, par exemple +12125551234.



The screenshot shows a mobile-style interface for setting up SMS MFA. At the top, the title 'Set up SMS MFA' is displayed in a large, dark font. Below the title, a prompt reads 'Enter your phone number, including country code.' A text input field contains the phone number '+12125551234'. Below the input field, there are two buttons: a blue button labeled 'Send code' and a white button labeled 'Back'.

2. Vous recevrez un SMS contenant un code. Sur la page Set up SMS MFA (Configurer l'authentification MFA par SMS), entrez le code. Si vous n'avez pas reçu de code et que vous souhaitez réessayer, choisissez Send a new code (Envoyer un nouveau code). Sélectionnez Back (Retour) pour saisir un nouveau numéro de téléphone.



6. La première fois que vous vous inscrivez et que vous confirmez vos informations, Amazon Cognito autorise l'accès à votre application une fois ce processus terminé.

Comment se connecter à l'interface utilisateur hébergée par Amazon Cognito

Ce guide vous montre comment vous connecter aux applications qui utilisent Amazon Cognito.

Note

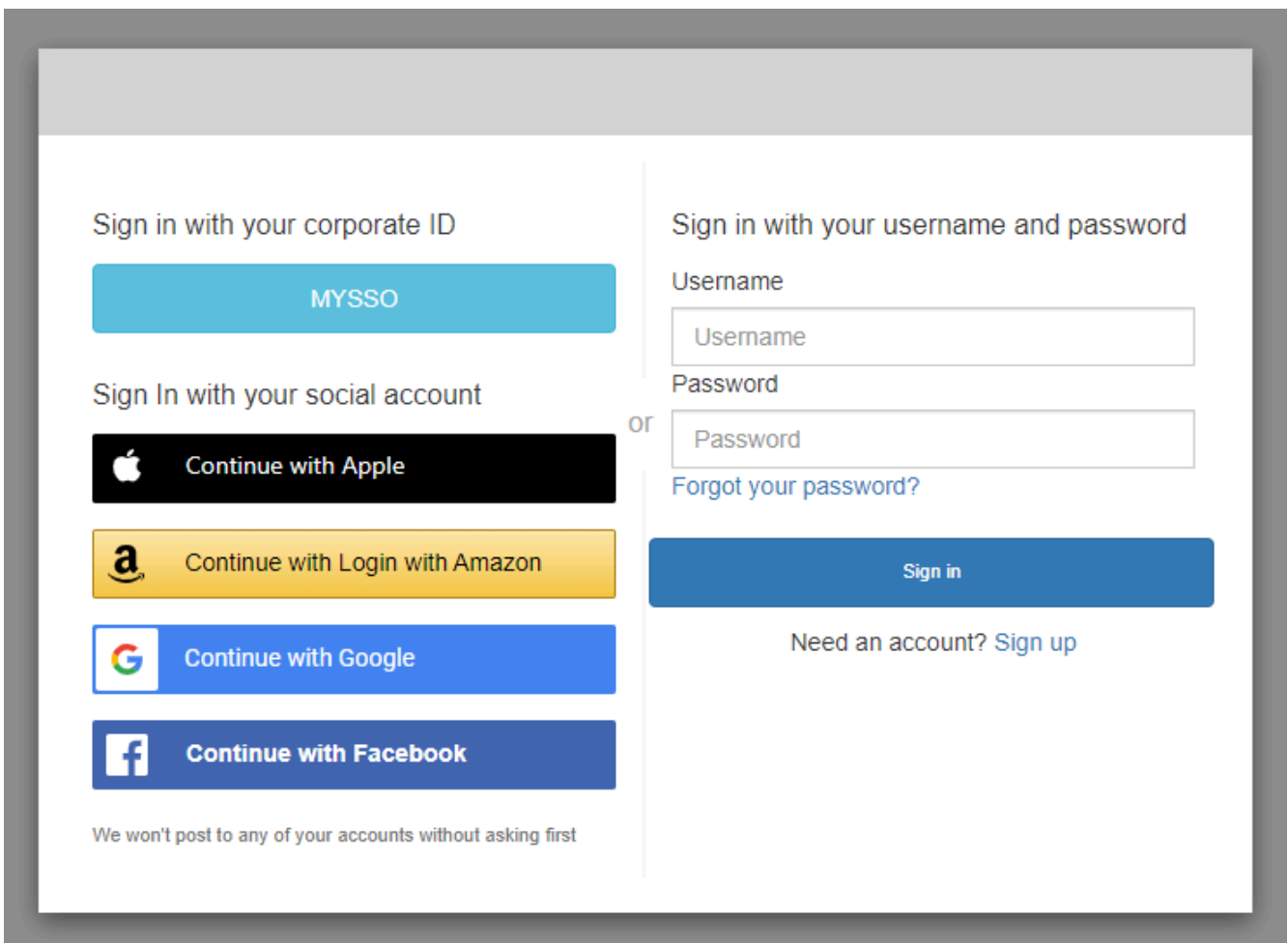
Lorsque vous vous connectez à une application qui utilise l'interface utilisateur hébergée par Amazon Cognito, vous pouvez voir une page que le propriétaire de l'application a personnalisée au-delà de la configuration de base présentée dans ce guide.

1. En fonction des options définies par le propriétaire de l'application, vous pouvez choisir entre différents fournisseurs pour vous connecter ou vous pouvez uniquement voir s'afficher un message demandant un nom d'utilisateur et un mot de passe. Lorsque vous vous connectez

à l'aide d'un nom d'utilisateur et d'un mot de passe à partir de cette page, Amazon Cognito est votre fournisseur de connexion. Dans le cas contraire, votre fournisseur de connexion est représenté par le bouton que vous choisissez.

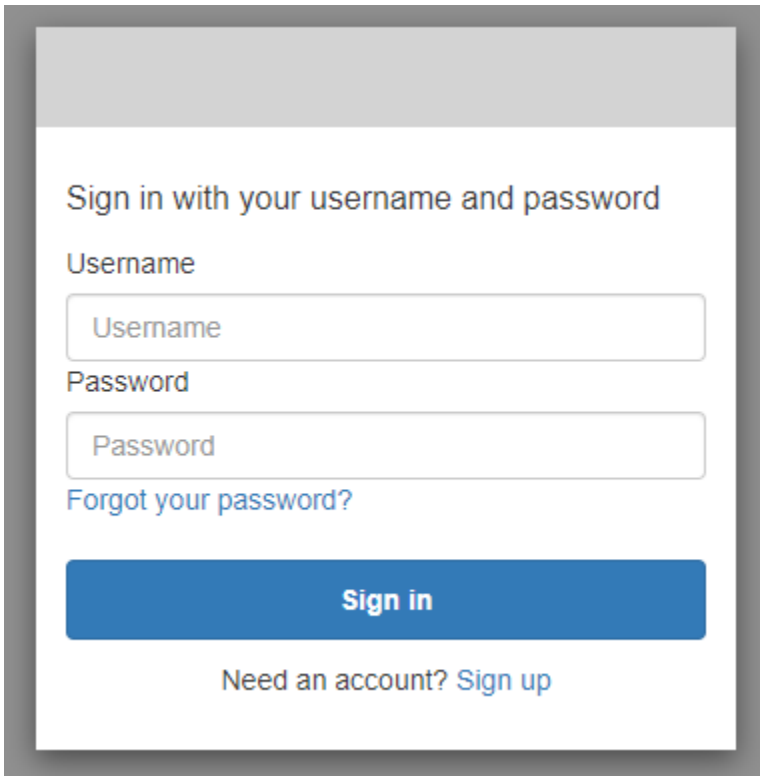
Vous pouvez choisir un fournisseur ici ou saisir un nom d'utilisateur et un mot de passe pour accéder immédiatement à votre application. Si Amazon Cognito est votre fournisseur de connexion, le propriétaire de l'application peut également avoir besoin d'une authentification multifactorielle.

With multiple sign-in providers



The image shows a sign-in interface with two main sections. The left section is titled "Sign in with your corporate ID" and features a blue button labeled "MYSSO". Below this is the heading "Sign In with your social account" followed by four buttons: "Continue with Apple" (black), "Continue with Login with Amazon" (yellow), "Continue with Google" (blue), and "Continue with Facebook" (dark blue). A small text note at the bottom of this section reads "We won't post to any of your accounts without asking first". The right section is titled "Sign in with your username and password" and contains a "Username" input field, a "Password" input field, and a "Forgot your password?" link. A blue "Sign in" button is positioned below the password field. A vertical line separates the two sections, with the word "or" centered between them.

With only Amazon Cognito as a sign-in provider



Sign in with your username and password

Username

Password

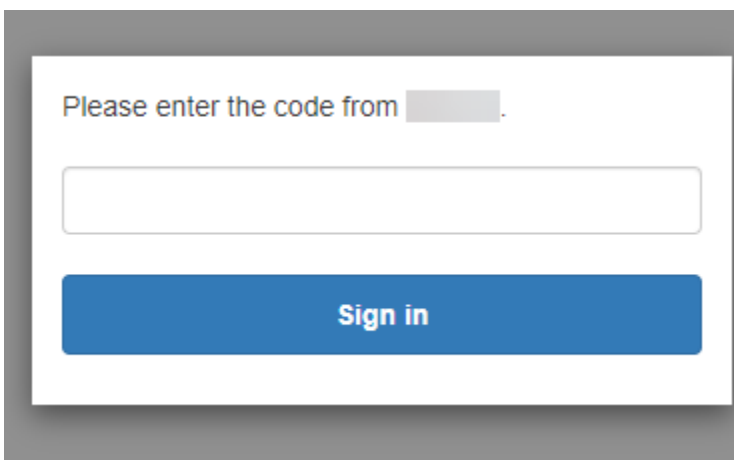
[Forgot your password?](#)

Sign in

Need an account? [Sign up](#)

2. Vous avez peut-être configuré l'authentification MFA lorsque vous vous êtes inscrit à l'application. Entrez le code d'authentification MFA que vous avez reçu par SMS ou qui est affiché dans votre application d'authentification. Vous devez saisir ce code dans les 3 minutes.

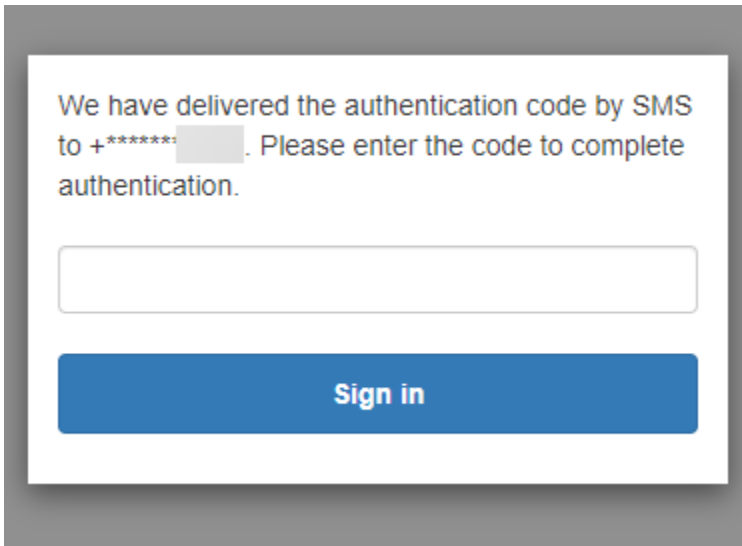
With an authenticator app



Please enter the code from .

Sign in

With an SMS code



3. Une fois que vous êtes connecté et que vous avez terminé l'authentification MFA, Amazon Cognito donne accès à votre application.

Comment réinitialiser un mot de passe avec l'interface utilisateur hébergée par Amazon Cognito

Ce guide vous montre comment réinitialiser votre mot de passe dans les applications qui utilisent Amazon Cognito.

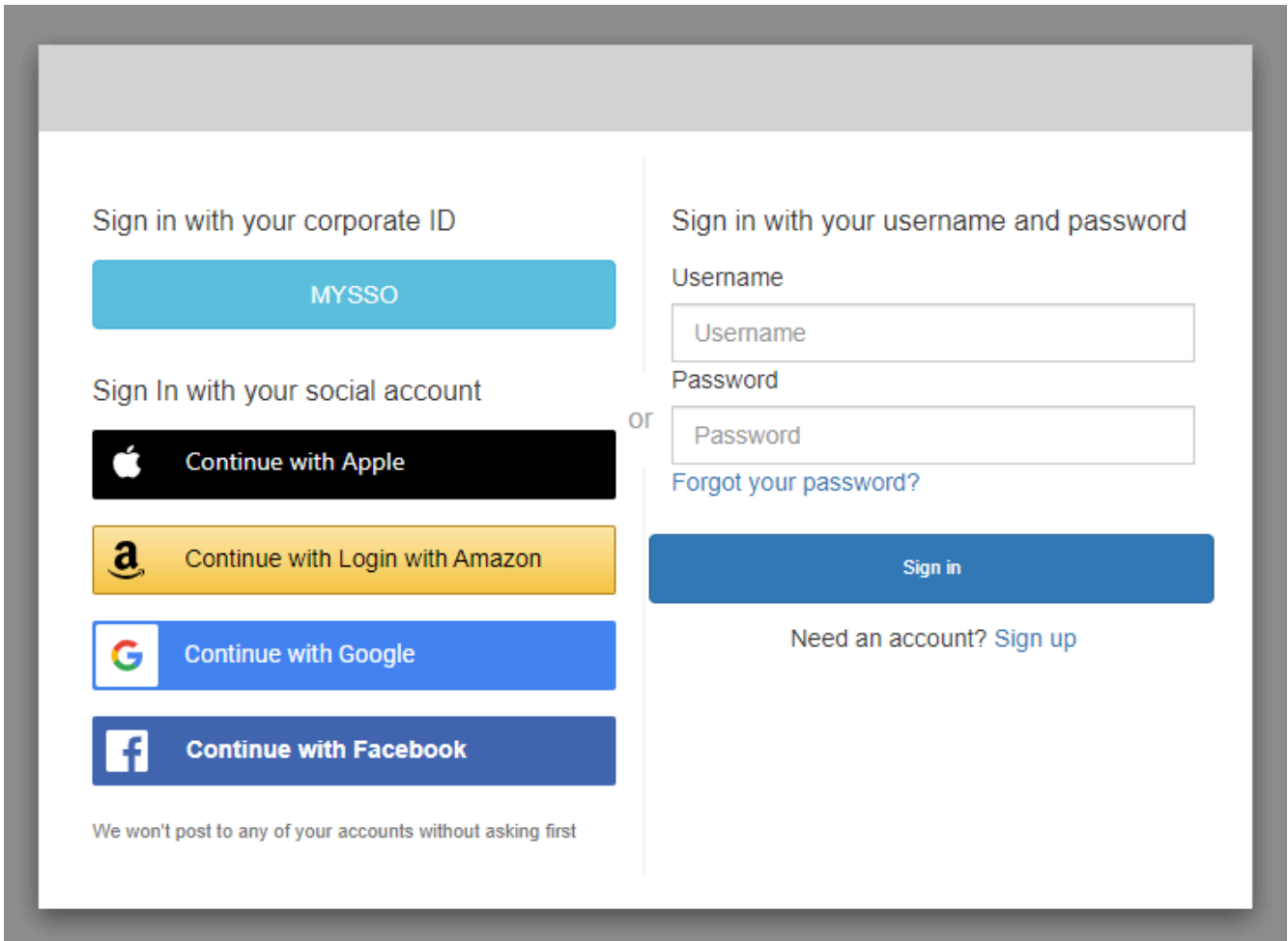
Note

Lorsque vous vous connectez à une application qui utilise l'interface utilisateur hébergée par Amazon Cognito, vous pouvez voir une page que le propriétaire de l'application a personnalisée au-delà de la configuration de base présentée dans ce guide.

1. En fonction des options définies par le propriétaire de l'application, vous pouvez choisir entre différents fournisseurs pour vous connecter ou vous pouvez uniquement voir s'afficher un message demandant un nom d'utilisateur et un mot de passe. Lorsque vous vous connectez à l'aide d'un nom d'utilisateur et d'un mot de passe à partir de cette page, Amazon Cognito est votre fournisseur de connexion. Dans le cas contraire, votre fournisseur de connexion est représenté par le bouton que vous choisissez.

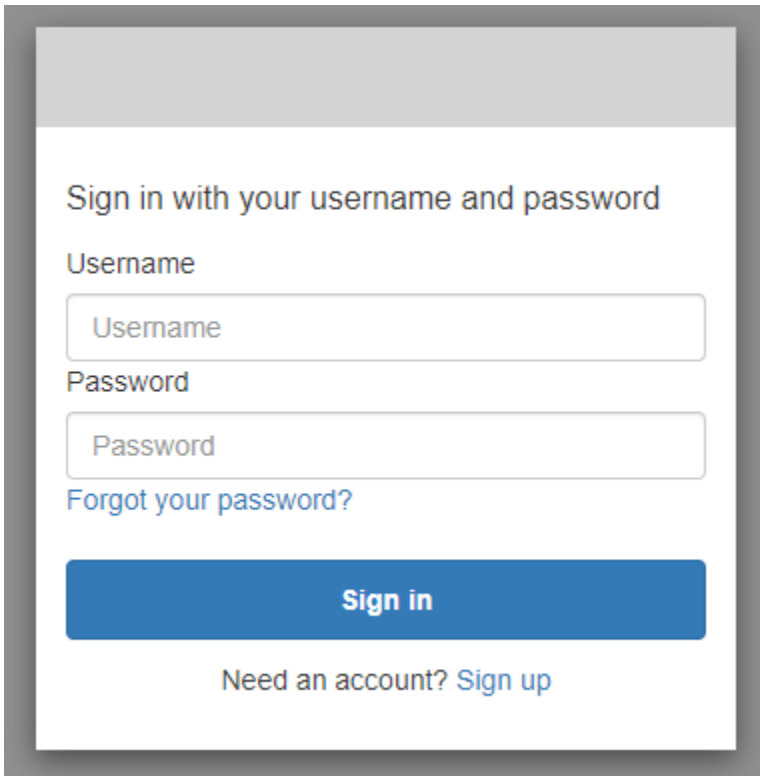
Si vous choisissez normalement un fournisseur à partir de la page de connexion et que votre mot de passe ne fonctionne pas, suivez la procédure pour réinitialiser votre mot de passe auprès du fournisseur. Si Amazon Cognito est votre fournisseur de connexion, choisissez **Forgot your password?** (Mot de passe oublié ?)

With multiple sign-in providers



The image shows a sign-in interface with two main sections. On the left, under "Sign in with your corporate ID", there is a blue button labeled "MYSSO". Below that, under "Sign In with your social account", there are four buttons: "Continue with Apple" (black), "Continue with Login with Amazon" (yellow), "Continue with Google" (blue), and "Continue with Facebook" (dark blue). At the bottom of this section, it says "We won't post to any of your accounts without asking first". On the right, under "Sign in with your username and password", there are input fields for "Username" and "Password", separated by an "or" label. Below the password field is a link "Forgot your password?". At the bottom of the right section is a blue "Sign in" button and a link "Need an account? Sign up".

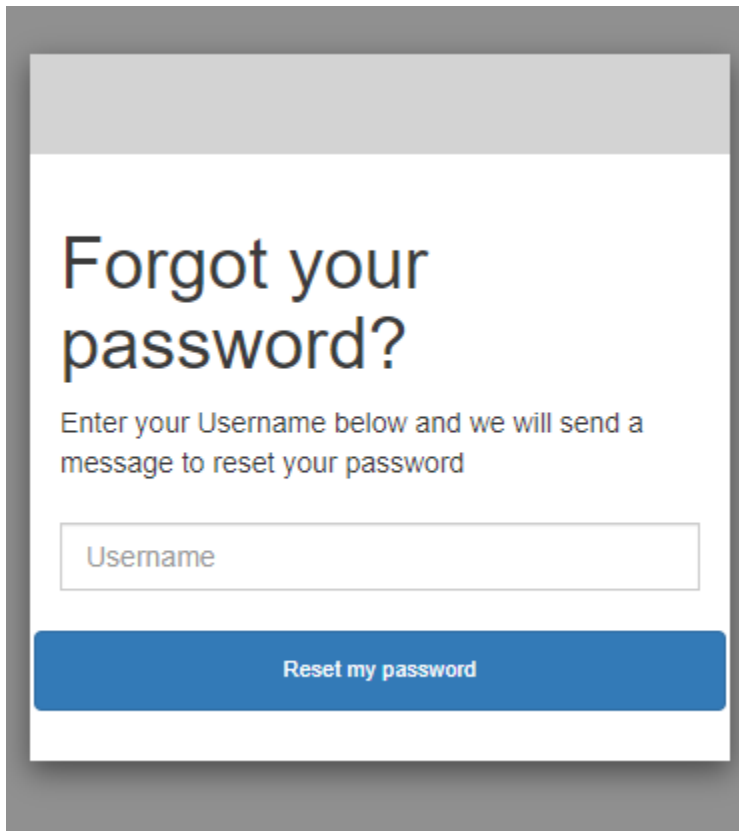
With only Amazon Cognito as a sign-in provider



The image shows a sign-in form with the following elements:

- Header: Sign in with your username and password
- Username field: A text input box with the placeholder text "Username".
- Password field: A text input box with the placeholder text "Password".
- Forgot your password? link: A blue text link.
- Sign in button: A blue button with the text "Sign in".
- Need an account? Sign up link: A blue text link.

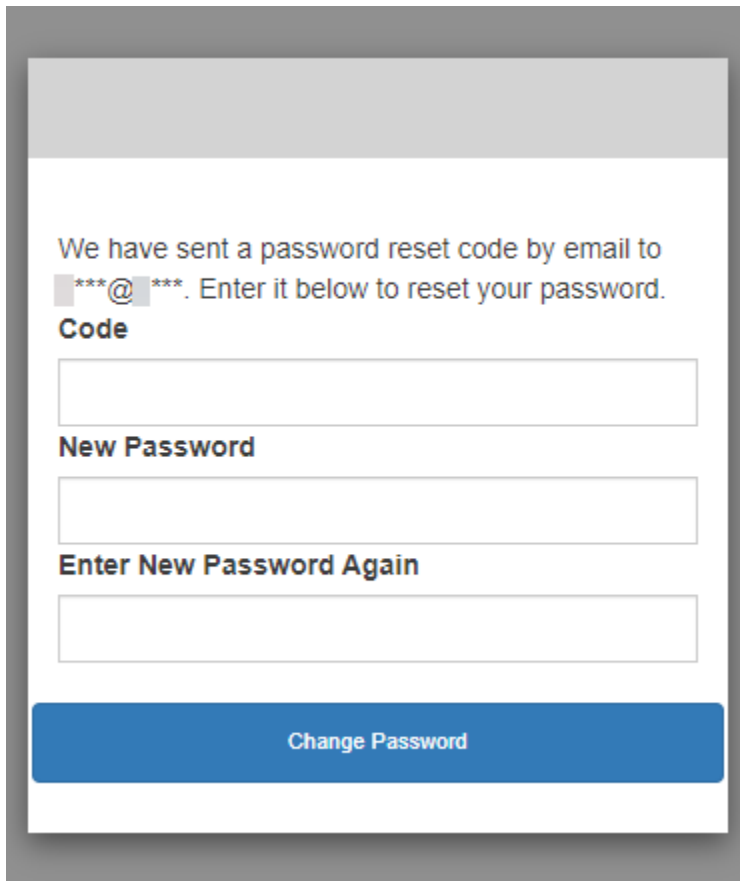
2. Sur la page [Forgot your password?](#) (Mot de passe oublié ?), Amazon Cognito vous invite à saisir les informations que vous utilisez pour vous connecter. Il peut s'agir de votre nom d'utilisateur, de votre adresse e-mail ou de votre numéro de téléphone.



The image shows a user interface for password reset. It features a large heading 'Forgot your password?' followed by a sub-heading 'Enter your Username below and we will send a message to reset your password'. Below this is a text input field with the placeholder text 'Username'. At the bottom of the form is a blue button labeled 'Reset my password'.

3. Amazon Cognito vous enverra un code par e-mail ou par SMS.

Entrez le code que vous avez reçu et entrez votre nouveau mot de passe deux fois dans les champs prévus à cet effet. Vous devez saisir votre code réinitialisé dans les 8 minutes.



The screenshot shows a mobile-style interface for password reset. At the top, a message reads: "We have sent a password reset code by email to [redacted]@[redacted]. Enter it below to reset your password." Below this, there are three input fields: "Code", "New Password", and "Enter New Password Again". At the bottom, there is a blue button labeled "Change Password".

- Après avoir modifié votre mot de passe, revenez à la page de connexion et connectez-vous avec votre nouveau mot de passe.

Champs d'application, M2M et autorisation d'API avec les serveurs de ressources

Une fois que vous avez configuré un domaine pour votre groupe d'utilisateurs, Amazon Cognito provisionne automatiquement un serveur d'autorisation OAuth 2.0 et une interface utilisateur Web hébergée avec des pages d'inscription et de connexion que votre application présentera à vos utilisateurs. Pour plus d'informations, consultez [Ajouter un client d'application avec l'interface utilisateur hébergée](#). Vous pouvez choisir les portées que vous souhaitez que le serveur d'autorisation ajoute aux jetons d'accès. Les portées autorisent l'accès aux serveurs de ressources et aux données utilisateur.

Un serveur de ressources est un [serveur d'API OAuth 2.0](#). Pour sécuriser les ressources à accès protégé, il vérifie que les jetons d'accès de votre groupe d'utilisateurs contiennent les portées autorisant la méthode et le chemin demandés dans l'API qu'il protège. Il vérifie l'émetteur en fonction

de la signature du jeton, la validité en fonction de l'heure d'expiration du jeton et le niveau d'accès en fonction des portées des enregistrements du jeton. Les étendues du groupe d'utilisateurs figurent dans la scope réclamation du jeton d'accès. Pour plus d'informations sur les enregistrements relatifs aux jetons d'accès Amazon Cognito, consultez [Utilisation du jeton d'accès](#).

Avec Amazon Cognito, les étendues des jetons d'accès peuvent autoriser l'accès à des API externes ou à des attributs utilisateur. Vous pouvez émettre des jetons d'accès à des utilisateurs locaux, à des utilisateurs fédérés ou à des identités de machine.

Autorisation Machine-to-machine (M2M)

Amazon Cognito prend en charge les applications qui accèdent aux données d'API avec des identités de machine. Les identités des machines dans les groupes d'utilisateurs sont des [clients confidentiels](#) qui s'exécutent sur des serveurs d'applications et se connectent à des API distantes. Leur fonctionnement s'effectue sans interaction de l'utilisateur : tâches planifiées, flux de données ou mises à jour des actifs. Lorsque ces clients autorisent leurs demandes à l'aide d'un jeton d'accès, ils effectuent une autorisation machine à machine, ou M2M. Dans le cadre de l'autorisation M2M, un secret partagé remplace les informations d'identification de l'utilisateur dans le contrôle d'accès.

Une application qui accède à une API avec une autorisation M2M doit disposer d'un identifiant client et d'un secret client. Dans votre groupe d'utilisateurs, vous devez créer un client d'application qui prend en charge les autorisations d'identification des clients. Pour prendre en charge les informations d'identification des clients, le client de votre application doit disposer d'un secret client et vous devez disposer d'un domaine de groupe d'utilisateurs. Dans ce flux, l'identité de votre machine demande un jeton d'accès directement auprès du [Point de terminaison de jeton](#). Vous ne pouvez autoriser que des étendues personnalisées provenant de [serveurs de ressources](#) dans des jetons d'accès pour l'attribution des informations d'identification des clients. Pour plus d'informations sur la configuration des clients d'applications, consultez [Clients de l'application de groupe d'utilisateurs](#).

Le jeton d'accès issu de l'octroi d'informations d'identification d'un client est une déclaration vérifiable des opérations que vous souhaitez autoriser à demander l'identité de votre machine à une API. Pour en savoir plus sur la façon dont les jetons d'accès autorisent les demandes d'API, poursuivez votre lecture. Pour un exemple d'application, consultez [Amazon Cognito et l'autorisation machine à machine basée sur API Gateway à l'aide AWS du CDK](#).

L'autorisation M2M repose sur un modèle de facturation différent de la façon dont les utilisateurs actifs mensuels (MAU) sont facturés. Lorsque l'authentification des utilisateurs entraîne un coût par utilisateur actif, la facturation M2M reflète les informations d'identification des clients actifs,

les clients des applications et le volume total de demandes de jetons. Pour plus d'informations, consultez [Tarification d'Amazon Cognito](#). Pour contrôler les coûts d'autorisation M2M, optimisez la durée des jetons d'accès et le nombre de demandes de jetons effectuées par vos applications. Découvrez comment utiliser [Jetons de mise en cache](#) la mise en cache d'API Gateway pour réduire les demandes de nouveaux jetons dans le cadre des autorisations M2M.

Pour plus d'informations sur l'optimisation des opérations Amazon Cognito qui ajoutent des coûts à votre AWS facture, consultez. [Gestion des coûts](#)

À propos des portées

Une portée est un niveau d'accès qu'une application peut demander à une ressource. Dans un jeton d'accès Amazon Cognito, la portée est renforcée par la confiance que vous avez établie avec votre groupe d'utilisateurs : un émetteur fiable de jetons d'accès doté d'une signature numérique connue. Les groupes d'utilisateurs peuvent générer des jetons d'accès dont les portées prouvent que votre client est autorisé à gérer tout ou partie de son propre profil utilisateur, ou à récupérer des données depuis une API principale. Les groupes d'utilisateurs Amazon Cognito émettent des jetons d'accès avec la portée d'API réservée aux groupes d'utilisateurs, des portées personnalisées et des portées standard.

Champ d'application de l'API réservé aux groupes d'utilisateurs

La portée `aws.cognito.signin.user.admin` autorise l'API des groupes d'utilisateurs Amazon Cognito. Il autorise le porteur d'un jeton d'accès à interroger et à mettre à jour toutes les informations concernant un utilisateur du pool d'utilisateurs avec, par exemple, les opérations de l'[UpdateUserAttributes](#) API [GetUser](#) et. Lorsque vous authentifiez votre utilisateur avec l'API des groupes d'utilisateurs Amazon Cognito, il s'agit de la seule portée que vous recevez dans le jeton d'accès. C'est également la seule portée dont vous avez besoin pour lire et écrire les attributs utilisateur pour lesquels vous avez donné des droits de lecture et d'écriture à votre client d'application. Vous pouvez également demander cette portée dans les demandes adressées à votre [Point de terminaison d'autorisation](#). La portée seule ne suffit pas à demander les attributs utilisateur au [Point de terminaison UserInfo](#). Pour les jetons d'accès qui autorisent à la fois l'API des groupes d'utilisateurs et les demandes `userInfo` pour vos utilisateurs, vous devez demander à la fois les portées `openid` et `aws.cognito.signin.user.admin` dans une demande `/oauth2/authorize`.

Portées personnalisées

Les portées personnalisées autorisent les demandes adressées aux API externes protégées par les serveurs de ressources. Vous pouvez demander des portées personnalisées avec d'autres types de portées. Vous trouverez plus d'informations sur les portées personnalisées sur cette page.

Portées standard

Lorsque vous authentifiez des utilisateurs auprès du serveur d'autorisation OAuth 2.0 de votre groupe d'utilisateurs, y compris avec l'interface utilisateur hébergée, vous devez demander des portées. Vous pouvez authentifier les utilisateurs locaux du groupe d'utilisateurs et les utilisateurs fédérés tiers sur votre serveur d'autorisation Amazon Cognito. Les portées OAuth 2.0 standard autorisent votre application à lire les informations utilisateur du [Point de terminaison UserInfo](#) de votre groupe d'utilisateurs. Le modèle OAuth, dans lequel vous interrogez les attributs utilisateur depuis le point de terminaison `userInfo`, permet d'optimiser votre application pour un volume élevé de demandes d'attributs utilisateur. Le point de terminaison `userInfo` renvoie des attributs à un niveau d'autorisation déterminé par les portées du jeton d'accès. Vous pouvez autoriser votre client d'application à émettre des jetons d'accès avec les portées OAuth 2.0 standard suivantes.

openid

Portée minimale pour les requêtes OpenID Connect (OIDC). Autorise le jeton d'identification, l'enregistrement d'identifiant unique `sub` et la possibilité de demander d'autres portées.

Note

Lorsque vous demandez la portée `openid` et aucune autre, votre jeton d'identification de groupe d'utilisateurs et votre réponse `userInfo` incluent les demandes de tous les attributs utilisateur que votre client d'application peut lire. Lorsque vous demandez `openid` et d'autres portées standard telles que `profile`, `email` et `phone`, le contenu du jeton d'identification et de la réponse [userInfo](#) est limité aux contraintes des portées supplémentaires.

Par exemple, une requête adressée au [Point de terminaison d'autorisation](#) avec le paramètre `scope=openid+email` renvoie un jeton d'identification avec `sub`, `email` et `email_verified`. Le jeton d'accès issu de cette demande renvoie les mêmes attributs depuis [Point de terminaison UserInfo](#). Une demande avec le paramètre `scope=openid` renvoie tous les attributs lisibles par le client dans le jeton d'identification et depuis `userInfo`.

profile

Autorise tous les attributs utilisateur que le client d'application peut lire.

e-mail

Autorise les attributs utilisateur `email` et `email_verified`. Amazon Cognito renvoie une valeur `email_verified` si une valeur a été définie de manière explicite.

phone

Autorise les attributs utilisateur `phone_number` et `phone_number_verified`.

À propos des serveurs de ressources

Une API de serveur de ressources peut autoriser l'accès aux informations d'une base de données ou contrôler vos ressources informatiques. Un jeton d'accès Amazon Cognito peut autoriser l'accès aux API qui prennent en charge le protocole OAuth 2.0. Les API REST Amazon API Gateway disposent d'une [prise en charge intégrée](#) des autorisations avec jetons d'accès Amazon Cognito. Votre application transmet le jeton d'accès dans l'appel d'API au serveur de ressources. Le serveur de ressources inspecte le jeton d'accès pour déterminer si l'accès doit être accordé.

Amazon Cognito pourrait apporter de futures mises à jour au schéma des jetons d'accès du groupe d'utilisateurs. Si votre application analyse le contenu du jeton d'accès avant de le transmettre à une API, vous devez concevoir votre code pour accepter les mises à jour du schéma.

Vous définissez les portées personnalisées, qui étendent les capacités d'autorisation d'un groupe d'utilisateurs pour inclure des objectifs non liés à l'interrogation et à la modification des utilisateurs et de leurs attributs. Par exemple, si vous disposez d'un serveur de ressources pour les photos, il peut définir deux portées : `photos.read` pour lire l'accès aux photos et `photos.write` pour écrire/supprimer l'accès. Vous pouvez configurer une API pour accepter des jetons d'accès à des fins d'autorisation et accorder des demandes d'accès à des jetons HTTP GET avec `photos.read` dans la revendication `scope`, et des demandes HTTP POST adressées à des jetons avec `photos.write`. Il s'agit de portées personnalisées.

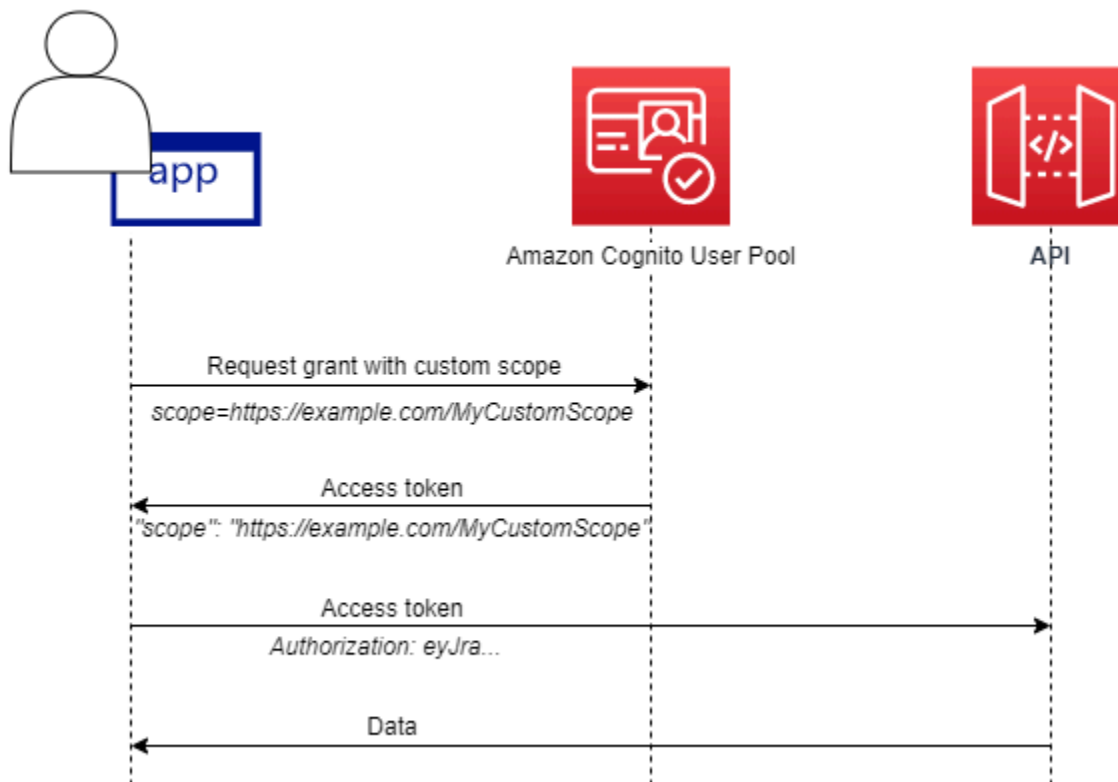
Note

Votre serveur de ressources doit vérifier la signature et la date d'expiration du jeton d'accès avant de traiter toute réclamation à l'intérieur du jeton. Pour plus d'informations sur la vérification des jetons, consultez [Vérification d'un jeton web JSON](#). Pour plus d'informations sur la vérification et l'utilisation des jetons de groupe d'utilisateurs dans Amazon API

Gateway, consultez le blog [Intégration des groupes d'utilisateurs Amazon Cognito avec API Gateway](#). API Gateway est une bonne option pour inspecter les jetons d'accès et protéger vos ressources. Pour plus d'informations sur les mécanismes d'autorisation Lambda API Gateway, consultez [Utilisation des mécanismes d'autorisation Lambda API Gateway](#).

Présentation

Amazon Cognito vous permet de créer des Serveurs de ressources OAuth 2.0 et d'y associer des Portées personnalisées. Les portées personnalisées d'un jeton d'accès autorisent des actions spécifiques dans votre API. Vous pouvez autoriser n'importe quel client d'application de votre groupe d'utilisateurs à émettre des portées personnalisées à partir de n'importe lequel de vos serveurs de ressources. Associez vos portées personnalisées à un client d'application et demandez ces portées dans le cadre des octrois de code d'autorisation OAuth 2.0, des octrois implicites et des octrois d'informations d'identification client à partir du [Point de terminaison de jeton](#). Amazon Cognito ajoute des portées personnalisées à la revendication scope d'un jeton d'accès. Un client peut utiliser le jeton d'accès sur son serveur de ressources, ce qui base la décision d'autorisation sur les périmètres présents dans le jeton. Pour plus d'informations sur la portée du jeton d'accès, consultez [Utilisation de jetons avec des groupes d'utilisateurs](#).



Pour obtenir un jeton d'accès avec des portées personnalisées, votre application doit adresser une demande au [Point de terminaison de jeton](#) pour utiliser un code d'autorisation ou pour demander un octroi d'informations d'identification client. Dans l'interface utilisateur hébergée, vous pouvez également demander des portées personnalisées dans un jeton d'accès à partir d'une autorisation implicite.

Note

Parce qu'elles sont conçues pour une authentification interactive avec le groupe d'utilisateurs comme IdP [InitiateAuth](#), [AdminInitiateAuth](#) et que les demandes ne produisent qu'une seule réclamation dans le jeton d'accès avec une valeur unique.

```
aws.cognito.signin.user.admin
```

Gestion du serveur de ressources et des périmètres personnalisés

Lors de la création d'un serveur de ressources, vous devez fournir un nom et un identifiant de serveur de ressources. Pour chaque portée que vous créez dans le serveur de ressources, vous devez fournir le nom et la description de la portée.

- Nom du serveur de ressources : nom convivial du serveur de ressources, tel que `Solar system object tracker` ou `Photo API`.
- Identifiant du serveur de ressources : identifiant unique du serveur de ressources. L'identifiant est un nom que vous pouvez associer à votre API, par exemple, `solar-system-data`. Vous pouvez configurer des identifiants plus longs, comme `https://solar-system-data-api.example.com`, en référence plus directe aux chemins d'URI des API, mais des chaînes plus longues augmentent la taille des jetons d'accès.
- Nom de la portée : valeur que vous souhaitez voir figurer dans vos revendications scope. Par exemple, `sunproximity.read`.
- Description : description conviviale de la portée. Par exemple, `Check current proximity to sun`.

Amazon Cognito peut inclure des portées personnalisées dans les jetons d'accès pour tous les utilisateurs, qu'ils soient locaux dans votre groupe d'utilisateurs ou fédérés avec un fournisseur d'identité tiers. Vous pouvez choisir les portées des jetons d'accès de vos utilisateurs lors des flux d'authentification avec le serveur d'autorisation OAuth 2.0 qui inclut l'interface utilisateur hébergée. L'authentification de votre utilisateur doit commencer au [Point de terminaison](#)

[d'autorisation](#) avec scope en tant que l'un des paramètres de la demande. Le format suivant est recommandé pour les serveurs de ressources. Pour un identifiant, utilisez un nom d'API convivial. Pour une portée personnalisée, utilisez l'action autorisée.

```
resourceServerIdentifiant/scopeName
```

Par exemple, vous avez découvert un nouvel astéroïde dans la ceinture de Kuiper et vous souhaitez l'enregistrer via votre API `solar-system-data`. La portée qui autorise les opérations d'écriture dans la base de données des astéroïdes est `asteroids.add`. Lorsque vous demandez le jeton d'accès qui vous autorisera à enregistrer votre découverte, indiquez le paramètre de votre demande HTTPS scope sous la forme `scope=solar-system-data/asteroids.add`.

La suppression d'une portée d'un serveur de ressources n'a pas pour effet de supprimer son association avec tous les clients. Au lieu de cela, la portée est marquée comme inactive. Amazon Cognito n'ajoute pas de portées inactives aux jetons d'accès, mais procède normalement si votre application en fait la demande. Si vous ajoutez à nouveau la portée à votre serveur de ressources ultérieurement, Amazon Cognito l'écrit à nouveau dans le jeton d'accès. Si vous demandez une portée que vous n'avez pas associée à votre client d'application, que vous l'ayez supprimée ou non du serveur de ressources de votre groupe d'utilisateurs, l'authentification échoue.

Vous pouvez utiliser l' AWS Management Console API ou la CLI pour définir les serveurs de ressources et les étendues de votre groupe d'utilisateurs.

Définition d'un serveur de ressources pour votre groupe d'utilisateurs (AWS Management Console)

Vous pouvez utiliser le AWS Management Console pour définir un serveur de ressources pour votre groupe d'utilisateurs.

Pour définir un serveur de ressources

1. Connectez-vous à la [console Amazon Cognito](#).
2. Dans le volet de navigation, choisissez Groupes d'utilisateurs, puis le groupe d'utilisateurs à modifier.
3. Choisissez l'onglet App integration (Intégration d'applications) et localisez Resource servers (Serveurs de ressources).
4. Choisissez Créer un serveur de ressources.

5. Saisissez un nom du serveur de ressources. Par exemple, `Photo Server`.
6. Saisissez un Identifiant de serveur de ressources. Par exemple, `com.example.photos`.
7. Saisissez les noms des périmètres personnalisés de vos ressources comme `read` et `write`.
8. Pour chacun des noms de portée, saisissez une description, telles que `view your photos` et `update your photos`.
9. Choisissez `Créer`.

Vos périmètres personnalisés peuvent être examinés dans l'onglet `App integration` (Intégration d'applications) sous `Resource servers` (Serveurs de ressources) dans `Custom scopes` (Périmètres personnalisés). Les périmètres personnalisés peuvent être activés pour les clients d'application à partir de l'onglet `App integration` (Intégration d'applications) sous `App clients` (Clients d'application). Sélectionnez un client d'application, recherchez `Paramètres d'interface utilisateur hébergée` et choisissez `Modifier`. Ajoutez des périmètres personnalisés et choisissez `Save changes` (Enregistrez les modifications).

Définition d'un serveur de ressources pour votre groupe d'utilisateurs (AWS CLI et AWS API)

Utilisez les commandes suivantes pour spécifier les paramètres de serveur de ressources pour votre groupe d'utilisateurs.

Pour créer un serveur de ressources

- AWS CLI: `aws cognito-idp create-resource-server`
- AWS API : [CreateResourceServer](#)

Pour obtenir des informations sur les paramètres de votre serveur de ressources

- AWS CLI: `aws cognito-idp describe-resource-server`
- AWS API : [DescribeResourceServer](#)

Pour répertorier des informations sur tous les serveurs de ressources pour votre groupe d'utilisateurs

- AWS CLI: `aws cognito-idp list-resource-servers`
- AWS API : [ListResourceServers](#)

Pour supprimer un serveur de ressources

- AWS CLI: `aws cognito-idp delete-resource-server`
- AWS API : [DeleteResourceServer](#)

Pour mettre à jour les paramètres d'un serveur de ressources

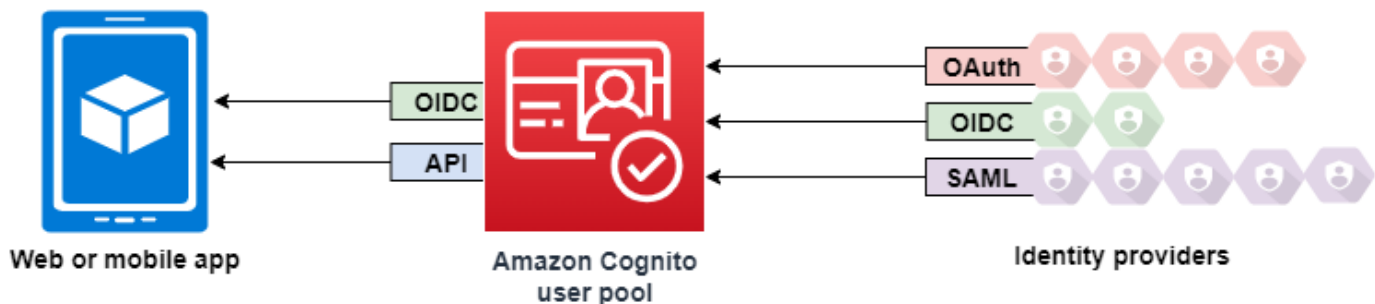
- AWS CLI: `aws cognito-idp update-resource-server`
- AWS API : [UpdateResourceServer](#)

Ajout de la connexion du groupe d'utilisateurs via un tiers

Les utilisateurs de votre application peuvent soit se connecter directement via un groupe d'utilisateurs, soit fédérer via un fournisseur d'identité (IdP) tiers. Le groupe d'utilisateurs gère les frais généraux liés à la gestion des jetons renvoyés lors de la connexion aux réseaux sociaux via Facebook, Google, Amazon et Apple, et depuis OpenID Connect (OIDC) et SAML. IdPs Grâce à l'interface utilisateur Web hébergée intégrée, Amazon Cognito permet le traitement et la gestion des jetons pour tous les utilisateurs authentifiés. IdPs Ainsi, vos systèmes backend peuvent ainsi utiliser un ensemble de jetons de groupe d'utilisateurs standard.

Fonctionnement de la connexion fédérée dans les groupes d'utilisateurs Amazon Cognito

La connexion via un tiers (fédération) est disponible dans les groupes d'utilisateurs Amazon Cognito. Cette fonctionnalité est indépendante de la fédération via les groupes d'identités Amazon Cognito (identités fédérées).



Amazon Cognito est un répertoire d'utilisateurs et un fournisseur d'identité (IdP) OAuth 2.0. Lorsque vous connectez des utilisateurs locaux dans l'annuaire Amazon Cognito, votre groupe d'utilisateurs

est un fournisseur d'identité pour votre application. Un utilisateur local existe exclusivement dans l'annuaire de votre groupe d'utilisateurs sans fédération via un fournisseur d'identité externe.

Lorsque vous connectez Amazon Cognito à un réseau social, à SAML ou à OpenID Connect (OIDC IdPs), votre groupe d'utilisateurs fait office de pont entre plusieurs fournisseurs de services et votre application. Pour votre fournisseur d'identité, Amazon Cognito est un fournisseur de services. Vous IdPs transmettez un jeton d'identification OIDC ou une assertion SAML à Amazon Cognito. Amazon Cognito lit les revendications concernant votre utilisateur dans le jeton ou l'assertion et les associe à un nouveau profil d'utilisateur dans le répertoire de votre groupe d'utilisateurs.

Amazon Cognito crée ensuite un profil utilisateur pour votre utilisateur fédéré dans son propre répertoire. Amazon Cognito ajoute des attributs à votre utilisateur en fonction des revendications de votre fournisseur d'identité et, dans le cas des fournisseurs d'identité sociale ou basé sur OIDC, un point de terminaison `userInfo` public géré par un fournisseur d'identité. Les attributs de votre utilisateur changent dans votre groupe d'utilisateurs lorsqu'un attribut IdP mappé change. Vous pouvez également ajouter d'autres attributs indépendants de ceux du fournisseur d'identité.

Une fois qu'Amazon Cognito a créé un profil pour votre utilisateur fédéré, il change de fonction et se présente comme le fournisseur d'identité de votre application, qui est maintenant le fournisseur de services. Amazon Cognito est un fournisseur d'identité combiné basé sur OIDC et OAuth 2.0. Il génère des jetons d'accès, des jetons d'identification et des jetons d'actualisation. Pour plus d'informations sur les jetons, consultez [Utilisation des jetons avec des groupes d'utilisateurs](#).

Vous devez concevoir une application qui s'intègre à Amazon Cognito pour authentifier et autoriser vos utilisateurs, qu'ils soient fédérés ou locaux.

Responsabilités d'une application en tant que fournisseur de services Amazon Cognito

Vérification et traitement des informations contenues dans les jetons

Dans la plupart des scénarios, Amazon Cognito redirige votre utilisateur authentifié vers une URL d'application à laquelle il ajoute un code d'autorisation. Votre application [échange ce code](#) pour obtenir les jetons d'accès, d'identification et d'actualisation. Ensuite, elle doit [vérifier la validité des jetons](#) et fournir des informations à votre utilisateur en fonction des revendications contenues dans les jetons.

Réponse aux événements d'authentification avec les demandes d'API Amazon Cognito

Votre application doit s'intégrer à l'[API des groupes d'utilisateurs Amazon Cognito](#) et aux [points de terminaison de l'API d'authentification](#). L'API d'authentification connecte et déconnecte votre utilisateur, et gère les jetons. L'API des groupes d'utilisateurs fournit diverses opérations qui gèrent votre groupe d'utilisateurs, vos utilisateurs et la sécurité de votre environnement d'authentification. Votre application doit savoir ce qu'il faut faire ensuite lorsqu'elle reçoit une réponse d'Amazon Cognito.

Informations utiles concernant la connexion tierce aux groupes d'utilisateurs Amazon Cognito

- Si vous souhaitez que vos utilisateurs se connectent avec des fournisseurs fédérés, vous devez choisir un domaine. Cela permet de configurer l'interface utilisateur hébergée Amazon Cognito et les [points de terminaison OIDC et de l'interface utilisateur hébergée](#). Pour plus d'informations, consultez [Utilisation de votre propre domaine pour l'interface utilisateur hébergée](#).
- Vous ne pouvez pas connecter des utilisateurs fédérés à l'aide d'opérations d'API telles que [InitiateAuth](#) et [AdminInitiateAuth](#). Les utilisateurs fédérés peuvent se connecter uniquement avec le [Point de terminaison de connexion](#) ou le [Point de terminaison d'autorisation](#).
- Le [Point de terminaison d'autorisation](#) est un point de terminaison de redirection. Si vous fournissez un paramètre `idp_identifiant` ou `identity_provider` dans votre demande, il effectue une redirection silencieuse vers votre fournisseur d'identité, en contournant l'interface utilisateur hébergée. Sinon, il effectue une redirection vers le [Point de terminaison de connexion](#) de l'interface utilisateur hébergée. Pour obtenir un exemple, consultez [Exemple de scénario : ajouter des applications Amazon Cognito à vos favoris dans un tableau de bord d'entreprise](#).
- Lorsque l'interface utilisateur hébergée redirige une séance vers un fournisseur d'identité fédéré, Amazon Cognito inclut l'en-tête `user-agent Amazon/Cognito` dans la demande.
- Amazon Cognito dérive l'attribut `username` pour un profil d'utilisateur fédéré à partir d'une combinaison d'un identifiant fixe et du nom de votre fournisseur d'identité. Pour générer un nom d'utilisateur correspondant à vos exigences personnalisées, créez un mappage avec l'attribut `preferred_username`. Pour plus d'informations, consultez [Choses à savoir sur les mappages](#).

Exemple : `MyIDP_bob@example.com`

- Amazon Cognito enregistre les informations sur l'identité de votre utilisateur fédéré dans un attribut, et une revendication dans le jeton d'identification, appelée `identities`. Cette revendication contient le fournisseur de votre utilisateur et son identifiant unique provenant du fournisseur. Vous

ne pouvez pas modifier l'attribut `identities` directement dans un profil utilisateur. Pour plus d'informations sur la liaison d'un utilisateur fédéré, consultez [Liaison d'utilisateurs fédérés à un profil utilisateur existant](#).

- Lorsque vous mettez à jour votre fournisseur d'identité dans une demande d'API [UpdateIdentityProvider](#), vos modifications peuvent prendre jusqu'à une minute avant d'apparaître dans l'interface utilisateur hébergée.
- Amazon Cognito prend en charge jusqu'à 20 redirections HTTP entre lui-même et votre fournisseur d'identité.
- Lorsque votre utilisateur se connecte via l'interface utilisateur hébergée, son navigateur stocke un cookie de session de connexion chiffré qui enregistre le client et le fournisseur avec lequel il s'est connecté. S'il tente de se reconnecter avec les mêmes paramètres, l'interface utilisateur hébergée réutilise toute session existante non expirée et l'utilisateur s'authentifie sans avoir à fournir à nouveau ses informations d'identification. Si votre utilisateur se reconnecte à l'aide d'un autre fournisseur d'identité, notamment via un basculement vers ou depuis une connexion de groupe d'utilisateurs local, il doit fournir des informations d'identification et générer une nouvelle session de connexion.

Vous pouvez attribuer n'importe quel groupe d'utilisateurs IdPs à n'importe quel client d'application, et les utilisateurs ne peuvent se connecter qu'avec un IdP que vous avez attribué à leur client d'application.

Rubriques

- [Configuration des fournisseurs d'identités pour votre groupe d'utilisateurs](#)
- [Utilisation de fournisseurs d'identité sociale avec un pool d'utilisateurs](#)
- [Utilisation de fournisseurs d'identité SAML avec un groupe d'utilisateurs](#)
- [Utilisation de fournisseurs d'identité OIDC avec un pool d'utilisateurs](#)
- [Spécification des mappages d'attributs du fournisseur d'identité pour votre groupe d'utilisateurs](#)
- [Liaison d'utilisateurs fédérés à un profil utilisateur existant](#)

Configuration des fournisseurs d'identités pour votre groupe d'utilisateurs

Dans l'onglet Expérience de connexion, sous Connexion au fournisseur d'identité fédéré, vous pouvez ajouter des fournisseurs d'identité (IdPs) à votre groupe d'utilisateurs. Pour plus d'informations, consultez [Ajout de la connexion du groupe d'utilisateurs via un tiers](#).

Rubriques

- [Configuration d'une connexion utilisateur avec un fournisseur d'identité social](#)
- [Configuration d'une connexion utilisateur avec un fournisseur d'identité social OIDC](#)
- [Configuration d'une connexion utilisateur avec un fournisseur d'identité SAML](#)

Configuration d'une connexion utilisateur avec un fournisseur d'identité social

Vous pouvez utiliser une fédération pour intégrer des groupes d'utilisateurs Amazon Cognito à des fournisseurs d'identité sociaux tels que Facebook, Google et Login with Amazon.

Pour ajouter un fournisseur d'identité social, vous devez d'abord créer un compte développeur à l'aide du fournisseur d'identité. Une fois que vous disposez de votre compte développeur, inscrivez votre application avec le fournisseur d'identité. Le fournisseur d'identité crée un ID d'application et une clé secrète d'application pour votre application, tandis que vous configurez ces valeurs dans vos groupes d'utilisateurs Amazon Cognito.

- [Plateforme d'identité Google](#)
- [Facebook pour développeurs](#)
- [Login with Amazon](#)
- [Se connecter avec Apple](#)

Pour intégrer la connexion utilisateur à un fournisseur d'identité social

1. Connectez-vous à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos informations d'identification AWS.
2. Dans le volet de navigation, choisissez Groupes d'utilisateurs, puis choisissez le groupe d'utilisateurs que vous souhaitez modifier.
3. Choisissez l'onglet Sign-in experience (Expérience de connexion) et localisez Federated sign-in (Connexion fédérée).
4. Choisissez Add an identity provider (Ajouter un fournisseur d'identité) ou choisissez le fournisseur d'identité Facebook, Google, Amazon ou Apple que vous avez configuré, localisez Identity provider information (Informations sur le fournisseur d'identité) et choisissez Edit (Modifier). Pour plus d'informations sur l'ajout de fournisseurs d'identité sociaux, consultez [Utilisation de fournisseurs d'identité sociale avec un pool d'utilisateurs](#).

5. Saisissez les informations de votre fournisseur d'identité sociale en effectuant l'une des étapes suivantes, en fonction de votre choix d'IdP :

Facebook, Google et Login with Amazon

Saisissez l'ID d'application et le secret d'application que vous avez reçus lors de la création de votre application cliente.

Connexion avec Apple

Saisissez l'ID de service que vous avez fourni à Apple, ainsi que l'ID d'équipe, l'ID de clé et la clé privée que vous avez reçus lors de la création de votre client d'application.

6. Pour Authorized scopes (Périmètres autorisés), saisissez les noms des périmètres de fournisseur d'identité social à mapper aux attributs de groupe d'utilisateurs. Les portées définissent les attributs d'utilisateur comme le nom et l'adresse e-mail auxquels vous souhaitez accéder avec votre application. Lorsque vous saisissez des portées, suivez les directives suivantes en fonction de votre choix de fournisseur d'identité :

- Facebook : séparez les périmètre par des virgules. Par exemple :

```
public_profile, email
```

- Google, Login with Amazon et Sign in with Apple : séparez les périmètres par des espaces. Par exemple :

- Google :profile email openid
- Login with Amazon :profile postal_code
- Sign in with Apple :name email

Note

Pour Sign In with Apple (console), utilisez les cases à cocher pour les sélectionner.

7. Sélectionnez Enregistrer les modifications.
8. De l'onglet Intégration du client d'application, choisissez l'un des Clients d'application dans la liste, puis choisissez Modifier les paramètres d'interface utilisateur hébergée. Ajoutez le nouveau fournisseur d'identité sociale au client de l'application sous Fournisseurs d'identité.
9. Sélectionnez Enregistrer les modifications.

Pour plus d'informations sur les réseaux sociaux IdPs, consultez [Utilisation de fournisseurs d'identité sociale avec un pool d'utilisateurs](#).

Configuration d'une connexion utilisateur avec un fournisseur d'identité social OIDC

Vous pouvez intégrer une connexion utilisateur à un fournisseur d'identité (IdP) OpenID Connect (OIDC) tel que Salesforce ou Ping Identity.


Pour ajouter un fournisseur OIDC à un groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos informations d'identification AWS.
2. Choisissez Groupes d'utilisateurs dans le menu de navigation.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet Expérience de connexion. Localisez Ouverture de session fédérée et sélectionnez Ajout d'un fournisseur d'identité.
5. Choisir un fournisseur d'identité OpenID Connect
6. Saisissez un nom unique dans le champ Nom du fournisseur.
7. Saisissez l'ID client que vous avez reçu de votre fournisseur dans Client ID (ID client).
8. Saisissez le secret client que vous avez reçu de votre fournisseur dans Client secret (Secret client).
9. Saisissez les périmètres d'autorisation pour ce fournisseur. Les périmètres définissent les groupes d'attributs utilisateur (tels que name et email) que votre demande demandera à votre fournisseur. Les périmètres doivent être séparés par des espaces, suivant la spécification [OAuth 2.0](#).

L'utilisateur doit accepter de fournir ces attributs à votre application.

10. Choisissez une méthode de demande d'attribut (Attribute request method) pour fournir à Amazon Cognito la méthode HTTP (GET ou POST) à utiliser pour récupérer les détails de l'utilisateur à partir du point de terminaison userInfo exploité par votre fournisseur.
11. Choisissez une Méthode de configuration pour récupérer les points de terminaison OpenID Connect soit par Remplissage automatique via l'URL du diffuseur soit par Saisie manuel. Utilisez Auto fill through issuer URL (Remplissage automatique via l'URL de l'émetteur) lorsque votre fournisseur dispose d'un point de terminaison `.well-known/openid-configuration` public sur lequel Amazon Cognito peut récupérer les URL des points de terminaison `authorization`, `token`, `userInfo` et `jwtks_uri`.

12. Saisissez l'URL du diffuseur ou les URL des points de terminaison `authorization`, `token`, `userInfo`, et `JWKS_URI` de votre IdP.

 Note

Vous pouvez utiliser uniquement les numéros de port 443 et 80 avec des URL de découverte, remplies automatiquement et saisies manuellement. Les connexions utilisateur échouent si votre fournisseur OIDC utilise des ports TCP non standard. L'URL de l'émetteur doit commencer par `https://` et ne doit pas se terminer par un caractère `/`. Par exemple, Salesforce utilise cette URL :

```
https://login.salesforce.com
```

Le document `openid-configuration` associé à l'URL de l'émetteur doit fournir des URL HTTPS pour les valeurs suivantes : `authorization_endpoint`, `token_endpoint`, `userinfo_endpoint` et `JWKS_URI` . De même, lorsque vous choisissez `Manual input` (Entrée manuelle), vous pouvez uniquement saisir des URL HTTPS.

13. Par défaut, la demande d'OIDC `sub` est mappée à l'attribut de groupe d'utilisateurs `Nom d'utilisateur`. Vous pouvez mapper d'autres [demandes](#) d'OIDC aux attributs de groupe d'utilisateurs. Saisissez la demande OIDC et choisissez l'attribut du groupe d'utilisateurs correspondant dans la liste déroulante. Par exemple, l'e-mail de demande est souvent mappé à l'e-mail de l'attribut de groupe d'utilisateurs.
14. Mappez d'autres attributs de votre fournisseur d'identité à votre groupe d'utilisateurs. Pour de plus amples informations, veuillez consulter [Spécification des mappages d'attribut du fournisseur d'identité pour votre groupe d'utilisateurs](#).
15. Choisissez `Créer`.
16. Dans l'onglet `Intégration du client d'application`, sélectionnez l'un des Clients d'application dans la liste et `Modifier les paramètres d'interface utilisateur hébergée`. Ajoutez le nouveau fournisseur d'identité OIDC au client de l'application sous `Fournisseurs d'identité`.
17. Sélectionnez `Enregistrer les modifications`.

Pour plus d'informations sur l'OIDC IdPs, voir [Utilisation de fournisseurs d'identité OIDC avec un pool d'utilisateurs](#).

Configuration d'une connexion utilisateur avec un fournisseur d'identité SAML

Vous pouvez utiliser une fédération afin que les groupes d'utilisateurs Amazon Cognito s'intègrent avec un fournisseur d'identité (IdP) SAML. Vous fournissez un document de métadonnées, soit en téléchargeant le fichier ou en entrant l'URL de point de terminaison d'un document de métadonnées. Pour plus d'informations sur l'obtention de documents de métadonnées pour le protocole SAML tiers IdPs, consultez [Configuration de votre fournisseur d'identité SAML tiers](#).

Pour configurer un fournisseur d'identité SAML 2.0 dans votre groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos informations d'identification AWS.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet Expérience de connexion. Localisez Connexion fédérée, puis sélectionnez Ajouter un fournisseur d'identité.
5. Choisissez un fournisseur d'identité SAML.
6. Saisissez Identifiants séparés par des virgules. Un identifiant indique à Amazon Cognito qu'il doit vérifier l'adresse e-mail de connexion de l'utilisateur, puis rediriger l'utilisateur vers le fournisseur correspondant à son domaine.
7. Choisissez Ajouter un flux de déconnexion si vous souhaitez qu'Amazon Cognito envoie des demandes de déconnexion signées à votre fournisseur lorsqu'un utilisateur se déconnecte. Configurez votre fournisseur d'identité SAML 2.0 pour envoyer les réponses de déconnexion vers le point de terminaison `https://mydomain.us-east-1.amazoncognito.com/saml2/logout` créé par Amazon Cognito lorsque vous configurez l'interface utilisateur hébergée. Ce point de terminaison `saml2/logout` utilise la liaison POST.

Note

Si vous sélectionnez cette option et que votre fournisseur d'identité SAML attend une demande de déconnexion signée, vous devez également configurer le certificat de signature fourni par Amazon Cognito avec votre fournisseur d'identité SAML. L'IdP SAML traite la demande de déconnexion signée et déconnecte votre utilisateur de la session Amazon Cognito.

8. Choisissez une Source du document de métadonnées. Si votre fournisseur d'identité propose des métadonnées SAML à une URL publique, vous pouvez choisir Metadata document URL

(URL du document de métadonnées) et saisir cette URL publique. Sinon, choisissez Upload metadata document (Charger un document de métadonnées) et sélectionnez un fichier de métadonnées que vous avez téléchargé depuis votre fournisseur précédemment.

Note

Si votre fournisseur dispose d'un point de terminaison public, nous vous recommandons de saisir une URL de document de métadonnées, au lieu de charger un fichier. Si vous utilisez l'URL, Amazon Cognito actualise automatiquement les métadonnées. En règle générale, l'actualisation des métadonnées a lieu toutes les 6 heures ou avant l'expiration des métadonnées, selon la première éventualité.

9. Mappage des attributs entre votre fournisseur SAML et votre application pour mapper les attributs du fournisseur SAML au profil utilisateur de votre groupe d'utilisateurs. Incluez les attributs requis de votre groupe d'utilisateurs dans votre carte attributaire.

Par exemple, lorsque vous choisissez Groupe d'utilisateurs email, saisissez le nom de l'attribut SAML tel qu'il apparaît dans l'assertion SAML de votre fournisseur d'identité. Votre fournisseur d'identité peut proposer des exemples d'assertions SAML à titre de référence. Certains fournisseurs d'identité utilisent des noms simples, par exemple email, tandis que d'autres utilisent des noms d'attributs au format URL, tels que :

```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

10. Choisissez Créer.

Note

Si vous voyez `InvalidParameterException` pendant la création d'un fournisseur d'identité SAML avec une URL de point de terminaison de métadonnées HTTPS, veillez à ce que SSL soit correctement configuré pour le point de terminaison de métadonnées et qu'un certificat SSL valide lui soit associé. Un exemple d'une telle exception serait :
« Erreur lors de la récupération des métadonnées à partir de *<point de terminaison de métadonnées>* ».

Pour configurer le fournisseur d'identité SAML afin d'ajouter un certificat de signature

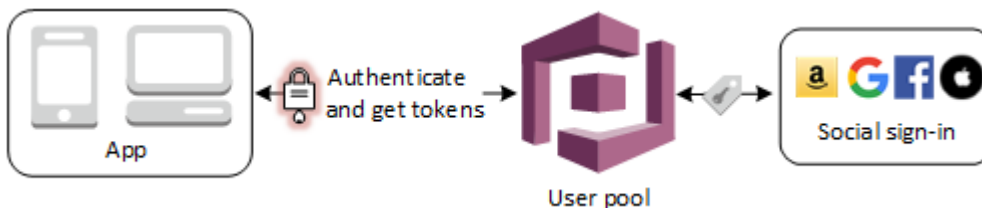
- Pour obtenir le certificat contenant la clé publique utilisée par le fournisseur d'identité pour vérifier la demande de déconnexion signée, choisissez Afficher le certificat de signature sous Fournisseurs SAML actifs dans la boîte de dialogue SAML sous Fournisseurs d'identité sur la page de console Fédération.

Pour plus d'informations sur le SAML, IdPs voir [Utilisation de fournisseurs d'identité SAML avec un groupe d'utilisateurs](#).

Utilisation de fournisseurs d'identité sociale avec un pool d'utilisateurs

Vos utilisateurs d'applications mobiles et Web peuvent se connecter via des fournisseurs d'identité sociaux (IdP) comme Facebook, Google, Amazon et Apple. Avec l'interface utilisateur web hébergée intégrée, Amazon Cognito permet de traiter et de gérer tous les utilisateurs authentifiés. Vos systèmes backend peuvent ainsi utiliser un ensemble de jetons de groupe d'utilisateurs standard. Vous devez activer l'interface utilisateur hébergée pour s'intégrer aux fournisseurs d'identité sociale pris en charge. Lorsqu'Amazon Cognito crée votre interface utilisateur hébergée, il crée des points de terminaison OAuth 2.0 qu'Amazon Cognito, votre OIDC et les réseaux sociaux utilisent pour échanger des informations. IdPs Pour plus d'informations, consultez la [Référence d'API pour l'authentification des groupes d'utilisateurs Amazon Cognito](#).

Vous pouvez ajouter un IdP social dans le ou vous pouvez utiliser la AWS Management Console AWS CLI ou l'API Amazon Cognito.



Note

La connexion via un tiers (fédération) est disponible dans les groupes d'utilisateurs Amazon Cognito. Cette fonctionnalité est indépendante de la fédération via les groupes d'identités Amazon Cognito (identités fédérées).

Rubriques

- [Prérequis](#)
- [Étape 1 : Inscription avec un fournisseur d'identité social](#)
- [Étape 2 : Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs](#)
- [Étape 3 : Test de la configuration de votre fournisseur d'identité social](#)

Prérequis

Avant de commencer, vous avez besoin de ce qui suit :

- Un groupe d'utilisateurs avec une application client et un domaine de groupe d'utilisateurs. Pour plus d'informations, consultez [Créer un groupe d'utilisateurs](#).
- Un fournisseur d'identité social.

Étape 1 : Inscription avec un fournisseur d'identité social

Avant de créer un fournisseur d'identité social avec Amazon Cognito, vous devez enregistrer votre application auprès du fournisseur d'identité social pour recevoir un ID client et une clé secrète de client.

Pour enregistrer une application avec Facebook

1. Créez un [compte développeur avec Facebook](#).
2. [Connectez-vous](#) avec vos informations d'identification Facebook.
3. Dans le menu Mes applications, choisissez Créer une nouvelle application.
4. Saisissez un nom pour votre application Facebook, puis choisissez Créer un ID d'application.
5. Dans la barre de navigation de gauche, sélectionnez Paramètres, puis Basique.
6. Notez l'ID d'app et la Clé secrète d'application. Vous les utiliserez dans la section suivante.
7. Au bas de la page, choisissez + Ajouter une plateforme.
8. Choisissez Site Web.
9. Sous Site Web, Saisissez le chemin d'accès à la page de connexion de votre application dans URL du site.

```
https://mydomain.us-east-1.amazoncognito.com/login?  
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```


10. Choisissez Save changes (Enregistrer les modifications).
11. Saisissez le chemin d'accès à la racine du domaine de votre groupe d'utilisateurs dans App Domains (Domaines d'application).

```
https://mydomain.us-east-1.amazoncognito.com
```

12. Sélectionnez Enregistrer les modifications.
13. Dans la barre de navigation, choisissez Ajouter un produit, puis Configurer depuis le produit Connexion avec Facebook.
14. Dans la barre de navigation, choisissez Connexion avec Facebook, puis Paramètres.

Saisissez le chemin d'accès au point de terminaison /oauth2/idpresponse pour votre domaine de groupe d'utilisateurs dans Valid OAuth Redirect URIs (URI de redirection OAuth valides).

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

15. Sélectionnez Enregistrer les modifications.

Pour enregistrer une application avec Amazon

1. Créez un [compte développeur avec Amazon](#).
2. [Connectez-vous](#) avec vos informations d'identification Amazon.
3. Vous devez créer un profil de sécurité Amazon pour recevoir l'ID client et de la clé secrète de client Amazon.

Choisissez Apps and Services (Applications et services) dans la barre de navigation en haut de la page, puis sélectionnez Login with Amazon.

4. Sélectionnez Create a Security Profile (Créer un profil de sécurité).
5. Saisissez un Nom du profil de sécurité, une Description du profil de sécurité et une URL de consentement à l'avis de confidentialité.
6. Choisissez Save (Enregistrer).
7. Choisissez Client ID (ID client) et Client Secret (Secret client) pour afficher l'ID et le secret client. Vous les utiliserez dans la section suivante.

8. Passez le curseur sur l'engrenage et choisissez l'icône Web Settings (Paramètres web), puis Modifier.
9. Saisissez le domaine de votre groupe d'utilisateurs dans le champ Allowed Origins (Origines autorisées).

```
https://mydomain.us-east-1.amazoncognito.com
```

10. Saisissez le domaine de votre groupe d'utilisateurs avec le point de terminaison /oauth2/idpresponse dans Allowed Return URLs (URL de retour autorisées).

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

11. Choisissez Enregistrer.

Pour enregistrer une application avec Google

Pour plus d'informations concernant OAuth 2.0 sur la plateforme Google Cloud, consultez [En savoir plus sur l'authentification et l'autorisation](#) (langue Français non garanti) dans la documentation Google Workspace for Developers.

1. Créez un [compte développeur avec Google](#).
2. Connectez-vous à la [console Google Cloud Platform](#).
3. Dans la barre de navigation supérieure, choisissez Select a project (Sélectionner un projet). Si vous avez déjà un projet sur la plateforme Google, ce menu affiche votre projet par défaut.
4. Sélectionnez NEW PROJECT (NOUVEAU PROJET).
5. Saisissez le nom de votre produit, puis choisissez CREATE (CRÉER).
6. Dans la barre de navigation de gauche, choisissez APIs and Services (API et services), puis OAuth consent screen (Écran de consentement OAuth).
7. Saisissez les informations sur l'application, avec App domain (Domaine d'application), Authorized domains (Domaines autorisés) et Developer contact information (Coordonnées du développeur). Vos Authorized domains (Domaines autorisés) doivent inclure amazoncognito.com et la racine de votre domaine personnalisé, par exemple example.com. Choisissez SAVE AND CONTINUE (ENREGISTRER ET CONTINUER).
8. 1. Sous Scopes (Étendues), choisissez Add or remove scopes (Ajouter ou supprimer des étendues), et choisissez, au minimum, les étendues OAuth suivantes.

1. .../auth/userinfo.email

2. .../auth/userinfo.profile
3. openid
9. Sous Test users (Utilisateurs test), choisissez Add users (Ajouter des utilisateurs). Saisissez votre adresse e-mail et tout autre utilisateur test autorisé, puis choisissez SAVE AND CONTINUE (ENREGISTRER ET CONTINUER).
10. Développez à nouveau la barre de navigation de gauche et choisissez APIs and Services (API et services), puis Credentials (Informations d'identification).
11. Choisissez CREATE CREDENTIALS (CRÉER DES INFORMATIONS D'IDENTIFICATION), puis OAuth client ID (ID client OAuth).
12. Choisissez un Application type (Type d'application) et donnez à votre client un Name (Nom).
13. Sous JavaScript Origines autorisées, choisissez AJOUTER UN URI. Saisissez le domaine de votre groupe d'utilisateurs.

```
https://mydomain.us-east-1.amazoncognito.com
```

14. Sous Authorized redirect URIs (URI de redirection autorisés), choisissez ADD URI (AJOUTER UN URI). Entrez le chemin d'accès au point de terminaison /oauth2/idpresponse de votre domaine de groupe d'utilisateurs.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

15. Choisissez CREATE (CRÉER).
16. Stockez en toute sécurité les valeurs que Google affiche sous Your client ID (ID de votre client) et Your client secret (Secret de votre client). Fournissez ces valeurs à Amazon Cognito lorsque vous ajoutez un fournisseur d'identité Google.

Pour enregistrer une application avec Apple

Pour en up-to-date savoir plus sur la configuration de Sign in with Apple, consultez la [section Configuration de votre environnement pour Sign in with Apple](#) dans la documentation destinée aux développeurs Apple.

1. Créez un [compte développeur Apple](#).
2. [Connectez-vous](#) avec vos informations d'identification Apple.
3. Dans la barre de navigation de gauche, choisissez Certificates, Identifiers & Profiles (Certificats, identifiants et profils).

4. Dans la barre de navigation de gauche, choisissez Identifiers (Identifiants).
5. Dans la page Identifiers (Identifiants), choisissez l'icône +.
6. Dans la page Register a New Identifier (Enregistrer un nouvel identifiant), choisissez App IDs (ID application), puis Continue (Continuer).
7. Sur la page Select a type (Sélectionner un type), choisissez App (Application), puis Continue (Continuer).
8. Dans la page Register an App ID (Enregistrer un ID d'application), procédez comme suit :
 1. Sous Description, entrez une description.
 2. Sous App ID Prefix (Préfixe d'ID d'application), saisissez un Bundle ID (ID de bundle). Notez la valeur sous Make a note of the value under (Préfixe d'ID d'application). Vous utiliserez cette valeur après avoir choisi Apple comme fournisseur d'identité dans [Étape 2 : Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs](#).
 3. Sous Capabilities (Capacités), choisissez Sign In with Apple (Connexion avec Apple), puis Edit (Modifier).
 4. Sur la page Sign in with Apple: App ID Configuration (Se connecter avec Apple : Configuration d'ID d'application), choisissez de configurer l'application comme principale ou groupée avec d'autres ID d'application, puis choisissez Save (Enregistrer).
 5. Choisissez Continue.
9. Dans la page Confirm your App ID (Confirmer votre ID d'application), choisissez Register (Inscrire).
10. Dans la page Identifiers (Identifiants), choisissez l'icône +.
11. Dans la page Register a New Identifier (Enregistrer un nouvel identifiant), choisissez Services IDs (ID de service), puis Continue (Continuer).
12. Dans la page Register an App ID (Enregistrer un ID d'application), procédez comme suit :
 1. Dans Description, saisissez une description.
 2. Sous Identifier (Identifiant), saisissez un identifiant. Notez les ID de ces services, car vous en aurez besoin après avoir choisi Apple comme fournisseur d'identité dans [Étape 2 : Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs](#).
 3. Choisissez Continue (Continuer), puis Register (Enregistrer).
13. Choisissez l'ID de services que vous venez de créer à partir de la page Identifiants.

1. Sélectionnez Sign In with Apple (Connexion avec Apple), puis choisissez Configure (Configurer).
2. Sur la page Web Authentication Configuration (Configuration de l'authentification web), sélectionnez l'ID d'application que vous avez créé précédemment comme Primary App ID (ID d'application principale).
3. Choisissez l'icône + en regard de Website URLs (URL de sites web).
4. Sous Domains and subdomains (Domaines et sous-domaines), entrez le domaine de votre groupe d'utilisateurs sans utiliser de préfixe `https://`.

```
mydomain.us-east-1.amazoncognito.com
```

5. Sous Return URLs (URL de retour), saisissez le chemin d'accès au point de terminaison `/oauth2/idpresponse` du domaine de votre groupe d'utilisateurs.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

6. Choisissez Next (Suivant), puis Done (Terminé). Vous n'avez pas besoin de vérifier le domaine.
 7. Choisissez Continue (Continuer), puis Save (Enregistrer).
14. Dans la barre de navigation de gauche, choisissez Keys (Clés).
 15. Dans la page Keys (Clés), choisissez l'icône +.
 16. Dans la page Register a New Key (Enregistrer une nouvelle clé), procédez comme suit :
 1. Sous Key Name (Nom de clé), saisissez un nom de clé.
 2. Sélectionnez Sign In with Apple (Connexion avec Apple), puis choisissez Configure (Configurer).
 3. Sur la page Configure Key (Configurer la clé), sélectionnez l'ID d'application que vous avez créé précédemment comme Primary App ID (ID d'application principale). Choisissez Enregistrer.
 4. Choisissez Continue (Continuer), puis Register (Enregistrer).
 17. Dans la page Télécharger votre clé, choisissez Télécharger pour télécharger la clé privée, puis choisissez Terminé. Vous aurez besoin de cette clé privée et de la valeur Key ID (ID de clé) affichées sur cette page après avoir choisi Apple comme fournisseur d'identité dans [Étape 2 : Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs](#).

Étape 2 : Ajout d'un fournisseur d'identité social à votre groupe d'utilisateurs

Pour configurer un IdP social de groupe d'utilisateurs avec le AWS Management Console

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet Expérience de connexion. Localisez Ouverture de session fédérée, puis sélectionnez Ajout d'un fournisseur d'identité.
5. Choisissez un fournisseur d'identité social : Facebook, Google, Login with Amazon ou Connexion avec Apple.
6. Exécutez l'une des étapes suivantes en fonction du fournisseur d'identité social que vous choisissez :
 - Google et Login with Amazon : Saisissez le l'ID du client d'application et la clé secrète du client générés dans la section précédente.
 - Facebook : Saisissez l'ID du client d'application et la clé secrète du client générés dans la section précédente, puis choisissez une version d'API (par exemple, la version 2.12). Nous recommandons de choisir la dernière version possible, car chaque API Facebook a un cycle de vie et une date d'arrêt de prise en charge. Les périmètres et attributs Facebook peuvent varier d'une version d'API à l'autre. Nous vous recommandons de tester votre connexion d'identité sociale avec Facebook pour vous assurer que la fédération fonctionne comme prévu.
 - Se connecter avec Apple : Saisissez l'ID de service, l'ID d'équipe, l'ID de clé, et la clé privée générés dans la section précédente.
7. Saisissez les noms des périmètres autorisés que vous voulez utiliser. Les périmètres définissent les attributs d'utilisateur (tels que name et email) auxquels vous souhaitez accéder avec votre application. Pour Facebook, ils doivent être séparés par des virgules. Pour Google et Login with Amazon, ils doivent être séparés par des espaces. Pour Sign in with Apple (Connexion avec Apple), activez les cases des périmètres auxquelles vous souhaitez accéder.

Fournisseur d'identité social	Exemple de règles
Facebook	public_profile, email
Google	profile email openid

Fournisseur d'identité social	Exemple de règles
Login with Amazon	profile postal_code
Se connecter avec Apple	email name

L'utilisateur de l'application est invité à accepter de fournir ces attributs à votre application. Pour plus d'informations sur les périmètres d'application des fournisseurs sociaux, consultez la documentation de Google, Facebook, Login with Amazon et Login with Apple.

Dans le cas de Sign in with Apple, les scénarios d'utilisation où les périmètres ne peuvent pas être renvoyés sont les suivants :

- Un utilisateur final rencontre une erreur après avoir quitté la page Apple de connexion (erreur interne au sein d'Amazon Cognito ou toute autre erreur écrite par le développeur).
 - L'identifiant de service est utilisé à travers les groupes d'utilisateurs et/ou autres services d'authentification.
 - Un développeur ajoute des périmètres supplémentaires après que l'utilisateur s'est connecté (aucune nouvelle information n'est extraite).
 - Un développeur supprime l'utilisateur et l'utilisateur se connecte à nouveau sans supprimer l'application de son profil Apple
8. Mappez les attributs de votre fournisseur d'identité dans votre groupe d'utilisateurs. Pour de plus amples informations, veuillez consulter [Spécification des mappages d'attribut du fournisseur d'identité pour votre groupe d'utilisateurs](#).
 9. Choisissez Créer.
 10. De l'onglet Intégration du client d'application, choisissez l'un des Clients d'application dans la liste et Modifier les paramètres d'interface utilisateur hébergée. Ajoutez le nouveau fournisseur d'identité social au client d'application sous Identity providers (Fournisseurs d'identité).
 11. Sélectionnez Enregistrer les modifications.

Étape 3 : Test de la configuration de votre fournisseur d'identité social

Vous pouvez créer une URL de connexion en utilisant les éléments des deux sections précédentes. Utilisez-les pour tester votre configuration de fournisseur d'identité social.

```
https://mydomain.us-east-1.amazonaws.com/login?
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Votre domaine se trouve sur la page de la console répertoriant le nom de domaine du groupe d'utilisateurs. L'ID client se trouve sur la page Paramètres du client d'application. Utilisez votre URL de rappel pour le paramètre `redirect_uri`. Il s'agit de l'URL de la page vers laquelle l'utilisateur est redirigé après une authentification réussie.

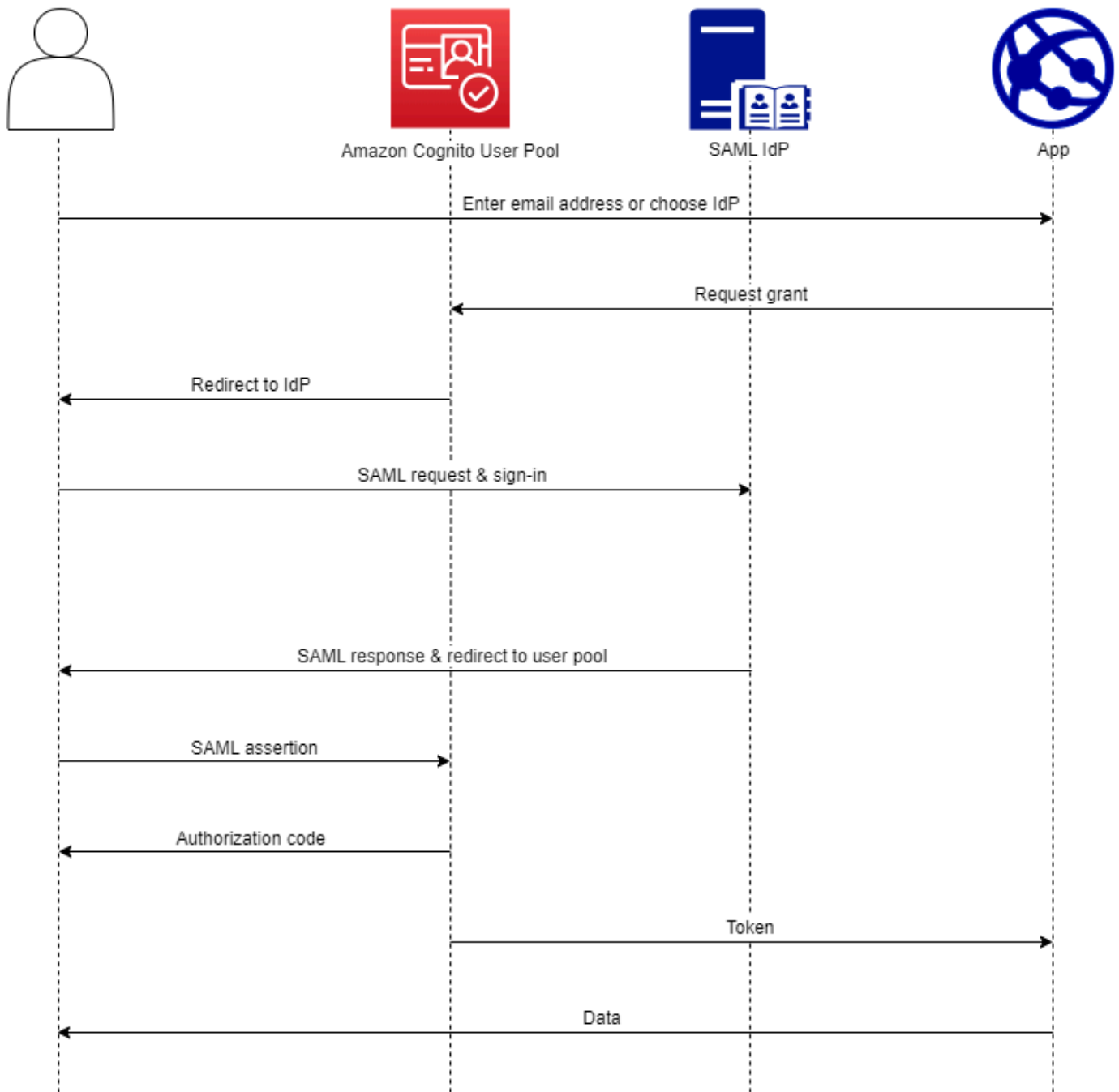
Note

Amazon Cognito annule les demandes d'authentification qui ne se terminent pas dans les 5 minutes et redirige l'utilisateur vers l'interface utilisateur hébergée. La page affiche un message d'erreur `Something went wrong`.

Utilisation de fournisseurs d'identité SAML avec un groupe d'utilisateurs

[Vous pouvez choisir de faire en sorte que les utilisateurs de vos applications Web et mobiles se connectent via un fournisseur d'identité SAML \(IdP\) tel que Microsoft Active Directory Federation Services \(ADFS\) ou Shibboleth.](#) Vous devez choisir un fournisseur d'identité SAML qui prend en charge la [norme SAML 2.0](#).

Grâce à l'interface utilisateur hébergée et aux points de terminaison de fédération, Amazon Cognito authentifie les utilisateurs IdP locaux et tiers et émet des jetons Web JSON (JWT). Avec les jetons émis par Amazon Cognito, vous pouvez consolider plusieurs sources d'identité dans une norme universelle OpenID Connect (OIDC) pour toutes vos applications. Amazon Cognito peut traiter les assertions SAML de vos fournisseurs tiers conformément à cette norme SSO. Vous pouvez créer et gérer un IdP SAML dans, via ou avec AWS Management Console l'API AWS CLI des groupes d'utilisateurs Amazon Cognito. Pour créer votre premier IdP SAML dans AWS Management Console le, voir. [Ajouter et gérer des fournisseurs d'identité SAML dans un groupe d'utilisateurs](#)



Note

La fédération avec connexion via un IdP tiers est une fonctionnalité des groupes d'utilisateurs Amazon Cognito. Les groupes d'identités Amazon Cognito, parfois appelés identités fédérées Amazon Cognito, sont une implémentation de fédération que vous devez configurer

séparément dans chaque pool d'identités. Un groupe d'utilisateurs peut être un IdP tiers d'un pool d'identités. Pour plus d'informations, consultez [Groupes d'identités Amazon Cognito](#).

Référence rapide pour la configuration de l'IdP

Vous devez configurer votre IdP SAML pour accepter les demandes et envoyer des réponses à votre groupe d'utilisateurs. La documentation de votre IdP SAML contiendra des informations sur la façon d'ajouter votre groupe d'utilisateurs en tant que partie utilisatrice ou application pour votre IdP SAML 2.0. La documentation suivante fournit les valeurs que vous devez fournir pour l'ID d'entité SP et l'URL du service client d'assertion (ACS).

Référence rapide des valeurs SAML du pool d'utilisateurs

ID de l'entité SP

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

URL DE L'ACS

```
https://Your user pool domain/saml2/idpresponse
```

Vous devez configurer votre groupe d'utilisateurs pour prendre en charge votre fournisseur d'identité. Les étapes de haut niveau pour ajouter un IdP SAML externe sont les suivantes.

1. Téléchargez les métadonnées SAML depuis votre IdP ou récupérez l'URL de votre point de terminaison de métadonnées. veuillez consulter [Configuration de votre fournisseur d'identité SAML tiers](#).
2. Ajoutez un nouvel IdP à votre groupe d'utilisateurs. Téléchargez les métadonnées SAML ou fournissez l'URL des métadonnées. veuillez consulter [Ajouter et gérer des fournisseurs d'identité SAML dans un groupe d'utilisateurs](#).
3. Assignez l'IdP aux clients de votre application. Consultez [Clients de l'application de groupe d'utilisateurs](#).

Rubriques

- [Ce qu'il faut savoir sur le protocole SAML IdPs dans les groupes d'utilisateurs Amazon Cognito](#)

- [Sensibilité à la casse des noms d'utilisateur SAML](#)
- [Ajouter et gérer des fournisseurs d'identité SAML dans un groupe d'utilisateurs](#)
- [Lancement de séance SAML dans les groupes d'utilisateurs Amazon Cognito](#)
- [Utilisation de la connexion SAML initiée par le SP](#)
- [Utilisation de la connexion SAML initiée par l'IdP](#)
- [Flux de déconnexion SAML](#)
- [Signature et chiffrement SAML](#)
- [Noms et identifiants des fournisseurs d'identité SAML](#)
- [Configuration de votre fournisseur d'identité SAML tiers](#)

Ce qu'il faut savoir sur le protocole SAML IdPs dans les groupes d'utilisateurs Amazon Cognito

Amazon Cognito traite les assertions SAML pour vous

Les groupes d'utilisateurs Amazon Cognito prennent en charge la fédération SAML 2.0 avec points de terminaison de liaison postérieure. Ainsi, votre application n'a plus besoin d'analyser ni de récupérer les réponses d'assertion SAML, étant donné que le groupe d'utilisateurs reçoit directement la réponse SAML de votre fournisseur d'identité via un agent utilisateur. Votre groupe d'utilisateurs agit en tant que fournisseur de services pour le compte de votre application. [Amazon Cognito prend en charge l'authentification unique \(SSO\) initiée par le SP et l'IDP, comme décrit dans les sections 5.1.2 et 5.1.4 de la présentation technique de SAML V2.0.](#)

Fournir un certificat de signature IdP valide

Le certificat de signature figurant dans les métadonnées de votre fournisseur SAML ne doit pas être expiré lorsque vous configurez l'IdP SAML dans votre groupe d'utilisateurs.

Les groupes d'utilisateurs prennent en charge plusieurs certificats de signature

Lorsque votre fournisseur d'identités SAML inclut plusieurs certificats de signature dans les métadonnées SAML, lors de la connexion, votre groupe d'utilisateurs détermine que l'assertion SAML est valide si elle correspond à un certificat figurant dans les métadonnées SAML. Chaque certificat de signature ne doit pas comporter plus de 4 096 caractères.

Maintenir le paramètre d'état du relais

Amazon Cognito et votre IdP SAML gèrent les informations de session à l'aide d'un paramètre `relayState`.

1. Amazon Cognito prend en charge `relayState` les valeurs supérieures à 80 octets. Alors que les spécifications SAML stipulent que la valeur `relayState` « ne doit pas dépasser 80 octets de longueur », les pratiques actuelles du secteur s'écartent souvent de ce comportement. En conséquence, le rejet des valeurs `relayState` de plus de 80 octets rend de nombreuses intégrations de fournisseurs SAML standard inutilisables.
2. Le `relayState` jeton est une référence opaque aux informations d'état gérées par Amazon Cognito. Amazon Cognito ne garantit pas le contenu du paramètre `relayState`. N'analysez pas son contenu de telle sorte que votre application dépende du résultat. Pour plus d'informations, consultez la page [Spécification SAML 2.0](#).

Identifiez le point de terminaison ACS

Votre fournisseur d'identité SAML exige que vous définissiez un point de terminaison consommateur d'assertion. Votre fournisseur d'identité redirige vos utilisateurs vers ce point de terminaison avec leur assertion SAML. Configurez le point de terminaison suivant dans le domaine de votre groupe d'utilisateurs pour la liaison POST SAML 2.0 dans votre fournisseur d'identité SAML.

```
https://Your user pool domain/saml2/idpresponse
```

With an Amazon Cognito domain:

```
https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse
```

With a custom domain:

```
https://auth.example.com/saml2/idpresponse
```

Pour plus d'informations sur les domaines de groupes d'utilisateurs, consultez [Configuration d'un domaine de groupe d'utilisateurs](#).

Aucune assertion rejouée

Vous ne pouvez pas répéter ni réutiliser une assertion SAML sur votre point de terminaison `saml2/idpresponse` Amazon Cognito. Une assertion SAML réutilisée possède un ID d'assertion qui duplique l'ID d'une demande d'API précédente.

L'ID du groupe d'utilisateurs est l'ID de l'entité SP

Vous devez fournir à votre IdP l'ID de votre groupe d'utilisateurs dans le fournisseur de services (SP)urn, également appelé URI d'audience ou ID d'entité SP. L'URI du public pour votre groupe d'utilisateurs a le format suivant.

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

Vous trouverez l'ID de votre groupe d'utilisateurs dans la section Vue d'ensemble du groupe d'utilisateurs dans la [console Amazon Cognito](#).

Cartographier tous les attributs requis

Configurez votre fournisseur d'identité SAML pour fournir des valeurs d'attributs que vous définissez pour tous les attributs requis dans votre groupe d'utilisateurs. Par exemple, `email` est un attribut courant requis pour les groupes d'utilisateurs. Avant que vos utilisateurs puissent se connecter, vos assertions de fournisseur d'identité SAML doivent inclure une demande que vous mappez à l'Attribut du groupe d'utilisateurs `email`. Pour plus d'informations sur le mappage d'attributs, consultez [Spécification des mappages d'attributs du fournisseur d'identité pour votre groupe d'utilisateurs](#).

Le format d'assertion a des exigences spécifiques

Votre IdP SAML doit inclure les revendications suivantes dans l'assertion SAML.

1. Une NameID réclamation. Amazon Cognito associe une assertion SAML à l'utilisateur de destination par NameID. En cas de NameID modification, Amazon Cognito considère que l'assertion est destinée à un nouvel utilisateur. L'attribut que vous avez défini NameID dans votre configuration IdP doit avoir une valeur persistante.

```
<saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:persistent">
  carlos
</saml2:NameID>
```

2. Un champ standard AudienceRestriction avec une valeur Audience qui définit l'ID d'entité SP de votre groupe d'utilisateurs comme cible de la réponse.

```
<saml:AudienceRestriction>
  <saml:Audience> urn:amazon:cognito:sp:us-east-1_EXAMPLE
</saml:AudienceRestriction>
```

3. Pour l'authentification unique initiée par le SP, Response élément dont InResponseTo la valeur est l'ID de demande SAML d'origine.

```
<saml2p:Response Destination="https://mydomain.us-east-1.amazoncognito.com/
saml2/idpresponse" ID="id123" InResponseTo="_dd0a3436-bc64-4679-
a0c2-cb4454f04184" IssueInstant="Date-time stamp" Version="2.0"
  xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:xs="http://
www.w3.org/2001/XMLSchema">
```

Note

Les assertions SAML initiées par l'IdP ne doivent pas contenir de valeur.
InResponseTo

4. SubjectConfirmationData Élément dont Recipient la valeur correspond au point de saml2/idpresponse terminaison de votre groupe d'utilisateurs et, pour le protocole SAML initié par SP, une InResponseTo valeur correspondant à l'ID de demande SAML d'origine.

```
<saml2:SubjectConfirmationData InResponseTo="_dd0a3436-bc64-4679-a0c2-cb4454f04184" NotOnOrAfter="Date-time stamp" Recipient="https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse"/>
```

Demandes de connexion initiées par le SP

Lorsque le [Point de terminaison d'autorisation](#) redirige votre utilisateur vers la page de connexion de votre fournisseur d'identité, Amazon Cognito inclut une demande SAML dans un paramètre d'URL de la demande HTTP GET. Une demande SAML contient des informations sur votre groupe d'utilisateurs, y compris votre point de terminaison ACS. Vous pouvez éventuellement appliquer une signature cryptographique à ces demandes.

Signer les demandes et chiffrer les réponses

Chaque groupe d'utilisateurs disposant d'un fournisseur SAML génère une paire de clés asymétriques et un certificat de signature pour une signature numérique qu'Amazon Cognito attribue aux demandes SAML. Chaque IdP SAML externe que vous configurez pour prendre en charge une réponse SAML chiffrée amène Amazon Cognito à générer une nouvelle paire de clés et un nouveau certificat de chiffrement pour ce fournisseur. Pour consulter et télécharger les certificats avec la clé publique, choisissez votre IdP dans l'onglet Expérience de connexion de la console Amazon Cognito.

Pour établir la confiance avec les demandes SAML provenant de votre groupe d'utilisateurs, fournissez à votre IdP une copie du certificat de signature SAML 2.0 de votre groupe d'utilisateurs. Votre IdP peut ignorer les demandes SAML signées par votre groupe d'utilisateurs si vous ne configurez pas l'IdP pour accepter les demandes signées.

1. Amazon Cognito applique une signature numérique aux demandes SAML que votre utilisateur transmet à votre IdP. Votre groupe d'utilisateurs signe toutes les demandes de déconnexion unique (SLO), et vous pouvez configurer votre groupe d'utilisateurs pour signer les demandes

d'authentification unique (SSO) pour n'importe quel IdP externe SAML. Lorsque vous fournissez une copie du certificat, votre IdP peut vérifier l'intégrité des demandes SAML de vos utilisateurs.

2. Votre IdP SAML peut chiffrer les réponses SAML à l'aide du certificat de chiffrement. Lorsque vous configurez un IdP avec le chiffrement SAML, votre IdP ne doit envoyer que des réponses chiffrées.

Coder des caractères non alphanumériques

Amazon Cognito n'accepte pas les caractères UTF-8 de 4 octets tels que # ou que votre IdP transmet comme valeur d'attribut. Vous pouvez coder le caractère en Base64, le transmettre sous forme de texte, puis le décoder dans votre application.

Dans l'exemple suivant, la revendication d'attribut ne sera pas acceptée :

```
<saml2:Attribute Name="Name" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xsd:string">#</saml2:AttributeValue>
</saml2:Attribute>
```

Contrairement à l'exemple précédent, la revendication d'attribut suivante sera acceptée :

```
<saml2:Attribute Name="Name" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xsd:string">8J+YkA==</saml2:AttributeValue>
</saml2:Attribute>
```

Le point de terminaison des métadonnées doit disposer d'une sécurité de couche de transport valide

Si `InvalidParameterException` s'affiche pendant la création d'un fournisseur d'identité SAML avec une URL de point de terminaison de métadonnées HTTPS, par exemple, « Error retrieving metadata from *<metadata endpoint>* » (Erreur lors de la récupération des métadonnées à partir de <point de terminaison de métadonnées>), veillez à ce que SSL soit correctement configuré pour le point de terminaison des métadonnées et qu'un certificat SSL valide y soit associé. Pour plus d'informations sur la validation des certificats, voir [Qu'est-ce qu'un certificat SSL/TLS ?](#).

Les clients d'applications utilisant le protocole SAML initié par l'IdP ne peuvent se connecter qu'avec le protocole SAML

Lorsque vous activez la prise en charge d'un IdP SAML 2.0 qui prend en charge la connexion initiée par l'IdP dans un client d'application, vous ne pouvez ajouter qu'un autre protocole SAML IdPs 2.0 à ce client d'application. Il vous est interdit d'ajouter l'annuaire des utilisateurs dans le groupe d'utilisateurs et tous les fournisseurs d'identité externes non SAML à un client d'application configuré de cette manière.

Les réponses de déconnexion doivent utiliser la liaison POST

Le `/saml2/logout` point de terminaison accepte `LogoutResponse` les HTTP POST demandes. Les groupes d'utilisateurs n'acceptent pas les réponses de déconnexion HTTP GET contraignantes.

Sensibilité à la casse des noms d'utilisateur SAML

Lorsqu'un utilisateur fédéré tente de se connecter, le fournisseur d'identité SAML (IdP) transmet un code unique à Amazon NameId Cognito dans l'assertion SAML de l'utilisateur. Amazon Cognito identifie un utilisateur fédéré SAML par sa demande NameId. Quels que soient les paramètres de distinction majuscules/minuscules de votre groupe d'utilisateurs, Amazon Cognito reconnaît un utilisateur fédéré qui revient d'un IdP SAML lorsqu'il transmet sa demande unique et différenciée par majuscules et minuscules. NameId Si vous mappez un attribut comme `email` à NameId, et que votre utilisateur change d'adresse e-mail, il ne peut pas se connecter à votre application.

Mappez NameId dans vos assertions SAML à partir d'un attribut de fournisseur d'identité dont les valeurs ne changent pas.

Par exemple, Carlos possède un profil utilisateur dans votre groupe d'utilisateurs insensible à la casse provenant d'une assertion SAML ADFS (Active Directory Federation Services) qui a transmis une valeur NameId de `Carlos@example.com`. La prochaine fois que Carlos tente de se connecter, votre fournisseur d'identité ADFS transmet une valeur NameId de `carlos@example.com`. Comme NameId doit être une correspondance exacte qui respecte la casse, la connexion ne réussit pas.

Si vos utilisateurs ne peuvent pas se connecter après le changement de NameID, supprimez leurs profils utilisateur de votre groupe d'utilisateurs. Amazon Cognito créera de nouveaux profils utilisateur la prochaine fois qu'ils se connecteront.

Rubriques

- [Ajouter et gérer des fournisseurs d'identité SAML dans un groupe d'utilisateurs](#)

- [Lancement de séance SAML dans les groupes d'utilisateurs Amazon Cognito](#)
- [Utilisation de la connexion SAML initiée par le SP](#)
- [Utilisation de la connexion SAML initiée par l'IdP](#)
- [Flux de déconnexion SAML](#)
- [Signature et chiffrement SAML](#)
- [Noms et identifiants des fournisseurs d'identité SAML](#)
- [Configuration de votre fournisseur d'identité SAML tiers](#)

Ajouter et gérer des fournisseurs d'identité SAML dans un groupe d'utilisateurs

Les procédures suivantes montrent comment créer, modifier et supprimer des fournisseurs SAML dans un groupe d'utilisateurs Amazon Cognito.

AWS Management Console


Vous pouvez utiliser le AWS Management Console pour créer et supprimer des fournisseurs d'identité SAML (IdPs).

Avant de créer un IdP SAML, vous devez disposer du document de métadonnées SAML obtenu auprès de l'IdP tiers. Pour obtenir des instructions sur l'obtention ou la génération du document de métadonnées SAML requis, consultez [Configuration de votre fournisseur d'identité SAML tiers](#).

Pour configurer un fournisseur d'identité SAML 2.0 dans votre groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos informations d'identification AWS .
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet Expérience de connexion. Localisez Ouverture de session fédérée et choisissez Ajout d'un fournisseur d'identité.
5. Choisissez un fournisseur d'identité SAML.
6. Entrez le nom du fournisseur. Vous pouvez transmettre ce nom convivial dans un paramètre de `identity_provider` requête au [Point de terminaison d'autorisation](#).
7. Saisissez Identifiants séparés par des virgules. Un identifiant indique à Amazon Cognito qu'il doit vérifier l'adresse électronique qu'un utilisateur saisit lorsqu'il se connecte, puis le redirige vers le fournisseur correspondant à son domaine.


8. Choisissez Ajouter un flux de déconnexion si vous souhaitez qu'Amazon Cognito envoie des demandes de déconnexion signées à votre fournisseur lorsqu'un utilisateur se déconnecte. Vous devez configurer votre fournisseur d'identité SAML 2.0 afin qu'il envoie les réponses de déconnexion au point de terminaison `https://mydomain.us-east-1.amazonaws.com/saml2/logout` créé lorsque vous configurez l'interface utilisateur hébergée. Ce point de terminaison `saml2/logout` utilise la liaison POST.

 Note

Si cette option est sélectionnée et que votre IdP SAML attend une demande de déconnexion signée, vous devez également fournir à votre IdP SAML le certificat de signature de votre groupe d'utilisateurs.

L'IdP SAML traite la demande de déconnexion signée et déconnecte votre utilisateur de la session Amazon Cognito.

9. Choisissez la configuration de connexion SAML initiée par l'IdP. Pour des raisons de sécurité, choisissez Accepter uniquement les assertions SAML initiées par le SP. Si vous avez préparé votre environnement pour accepter en toute sécurité les sessions de connexion SAML non sollicitées, choisissez Accepter les assertions SAML initiées par le SP et par l'IdP. Pour plus d'informations, consultez [Lancement de séance SAML dans les groupes d'utilisateurs Amazon Cognito](#).
10. Choisissez une Source du document de métadonnées. Si votre fournisseur d'identité propose des métadonnées SAML à une URL publique, vous pouvez choisir Metadata document URL (URL du document de métadonnées) et saisir cette URL publique. Sinon, choisissez Upload metadata document (Charger un document de métadonnées) et sélectionnez un fichier de métadonnées que vous avez téléchargé depuis votre fournisseur précédemment.

 Note

Nous vous recommandons de saisir l'URL d'un document de métadonnées si votre fournisseur dispose d'un point de terminaison public au lieu de télécharger un fichier. Amazon Cognito actualise automatiquement les métadonnées à partir de l'URL des métadonnées. En règle générale, l'actualisation des métadonnées a lieu toutes les 6 heures ou avant l'expiration des métadonnées, selon la première éventualité.

11. Mappez les attributs entre votre fournisseur SAML et votre groupe d'utilisateurs pour mapper les attributs du fournisseur SAML au profil utilisateur de votre groupe d'utilisateurs. Incluez les attributs requis de votre groupe d'utilisateurs dans votre carte attributaire.

Par exemple, lorsque vous choisissez l'attribut de groupe d'utilisateurs `email`, saisissez le nom de l'attribut SAML tel qu'il apparaît dans l'assertion SAML de votre fournisseur d'identité. Si votre fournisseur d'identité propose des exemples d'assertions SAML, vous pouvez les utiliser pour identifier le nom. Certains IdPs utilisent des noms simples, tels que `email`, tandis que d'autres utilisent des noms tels que les suivants.

```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

12. Choisissez Créer.

API/CLI

Utilisez les commandes suivantes pour créer et gérer un fournisseur d'identité SAML.

Pour créer un fournisseur d'identité et charger un document de métadonnées

- AWS CLI: `aws cognito-idp create-identity-provider`

```
Exemple avec fichier de métadonnées : aws cognito-idp create-identity-provider  
--user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1  
--provider-type SAML --provider-details file:///details.json --  
attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/  
identity/claims/emailaddress
```

Où `details.json` contient :

```
"ProviderDetails": {  
  "MetadataFile": "<SAML metadata XML>",  
  "IDPSignout" : "true",  
  "RequestSigningAlgorithm" : "rsa-sha256",  
  "EncryptedResponses" : "true",  
  "IDPInit" : "true"  
}
```

Note

S'il <SAML metadata XML>contient des instances du personnage", vous devez ajouter \ en tant que caractère d'échappement :\".

```
Exemple avec URL de métadonnées : aws cognito-idp create-identity-provider
--user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1
--provider-type SAML --provider-details MetadataURL=https://
myidp.example.com/sso/saml/metadata --attribute-mapping email=http://
schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

- AWS API : [CreateIdentityProvider](#)


Pour charger un nouveau document de métadonnées pour un fournisseur d'identité

- AWS CLI: `aws cognito-idp update-identity-provider`

```
Exemple avec fichier de métadonnées : aws cognito-idp update-identity-provider
--user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1
--provider-details file:///details.json --attribute-mapping
email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
emailaddress
```

Où `details.json` contient :

```
"ProviderDetails": {
  "MetadataFile": "<SAML metadata XML>",
  "IDPSignout" : "true",
  "RequestSigningAlgorithm" : "rsa-sha256",
  "EncryptedResponses" : "true",
  "IDPInit" : "true"
}
```

 Note

S'il <SAML metadata XML>contient des instances du personnage", vous devez ajouter \ en tant que caractère d'échappement :\".

```
Exemple avec URL de métadonnées : aws cognito-idp update-identity-provider
--user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1 --
provider-details MetadataURL=https://myidp.example.com/sso/saml/
metadata --attribute-mapping email=http://schemas.xmlsoap.org/
ws/2005/05/identity/claims/emailaddress
```

- AWS API : [UpdateIdentityProvider](#)

Pour obtenir des informations sur un fournisseur d'identité spécifique

- AWS CLI: `aws cognito-idp describe-identity-provider`

```
aws cognito-idp describe-identity-provider --user-pool-id us-
east-1_EXAMPLE --provider-name=SAML_provider_1
```

- AWS API : [DescribeIdentityProvider](#)

Pour répertorier les informations concernant tous IdPs

- AWS CLI: `aws cognito-idp list-identity-providers`

```
Exemple : aws cognito-idp list-identity-providers --user-pool-id us-
east-1_EXAMPLE --max-results 3
```

- AWS API : [ListIdentityProviders](#)

Pour supprimer un IdP

- AWS CLI: `aws cognito-idp delete-identity-provider`

```
aws cognito-idp delete-identity-provider --user-pool-id us-
east-1_EXAMPLE --provider-name=SAML_provider_1
```

- AWS API : [DeleteIdentityProvider](#)

Pour configurer le fournisseur d'identité SAML afin d'ajouter un groupe d'utilisateurs en tant que partie fiable

- L'URN du fournisseur de services du groupe d'utilisateurs est : `urn:amazon:cognito:sp:us-east-1_EXAMPLE`. Amazon Cognito requiert une valeur de restriction d'audience correspondant à cet URN dans la réponse SAML. Configurez votre IdP pour utiliser le point de terminaison de liaison POST suivant pour le message de réponse IdP à SP.

```
https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse
```

- Votre IdP SAML doit NameID renseigner tous les attributs requis pour votre groupe d'utilisateurs dans l'assertion SAML. NameIDest utilisé pour identifier de manière unique votre utilisateur fédéré SAML dans le groupe d'utilisateurs. Votre IdP doit transmettre l'ID de nom SAML de chaque utilisateur dans un format cohérent et en distinguant majuscules et minuscules. Toute variation de la valeur du nom ID d'un utilisateur crée un nouveau profil utilisateur.

Pour fournir un certificat de signature à votre IDP SAML 2.0

- Pour télécharger une copie de la clé publique depuis Amazon Cognito que votre IdP peut utiliser pour valider les demandes de déconnexion SAML, choisissez l'onglet Expérience de connexion de votre groupe d'utilisateurs, sélectionnez votre IdP, puis sous Afficher le certificat de signature, sélectionnez Télécharger au format .crt.

Vous pouvez supprimer n'importe quel fournisseur SAML que vous avez configuré dans votre groupe d'utilisateurs avec la console Amazon Cognito.

Pour supprimer un fournisseur SAML

1. Connectez-vous à la [console Amazon Cognito](#).
2. Dans le volet de navigation, choisissez Groupes d'utilisateurs, puis choisissez le groupe d'utilisateurs que vous souhaitez modifier.
3. Choisissez l'onglet Expérience de connexion et recherchez la connexion du fournisseur d'identité fédéré.
4. Sélectionnez le bouton radio à côté du SAML IdPs que vous souhaitez supprimer.
5. Lorsque vous êtes invité à Supprimer un fournisseur d'identité, saisissez le nom du fournisseur SAML pour confirmer la suppression, puis choisissez Delete (Supprimer).

Lancement de séance SAML dans les groupes d'utilisateurs Amazon Cognito

Amazon Cognito prend en charge l'authentification unique (SSO) initiée par le fournisseur de services (initiée par le fournisseur de services) et l'authentification unique initiée par l'IdP. En tant que meilleure pratique de sécurité, implémentez le SSO initié par le SP dans votre groupe d'utilisateurs. La section 5.1.2 de la [Présentation technique de SAML V2.0](#) décrit l'authentification unique initiée par le fournisseur de services. Amazon Cognito est le fournisseur d'identité de votre application. L'application est le fournisseur de services qui récupère les jetons pour les utilisateurs authentifiés. Toutefois, quand vous utilisez un fournisseur d'identité tiers pour authentifier les utilisateurs, Amazon Cognito est le fournisseur de services. Lorsque vos utilisateurs SAML 2.0 s'authentifient via un flux initié par le SP, ils doivent toujours d'abord envoyer une demande à Amazon Cognito et être redirigés vers l'IdP pour s'authentifier.

Pour certains cas d'utilisation d'entreprise, l'accès aux applications internes commence à un marque-page sur un tableau de bord hébergé par le fournisseur d'identité de l'entreprise. Lorsqu'un utilisateur sélectionne un marque-page, le fournisseur d'identité génère une réponse SAML et l'envoie au fournisseur de services pour authentifier l'utilisateur auprès de l'application.

Vous pouvez configurer un IdP SAML dans votre groupe d'utilisateurs pour prendre en charge le SSO initié par l'IdP. Lorsque vous prenez en charge l'authentification initiée par l'IdP, Amazon Cognito ne peut pas vérifier qu'il a sollicité la réponse SAML qu'il reçoit, car Amazon Cognito n'initie pas l'authentification par le biais d'une demande SAML. Dans le SSO initié par SP, Amazon Cognito définit des paramètres d'état qui valident une réponse SAML par rapport à la demande d'origine. Avec la connexion initiée par le SP, vous pouvez également vous prémunir contre la falsification de requêtes intersites (CSRF).

Pour un exemple de création du protocole SAML initié par le SP dans un environnement dans lequel vous ne souhaitez pas que vos utilisateurs interagissent avec l'interface utilisateur hébergée par le groupe d'utilisateurs, consultez. [Exemple de scénario : ajouter des applications Amazon Cognito à vos favoris dans un tableau de bord d'entreprise](#)

Rubriques

- [Exemple de scénario : ajouter des applications Amazon Cognito à vos favoris dans un tableau de bord d'entreprise](#)

Exemple de scénario : ajouter des applications Amazon Cognito à vos favoris dans un tableau de bord d'entreprise

Vous pouvez créer des signets dans vos tableaux de bord IDP SAML ou [OIDC](#) afin de fournir aux groupes d'utilisateurs Amazon Cognito un accès SSO aux applications Web. Vous pouvez créer un lien vers Amazon Cognito de manière à ne pas obliger les utilisateurs à se connecter à l'interface utilisateur hébergée. Pour ce faire, ajoutez un signet de connexion à votre portail qui redirige vers le groupe d'utilisateurs de votre groupe [Point de terminaison d'autorisation](#) d'utilisateurs Amazon Cognito au format suivant.

```
https://mydomain.us-east-1.amazoncognito.com/authorize?  
response_type=code&identity_provider=MySAMLIdP&client_id=1example23456789&redirect_uri=www.example.com
```

Note

Vous pouvez également utiliser un paramètre `idp_identifieur` au lieu d'un paramètre `identity_provider` dans votre demande au point de terminaison d'autorisation. Un identifiant IdP est un nom ou un domaine de messagerie alternatif que vous pouvez configurer lorsque vous créez un fournisseur d'identité dans votre groupe d'utilisateurs. Veuillez consulter [Noms et identifiants des fournisseurs d'identité SAML](#).

Lorsque vous utilisez les paramètres appropriés dans votre demande pour `/authorize`, Amazon Cognito démarre silencieusement le flux de connexion initié par le fournisseur de services et redirige votre utilisateur pour qu'il se connecte auprès de votre fournisseur d'identité.

Pour commencer, ajoutez un IdP SAML à votre groupe d'utilisateurs. Créez un client d'application qui utilise votre fournisseur d'identité SAML pour se connecter et qui possède l'URL de votre application en tant qu'URL de rappel autorisée. Pour plus d'informations sur les clients d'application, consultez [Clients de l'application de groupe d'utilisateurs](#).

Avant de déployer cet accès authentifié sur votre portail, testez la connexion initiée par le SP à votre application depuis votre interface utilisateur hébergée. Pour plus d'informations sur la configuration d'un fournisseur d'identité SAML dans Amazon Cognito, consultez [Configuration de votre fournisseur d'identité SAML tiers](#).

Le diagramme suivant montre un flux d'authentification qui émule l'authentification unique initiée par le fournisseur d'identité. Vos utilisateurs peuvent s'authentifier auprès d'Amazon Cognito à partir d'un lien sur le portail de votre entreprise.

Une fois que vous avez satisfait aux exigences, créez un signet [Point de terminaison d'autorisation](#) contenant un `identity_provider` ou un `idp_identifieur` paramètre. L'authentification de l'utilisateur se déroule comme suit.

1. Votre utilisateur se connecte au tableau de bord du fournisseur d'identité SSO. Les applications métier auxquelles l'utilisateur est autorisé à accéder remplissent ce tableau de bord.
2. Votre utilisateur choisit le lien vers l'application qui s'authentifie auprès d'Amazon Cognito. Dans de nombreux portails SSO, vous pouvez ajouter un lien d'application personnalisé. Toute fonctionnalité que vous pouvez utiliser pour créer un lien vers une URL publique dans votre portail SSO fonctionnera.
3. Votre lien d'application personnalisé dans le portail SSO dirige l'utilisateur vers le [Point de terminaison d'autorisation](#) du groupe d'utilisateurs. Ce lien inclut des paramètres pour `response_type`, `client_id`, `redirect_uri` et `identity_provider`. Le paramètre `identity_provider` est le nom que vous avez attribué au fournisseur d'identité dans votre groupe d'utilisateurs. Vous pouvez également utiliser un paramètre `idp_identifieur` à la place du paramètre `identity_provider`. Un utilisateur accède à votre point de terminaison de fédération à partir d'un lien contenant un `identity_provider` paramètre `idp_identifieur` ou. Cet utilisateur contourne la page de connexion et navigue directement pour s'authentifier auprès de votre fournisseur d'identité. Pour plus d'informations sur la dénomination SAML IdPs, consultez [Noms et identifiants des fournisseurs d'identité SAML](#).

Exemple d'URL

```
https://mydomain.us-east-1.amazoncognito.com/authorize?  
response_type=code&  
identity_provider=MySAMLIdP&  
client_id=1example23456789&  
redirect_uri=https://www.example.com
```

4. Amazon Cognito redirige la séance utilisateur vers votre fournisseur d'identité avec une demande SAML.
5. Votre utilisateur a peut-être reçu un cookie de séance de votre fournisseur d'identité lorsqu'il s'est connecté au tableau de bord. Votre fournisseur d'identité utilise ce cookie pour valider l'utilisateur en silence et le rediriger vers le point de terminaison `idpresponse` Amazon

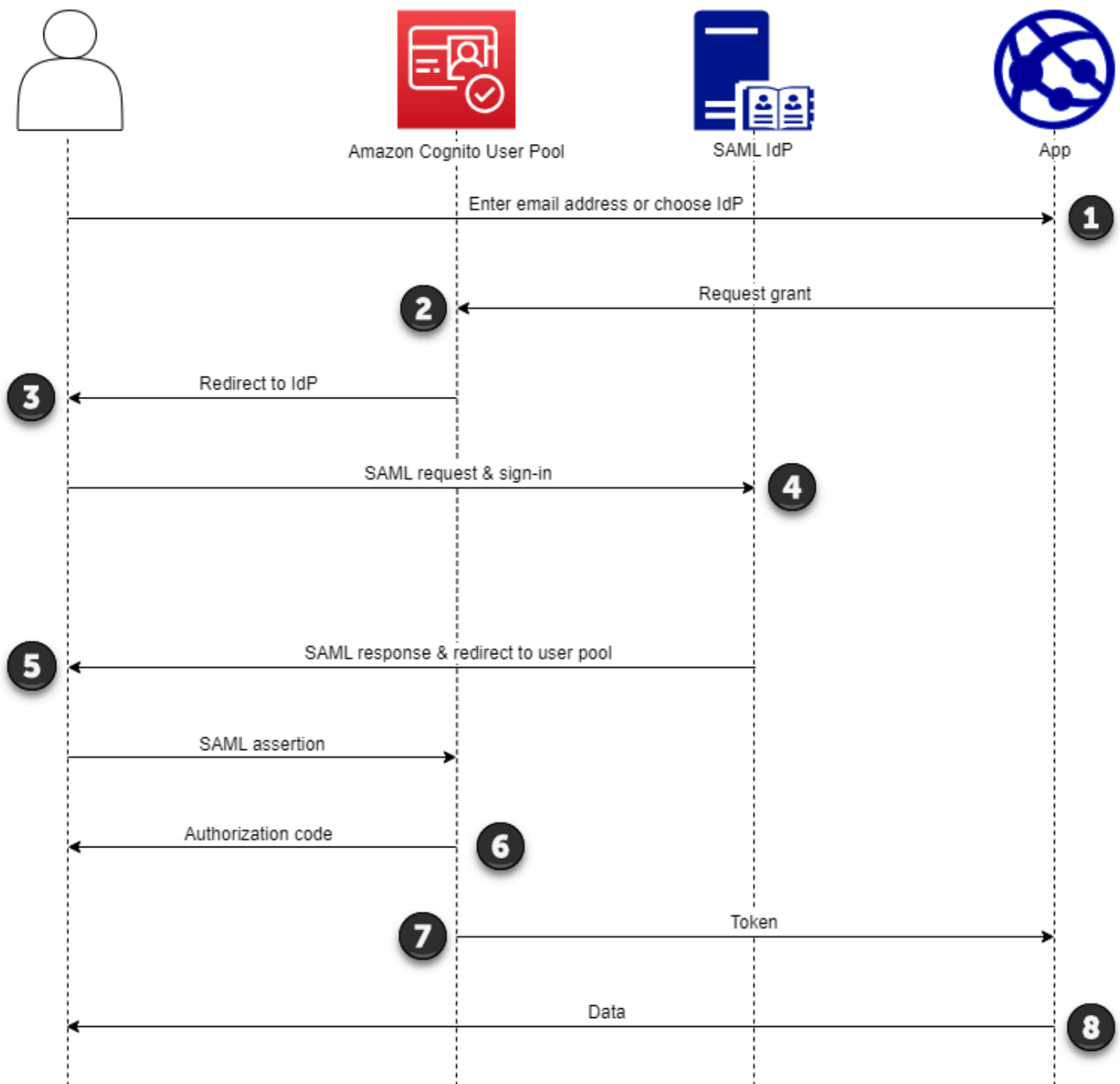
Cognito avec une réponse SAML. Si aucune session active n'existe, votre fournisseur d'identité réauthentifie l'utilisateur avant de publier la réponse SAML.

6. Amazon Cognito valide la réponse SAML et crée ou met à jour le profil utilisateur en fonction de l'assertion SAML.
7. Amazon Cognito redirige l'utilisateur vers votre application interne avec un code d'autorisation. Vous avez configuré l'URL de votre application interne en tant qu'URL de redirection autorisée pour votre client d'application.
8. Votre application échange le code d'autorisation contre des jetons Amazon Cognito. Pour plus d'informations, consultez [Point de terminaison de jeton](#).

Utilisation de la connexion SAML initiée par le SP


La meilleure pratique consiste à implémenter une connexion service-provider-initiated (initiée par le SP) à votre groupe d'utilisateurs. Amazon Cognito lance la session de votre utilisateur et le redirige vers votre IdP. Avec cette méthode, vous avez le meilleur contrôle sur les personnes qui présentent les demandes de connexion. Vous pouvez également autoriser la connexion initiée par l'IdP sous certaines conditions. Pour plus d'informations, consultez [Lancement de séance SAML dans les groupes d'utilisateurs Amazon Cognito](#).

Le processus suivant montre comment les utilisateurs se connectent à votre groupe d'utilisateurs via un fournisseur SAML.



1. Votre utilisateur saisit son adresse e-mail sur une page de connexion. Pour déterminer la redirection de votre utilisateur vers son IdP, vous pouvez collecter son adresse e-mail dans une application personnalisée ou invoquer l'interface utilisateur hébergée en mode Web. Vous pouvez configurer votre interface utilisateur hébergée pour afficher une liste d'adresses e-mail IdPs ou pour ne demander qu'une adresse e-mail.

2. Votre application appelle le point de terminaison de redirection de votre groupe d'utilisateurs et demande une session avec l'ID client correspondant à l'application et l'identifiant IdP correspondant à l'utilisateur.
3. Amazon Cognito redirige votre utilisateur vers l'IdP avec une demande SAML, [éventuellement](#) signée, dans un élément. `AuthnRequest`
4. L'IdP authentifie l'utilisateur de manière interactive ou à l'aide d'une session mémorisée dans un cookie de navigateur.
5. L'IdP redirige votre utilisateur vers le point de terminaison de réponse SAML de votre groupe d'utilisateurs avec l'assertion SAML [éventuellement cryptée dans sa](#) charge utile POST.

 Note

Amazon Cognito annule les sessions qui ne reçoivent pas de réponse dans les 5 minutes et redirige l'utilisateur vers l'interface utilisateur hébergée. Lorsque votre utilisateur rencontre ce résultat, il reçoit un message `Something went wrong` d'erreur.

6. Après avoir vérifié l'assertion SAML et [mappé les attributs utilisateur](#) à partir des demandes figurant dans la réponse, Amazon Cognito crée ou met à jour en interne le profil de l'utilisateur dans le groupe d'utilisateurs. Généralement, votre groupe d'utilisateurs renvoie un code d'autorisation à la session de navigation de l'utilisateur.
7. Votre utilisateur présente son code d'autorisation à votre application, qui échange le code contre des jetons Web JSON (JWT).
8. Votre application accepte et traite le jeton d'identification de votre utilisateur comme authentification, génère des demandes autorisées aux ressources avec leur jeton d'accès et stocke leur jeton d'actualisation.

Lorsqu'un utilisateur s'authentifie et reçoit un code d'autorisation octroyé, le groupe d'utilisateurs renvoie des jetons d'identification, d'accès et d'actualisation. Le jeton d'identification est un objet d'authentification pour la gestion des identités basée sur l'OIDC. Le jeton d'accès est un objet d'autorisation dont le champ d'[application est OAuth 2.0](#). Le jeton d'actualisation est un objet qui génère un nouvel identifiant et des jetons d'accès lorsque les jetons actuels de votre utilisateur ont expiré. Vous pouvez configurer la durée des jetons des utilisateurs dans le client d'application de votre groupe d'utilisateurs.

Vous pouvez également choisir la durée des jetons d'actualisation. Une fois le jeton d'actualisation d'un utilisateur expiré, celui-ci doit se reconnecter. S'ils se sont authentifiés via un IdP SAML, la durée

de session de vos utilisateurs est définie par l'expiration de leurs jetons, et non par l'expiration de leur session avec leur IdP. Votre application doit stocker le jeton d'actualisation de chaque utilisateur et renouveler sa session à son expiration. L'interface utilisateur hébergée conserve les sessions utilisateur dans un cookie de navigateur valide pendant 1 heure.

Utilisation de la connexion SAML initiée par l'IdP

Lorsque vous configurez votre fournisseur d'identité pour la connexion SAML 2.0 initiée par l'IDP, vous pouvez présenter des assertions SAML au point de `saml2/idpresponse` terminaison du domaine de votre groupe d'utilisateurs sans avoir à lancer la session au [Point de terminaison d'autorisation](#). Un groupe d'utilisateurs doté de cette configuration accepte les assertions SAML initiées par l'IdP provenant d'un fournisseur d'identité externe du groupe d'utilisateurs pris en charge par le client d'application demandé. Les étapes suivantes décrivent le processus global de configuration et de connexion à un fournisseur SAML 2.0 initié par un IdP.

1. Créez ou désignez un groupe d'utilisateurs et un client d'application.
2. Créez un IdP SAML 2.0 dans votre groupe d'utilisateurs.
3. Configurez votre IdP pour prendre en charge l'initiation de l'IdP. Le protocole SAML initié par l'IdP introduit des considérations de sécurité auxquelles les autres fournisseurs de SSO ne sont pas soumis. Pour cette raison, vous ne pouvez pas ajouter de contenu non SAML IdPs, y compris le groupe d'utilisateurs lui-même, à un client d'application qui utilise un fournisseur SAML avec une connexion initiée par l'IdP.
4. Associez votre fournisseur SAML initié par l'IdP à un client d'application de votre groupe d'utilisateurs.
5. Dirigez votre utilisateur vers la page de connexion de votre IdP SAML et récupérez une assertion SAML.
6. Dirigez votre utilisateur vers le point de `saml2/idpresponse` terminaison de votre groupe d'utilisateurs à l'aide de son assertion SAML.
7. Recevez des jetons Web JSON (JWT).

Pour accepter des assertions SAML non sollicitées dans votre groupe d'utilisateurs, vous devez tenir compte de leur effet sur la sécurité de votre application. L'usurpation de demande et les tentatives de CSRF sont probables lorsque vous acceptez des demandes initiées par l'IdP. Bien que votre groupe d'utilisateurs ne puisse pas vérifier une session de connexion initiée par un IdP, Amazon Cognito valide les paramètres de votre demande et vos assertions SAML.

De plus, votre assertion SAML ne doit pas contenir de InResponseTo réclamation et doit avoir été émise dans les 6 minutes précédentes.

Vous devez envoyer des demandes avec le protocole SAML initié par l'IdP à votre `/saml2/idpresponse` Pour les demandes d'autorisation d'interface utilisateur initiées par SP et hébergées, vous devez fournir des paramètres identifiant le client d'application demandé, les champs d'application, l'URI de redirection et d'autres informations sous forme de paramètres de chaîne de requête dans HTTP GET les demandes. Toutefois, pour les assertions SAML initiées par l'IdP, les détails de votre demande doivent être formatés en tant que RelayState paramètre dans le corps de la demande. HTTP POST Le corps de la demande doit également contenir votre assertion SAML en tant que SAMLResponse paramètre.

Voici un exemple de demande pour un fournisseur SAML initié par un IdP.

```
POST /saml2/idpresponse HTTP/1.1
User-Agent: USER_AGENT
Accept: */*
Host: example.auth.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded

SAMLResponse=[Base64-encoded SAML assertion]&RelayState=identity_provider
%3DMySAMLIdP%26client_id%3D1example23456789%26redirect_uri%3Dhttps%3A%2F%2Fwww.example.com%26response_type%3Dcode%26scope%3Demail%2Bopenid%2Bphone

HTTP/1.1 302 Found
Date: Wed, 06 Dec 2023 00:15:29 GMT
Content-Length: 0
x-amz-cognito-request-id: 8aba6eb5-fb54-4bc6-9368-c3878434f0fb
Location: https://www.example.com?code=[Authorization code]
```

AWS Management Console

Pour configurer un IdP pour le SAML initié par l'IdP

1. Créez un [groupe d'utilisateurs](#), un [client d'application](#) et un fournisseur d'identité SAML.
2. Dissociez tous les fournisseurs d'identité sociaux et OIDC de votre client d'application, le cas échéant.
3. Accédez à l'onglet Expérience de connexion de votre groupe d'utilisateurs.
4. Sous Connexion au fournisseur d'identité fédéré, modifiez ou ajoutez un fournisseur SAML.

5. Sous Connexion SAML initiée par l'IdP, choisissez Accepter les assertions SAML initiées par le SP et initiées par l'IdP.
6. Sélectionnez Enregistrer les modifications.

API/CLI

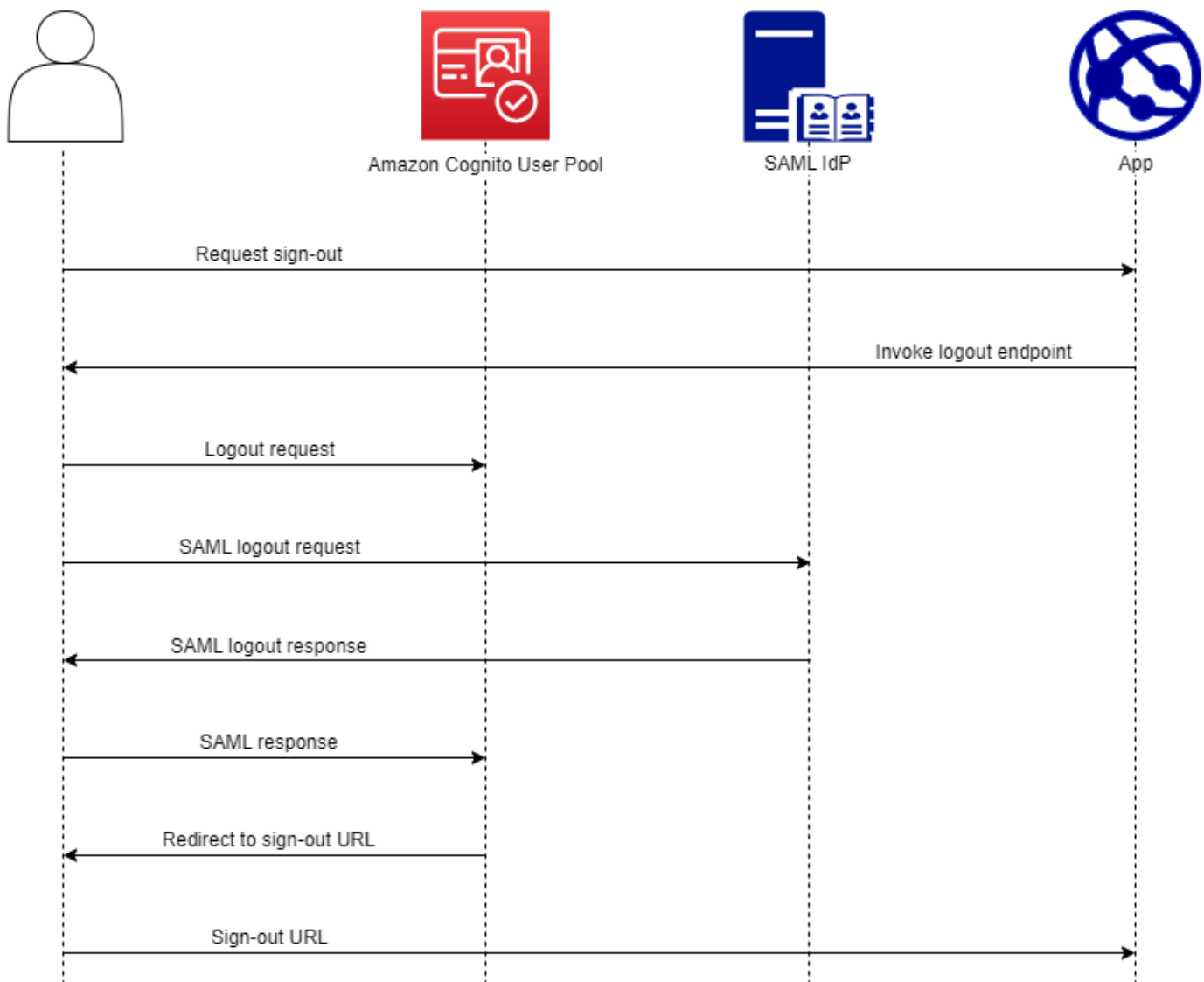
Pour configurer un IdP pour le SAML initié par l'IdP

Configurez le SAML initié par l'IdP avec le `IDPInit` paramètre dans une demande d'API [CreateIdentityProvider](#) ou [UpdateIdentityProvider](#) d'API. Voici un exemple d'IdP qui prend `ProviderDetails` en charge le SAML initié par l'IdP.

```
"ProviderDetails": {  
  "MetadataURL" : "https://myidp.example.com/saml/metadata",  
  "IDPSignout" : "true",  
  "RequestSigningAlgorithm" : "rsa-sha256",  
  "EncryptedResponses" : "true",  
  "IDPInit" : "true"  
}
```

Flux de déconnexion SAML

[Amazon Cognito prend en charge la déconnexion unique SAML 2.0](#). Lorsque vous configurez votre IdP SAML pour prendre en charge le flux de déconnexion, Amazon Cognito redirige votre utilisateur avec une demande de déconnexion SAML signée vers votre IdP. Amazon Cognito détermine l'emplacement de la redirection à partir de `SingleLogoutServiceURL` figurant dans les métadonnées de votre IdP. Amazon Cognito signe la demande de déconnexion avec le certificat de signature de votre groupe d'utilisateurs.



Lorsque vous dirigez un utilisateur disposant d'une session SAML vers le point de `/logout` terminaison de votre groupe d'utilisateurs, Amazon Cognito redirige votre utilisateur SAML avec la demande suivante vers le point de terminaison SLO spécifié dans les métadonnées de l'IdP.

```

https://[SingleLogoutService endpoint]?
SAMLRequest=[encoded SAML request]&
RelayState=[RelayState]&
SigAlg=http://www.w3.org/2001/04/xmldsig-more#rsa-sha256&
Signature=[User pool RSA signature]
  
```


Votre utilisateur retourne ensuite sur votre `saml2/logout` terminal avec un identifiant `LogoutResponse` provenant de son IdP. Votre IdP doit envoyer une `LogoutResponse` HTTP POST demande. Amazon Cognito les redirige ensuite vers la destination de redirection depuis leur demande de déconnexion initiale.

Votre fournisseur SAML peut envoyer un fichier `LogoutResponse` contenant plusieurs `AuthnStatement` fichiers. Le `sessionIndex` premier `AuthnStatement` élément d'une réponse de ce type doit correspondre `sessionIndex` à celui de la réponse SAML qui a initialement authentifié l'utilisateur. S'il se `sessionIndex` trouve dans une autre `sessionAuthnStatement`, Amazon Cognito ne reconnaîtra pas la session et votre utilisateur ne sera pas déconnecté.

AWS Management Console

Pour configurer la déconnexion SAML

1. Créez un [groupe d'utilisateurs](#), un [client d'application](#) et un IdP SAML.
2. Lorsque vous créez ou modifiez votre fournisseur d'identité SAML, sous Informations sur le fournisseur d'identité, cochez la case intitulée Ajouter un flux de déconnexion.
3. Dans l'onglet Expérience de connexion de votre groupe d'utilisateurs, sous Connexion au fournisseur d'identité fédéré, choisissez votre IdP et recherchez le certificat de signature.
4. Choisissez Télécharger au format `.crt`.
5. Configurez votre fournisseur SAML pour qu'il prenne en charge la déconnexion unique et la signature des demandes SAML, et téléchargez le certificat de signature du groupe d'utilisateurs. Votre IdP doit être redirigé vers le domaine `/saml2/logout` de votre groupe d'utilisateurs.

API/CLI

Pour configurer la déconnexion SAML

Configurez une déconnexion unique avec le `IDPSignout` paramètre d'une requête [CreateIdentityProvider](#) ou d'une demande d'[UpdateIdentityProvider](#) API. Voici un exemple `ProviderDetails` d'IdP qui prend en charge la déconnexion unique SAML.

```
"ProviderDetails": {
  "MetadataURL" : "https://myidp.example.com/saml/metadata",
  "IDPSignout" : "true",
  "RequestSigningAlgorithm" : "rsa-sha256",
```

```
"EncryptedResponses" : "true",  
"IDPInit" : "true"  
}
```

Signature et chiffrement SAML

Amazon Cognito prend en charge les demandes SAML signées et les réponses SAML cryptées pour la connexion et la déconnexion. Toutes les opérations cryptographiques effectuées au cours des opérations SAML du pool d'utilisateurs doivent générer des signatures et du texte chiffré à l'aide des clés générées par user-pool-provided Amazon Cognito. Actuellement, vous ne pouvez pas configurer un groupe d'utilisateurs pour signer des demandes ou accepter des assertions chiffrées avec une clé externe.

Note

Les certificats de votre groupe d'utilisateurs sont valides pendant 10 ans. Une fois par an, Amazon Cognito génère de nouveaux certificats de signature et de chiffrement pour votre groupe d'utilisateurs. Amazon Cognito renvoie le certificat le plus récent lorsque vous demandez le certificat de signature, et signe les demandes avec le certificat de signature le plus récent. Votre IdP peut chiffrer les assertions SAML avec n'importe quel certificat de chiffrement de groupe d'utilisateurs qui n'est pas expiré. Vos anciens certificats restent valables pendant toute leur durée de validité. Il est recommandé de mettre à jour le certificat dans la configuration de votre fournisseur chaque année.

Rubriques

- [Accepter les réponses SAML cryptées de votre IdP](#)
- [Signature de requêtes SAML](#)

Accepter les réponses SAML cryptées de votre IdP

Amazon Cognito et votre IdP peuvent garantir la confidentialité des réponses SAML lorsque les utilisateurs se connectent et se déconnectent. Amazon Cognito attribue une paire de clés RSA publique-privée et un certificat à chaque fournisseur SAML externe que vous configurez dans votre groupe d'utilisateurs. Lorsque vous activez le chiffrement des réponses pour le fournisseur SAML de votre groupe d'utilisateurs, vous devez télécharger votre certificat sur un IdP qui prend en charge les réponses SAML chiffrées. La connexion de votre groupe d'utilisateurs à votre IdP SAML ne

fonctionne pas avant que celui-ci ne commence à chiffrer toutes les assertions SAML avec la clé fournie.

Voici un aperçu du flux d'une connexion SAML cryptée.

1. Votre utilisateur commence à se connecter et choisit son IdP SAML.
2. Votre groupe d'utilisateurs [Point de terminaison d'autorisation](#) redirige votre utilisateur vers son IdP SAML avec une demande de connexion SAML. Votre groupe d'utilisateurs peut éventuellement accompagner cette demande d'une signature permettant à l'IdP de vérifier son intégrité. Lorsque vous souhaitez signer des demandes SAML, vous devez configurer votre IdP pour accepter les demandes que votre groupe d'utilisateurs a signées avec la clé publique figurant dans le certificat de signature.
3. L'IdP SAML connecte votre utilisateur et génère une réponse SAML. L'IdP chiffre la réponse avec la clé publique et redirige votre utilisateur vers le point de terminaison de votre groupe d'utilisateurs. /saml2/idpresponse L'IdP doit chiffrer la réponse conformément à la spécification SAML 2.0. Pour plus d'informations, consultez l'élément `<EncryptedAssertion>` la section [Assertions et protocoles pour le langage de balisage d'assertions de sécurité \(SAML\) OASIS V2.0](#).
4. Votre groupe d'utilisateurs déchiffre le texte chiffré dans la réponse SAML à l'aide de la clé privée et signe votre utilisateur.

Important

Lorsque vous activez le chiffrement des réponses pour un IdP SAML dans votre groupe d'utilisateurs, celui-ci doit chiffrer toutes les réponses avec une clé publique spécifique au fournisseur. Amazon Cognito n'accepte pas les réponses SAML non chiffrées provenant d'un IdP externe SAML que vous configurez pour prendre en charge le chiffrement.

Tout IdP SAML externe de votre groupe d'utilisateurs peut prendre en charge le chiffrement des réponses, et chaque IdP reçoit sa propre paire de clés.

AWS Management Console

Pour configurer le chiffrement des réponses SAML

1. Créez un [groupe d'utilisateurs](#), un [client d'application](#) et un IdP SAML.

2. Lorsque vous créez ou modifiez votre fournisseur d'identité SAML, sous Signer les demandes et chiffrer les réponses, cochez la case intitulée Exiger des assertions SAML chiffrées de la part de ce fournisseur.
3. Dans l'onglet Expérience de connexion de votre groupe d'utilisateurs, sous Connexion au fournisseur d'identité fédéré, sélectionnez votre IdP SAML et choisissez Afficher le certificat de chiffrement.
4. Choisissez Télécharger au format .crt et fournissez le fichier téléchargé à votre IdP SAML. Configurez votre IdP SAML pour chiffrer les réponses SAML à l'aide de la clé contenue dans le certificat.

API/CLI

Pour configurer le chiffrement des réponses SAML

Configurez le chiffrement des réponses avec le `EncryptedResponses` paramètre d'une requête [CreateIdentityProvider](#) ou d'une demande d'[UpdateIdentityProvider](#) API. Voici un exemple `ProviderDetails` d'IdP qui prend en charge la signature des demandes.

```
"ProviderDetails": {
  "MetadataURL" : "https://myidp.example.com/saml/metadata",
  "IDPSignout" : "true",
  "RequestSigningAlgorithm" : "rsa-sha256",
  "EncryptedResponses" : "true",
  "IDPInit" : "true"
}
```

Signature de requêtes SAML

La capacité à prouver l'intégrité des requêtes SAML 2.0 adressées à votre IdP constitue un avantage de sécurité de la connexion SAML initiée par Amazon Cognito SP. Chaque groupe d'utilisateurs doté d'un domaine reçoit un certificat de signature X.509 du groupe d'utilisateurs. Avec la clé publique contenue dans ce certificat, les groupes d'utilisateurs appliquent une signature cryptographique aux demandes de déconnexion que votre groupe d'utilisateurs génère lorsque vos utilisateurs sélectionnent un IdP SAML. Vous pouvez éventuellement configurer votre client d'application pour signer les demandes de connexion SAML. Lorsque vous signez vos demandes SAML, votre IdP peut vérifier que la signature figurant dans les métadonnées XML de vos demandes correspond à la clé publique figurant dans le certificat de groupe d'utilisateurs que vous fournissez.

AWS Management Console

Pour configurer la signature des demandes SAML

1. Créez un [groupe d'utilisateurs](#), un [client d'application](#) et un IdP SAML.
2. Lorsque vous créez ou modifiez votre fournisseur d'identité SAML, sous Signer les demandes et chiffrer les réponses, cochez la case intitulée Signer les demandes SAML à ce fournisseur.
3. Dans l'onglet Expérience de connexion de votre groupe d'utilisateurs, sous Connexion au fournisseur d'identité fédéré, choisissez Afficher le certificat de signature.
4. Choisissez Télécharger au format .crt et fournissez le fichier téléchargé à votre IdP SAML. Configurez votre IdP SAML pour vérifier la signature des demandes SAML entrantes.

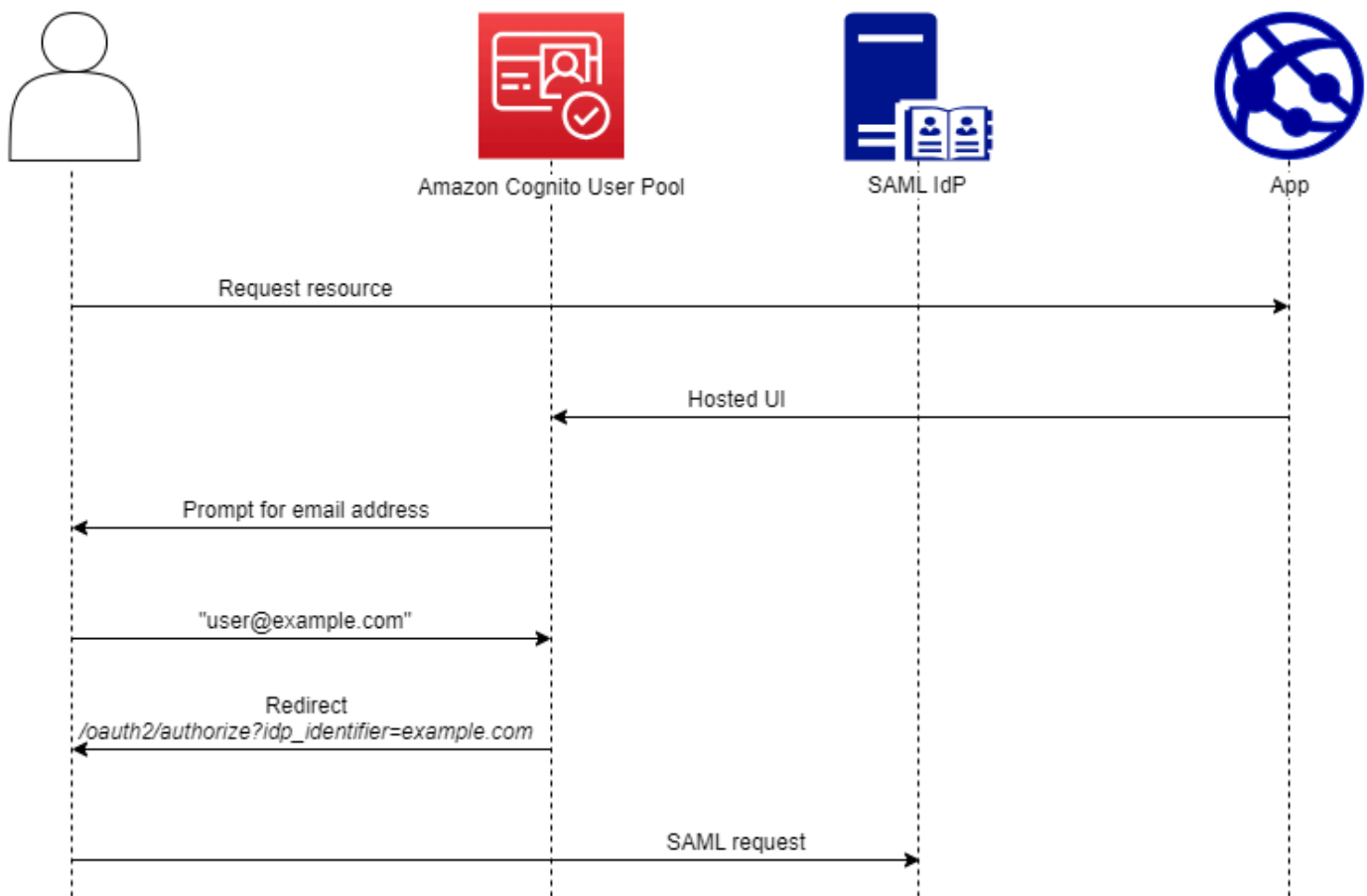
API/CLI

Pour configurer la signature des demandes SAML

Configurez la signature des demandes avec le `RequestSigningAlgorithm` paramètre d'une demande [CreateIdentityProvider](#) ou d'une demande d'[UpdateIdentityProvider](#) API. Voici un exemple `ProviderDetails` d'IdP qui prend en charge la signature des demandes.

```
"ProviderDetails": {
  "MetadataURL" : "https://myidp.example.com/saml/metadata",
  "IDPSignout" : "true",
  "RequestSigningAlgorithm" : "rsa-sha256",
  "EncryptedResponses" : "true",
  "IDPInit" : "true"
}
```

Noms et identifiants des fournisseurs d'identité SAML



Lorsque vous nommez vos fournisseurs d'identité SAML (IdPs) et que vous attribuez des identifiants IdP, vous pouvez automatiser le flux des demandes de connexion et de déconnexion initiées par le SP à ce fournisseur. Pour plus d'informations sur les contraintes de chaîne appliquées au nom du fournisseur, consultez la `ProviderName` propriété de [CreateIdentityProvider](#).

Vous pouvez également choisir jusqu'à 50 identifiants pour vos fournisseurs SAML. Un identifiant est un nom convivial pour un IdP de votre groupe d'utilisateurs. Il doit être unique au sein de ce groupe d'utilisateurs. Si vos identifiants SAML correspondent aux domaines de messagerie de vos utilisateurs, l'interface utilisateur hébergée par Amazon Cognito demande l'adresse e-mail de chaque utilisateur, évalue le domaine dans son adresse e-mail et les redirige vers l'IdP correspondant à leur domaine. Étant donné qu'une même organisation peut posséder plusieurs domaines, un même IdP peut avoir plusieurs identifiants.

Que vous utilisiez ou non des identifiants de domaine de messagerie, vous pouvez utiliser des identifiants dans une application mutualisée pour rediriger les utilisateurs vers le bon

IdP. Lorsque vous souhaitez contourner complètement l'interface utilisateur hébergée, vous pouvez personnaliser les liens que vous présentez aux utilisateurs afin qu'ils les redirigent [Point de terminaison d'autorisation](#) directement vers leur IdP. Pour connecter vos utilisateurs à l'aide d'un identifiant et les rediriger vers leur IdP, incluez l'identifiant dans le format indiqué `idp_identifiant=myidp.example.com` dans les paramètres de demande de leur demande d'autorisation initiale.

Une autre méthode pour transmettre un utilisateur à votre IdP consiste à renseigner le paramètre `identity_provider` avec le nom de votre IdP au format d'URL suivant.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
identity_provider=MySAMLIdP&
client_id=1example23456789&
redirect_uri=https://www.example.com
```

Une fois qu'un utilisateur s'est connecté avec votre IdP SAML, celui-ci le redirige avec une réponse SAML dans le corps vers votre terminal. HTTP POST `/saml2/idpresponse` Amazon Cognito traite l'assertion SAML et, si les affirmations contenues dans la réponse répondent aux attentes, redirige vers l'URL de rappel du client de votre application. Une fois que votre utilisateur s'est authentifié de cette manière, il n'a interagi qu'avec les pages Web de votre IdP et de votre application.

Avec des identifiants IdP au format de domaine, l'interface utilisateur hébergée par Amazon Cognito demande des adresses e-mail lors de la connexion, puis, lorsque le domaine de messagerie correspond à un identifiant IdP, redirige les utilisateurs vers la page de connexion de leur IdP. Par exemple, vous créez une application qui nécessite la connexion des employés de deux entreprises différentes. La première société, AnyCompany A, possède `exampleA.com` et `exampleA.co.uk`. La deuxième société, AnyCompany B, est propriétaire `exampleB.com`. Pour cet exemple, vous en avez configuré deux IdPs, un pour chaque entreprise, comme suit :

- Pour le fournisseur d'identité A, vous définissez les identifiants `exampleA.com` et `exampleA.co.uk`.
- Pour le fournisseur d'identité B, vous définissez l'identifiant `exampleB.com`.

Dans votre application, appelez l'interface utilisateur hébergée de votre client d'application pour inviter chaque utilisateur à saisir son adresse e-mail. Amazon Cognito déduit le domaine à partir de l'adresse e-mail, met en corrélation le domaine avec un IdP avec un identifiant de domaine et redirige votre utilisateur vers le bon IdP en envoyant une demande à celui qui contient un paramètre

de demande. [Point de terminaison d'autorisation](#) `idp_identifieur` Par exemple, si un utilisateur `entrebob@exampleA.co.uk`, la page suivante avec laquelle il interagit est la page de connexion à l'IdP à l'adresse `https://auth.exampleA.co.uk/sso/saml`

Vous pouvez également implémenter la même logique indépendamment. Dans votre application, vous pouvez créer un formulaire personnalisé qui collecte les informations des utilisateurs et les met en corrélation avec l'IdP approprié selon votre propre logique. Vous pouvez générer des portails d'applications personnalisés pour chacun de vos locataires d'applications, chacun étant lié au point de terminaison autorisé avec l'identifiant du locataire dans les paramètres de demande.

Pour collecter une adresse e-mail et analyser le domaine dans l'interface utilisateur hébergée, attribuez au moins un identifiant à chaque IdP SAML que vous avez attribué à votre client d'application. Par défaut, l'écran de connexion à l'interface utilisateur hébergée affiche un bouton pour chacun des éléments IdPs que vous avez attribués à votre client d'application. Toutefois, si vous avez attribué des identifiants avec succès, la page de connexion à l'interface utilisateur hébergée ressemble à l'image suivante.

L'analyse des domaines dans l'interface utilisateur hébergée nécessite que vous utilisiez des domaines comme identifiants d'IdP. Si vous attribuez un identifiant de n'importe quel type à chaque SAML IdPs d'un client d'application, l'interface utilisateur hébergée pour cette application n'affiche plus les boutons de sélection d'IDP. Ajoutez des identifiants IdP pour SAML lorsque vous avez l'intention d'utiliser l'analyse des e-mails ou une logique personnalisée pour générer des redirections. Lorsque vous souhaitez générer des redirections silencieuses et que vous souhaitez également que votre interface utilisateur hébergée affiche une liste de IdPs, n'attribuez pas d'identifiants et utilisez le paramètre de `identity_provider` demande dans vos demandes d'autorisation.

- Si vous n'attribuez qu'un seul fournisseur d'identité SAML à votre client d'application, la page de connexion de l'interface utilisateur hébergée affiche un bouton permettant de se connecter avec ce fournisseur d'identité.
- Si vous attribuez un identifiant à chaque IdP SAML que vous activez pour le client de votre application, une invite de saisie d'adresse e-mail apparaît sur la page de connexion de l'interface utilisateur hébergée.
- Si vous en avez plusieurs IdPs et que vous ne leur attribuez pas d'identifiant, la page de connexion de l'interface utilisateur hébergée affiche un bouton permettant de se connecter à chaque IdP attribué.
- Si vous avez attribué des identifiants à votre client d'application IdPs et que vous souhaitez que votre interface utilisateur hébergée affiche une sélection de boutons IdP, ajoutez un nouvel IdP

sans identifiant à votre client d'application ou créez un nouveau client d'application. Vous pouvez également supprimer un IdP existant et l'ajouter à nouveau sans identifiant. Si vous créez un nouvel IdP, vos utilisateurs SAML créeront de nouveaux profils utilisateur. Cette duplication d'utilisateurs actifs peut avoir un impact sur la facturation le mois au cours duquel vous modifiez la configuration de votre IdP.

Pour plus d'informations sur la configuration du fournisseur d'identité, consultez [Configuration des fournisseurs d'identités pour votre groupe d'utilisateurs](#).

Configuration de votre fournisseur d'identité SAML tiers

Pour configurer des solutions de fournisseur d'identité (IdP) SAML 2.0 tiers afin qu'elles fonctionnent avec la fédération pour les groupes d'utilisateurs Amazon Cognito, vous devez configurer votre IdP SAML pour qu'il soit redirigé vers l'URL Assertion Consumer Service (ACS) suivante :

<https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse> Si votre groupe d'utilisateurs dispose d'un domaine Amazon Cognito, vous pouvez trouver le chemin du domaine de votre groupe d'utilisateurs dans l'onglet App integration (Intégration d'application) de votre groupe d'utilisateurs dans la [console Amazon Cognito](#).

Certains protocoles SAML IdPs exigent que vous fournissiez leurn, également appelé URI d'audience ou ID d'entité SP, dans le formulaireurn:amazon:cognito:sp:us-east-1_EXAMPLE. Vous trouverez l'ID de votre groupe d'utilisateurs dans la section Vue d'ensemble du groupe d'utilisateurs dans la console Amazon Cognito.

Vous devez également configurer votre IdP SAML pour fournir des valeurs pour tous les attributs que vous avez désignés comme étant obligatoires dans votre groupe d'utilisateurs. Généralement, email il s'agit d'un attribut obligatoire pour les groupes d'utilisateurs, auquel cas l'IdP SAML doit fournir email une forme de réclamation dans son assertion SAML, et vous devez associer la réclamation à l'attribut de ce fournisseur.

Les informations de configuration suivantes pour les solutions IdP SAML 2.0 tierces constituent un bon point de départ pour configurer la fédération avec les groupes d'utilisateurs Amazon Cognito. Pour obtenir les informations les plus récentes, consultez directement la documentation de votre fournisseur.

Pour signer des demandes SAML, vous devez configurer votre IdP pour qu'il approuve les demandes signées par le certificat de signature de votre groupe d'utilisateurs. Pour accepter les réponses SAML chiffrées, vous devez configurer votre IdP pour chiffrer toutes les réponses SAML envoyées

à votre groupe d'utilisateurs. Votre fournisseur disposera de la documentation sur la configuration de ces fonctionnalités. Pour un exemple fourni par Microsoft, voir [Configurer le chiffrement par jeton Microsoft Entra SAML](#).

Note

Amazon Cognito a uniquement besoin du document de métadonnées de votre fournisseur d'identité. Votre fournisseur peut fournir des informations de configuration pour Compte AWS la fédération avec SAML 2.0 ; ces informations ne sont pas pertinentes pour l'intégration d'Amazon Cognito.

Solution	En savoir plus
Microsoft Active Directory Federation Services (AD FS)	Explorateur de métadonnées de fédération
Okta	Comment télécharger les métadonnées IdP et les certificats de signature SAML pour l'intégration d'une application SAML
Auth0	Configurer Auth0 en tant que fournisseur d'identité SAML
Identité Ping (PingFederate)	Exportation de métadonnées SAML depuis PingFederate
JumpCloud	Remarques sur la configuration SAML
SecureAuth	Intégration des applications SAML

Utilisation de fournisseurs d'identité OIDC avec un pool d'utilisateurs

Vous pouvez permettre à vos utilisateurs qui possèdent déjà des comptes auprès des fournisseurs d'identité [OpenID Connect \(OIDC\)](#) (IdPs) de sauter l'étape d'inscription et de se connecter à votre application à l'aide d'un compte existant. Avec l'interface utilisateur web hébergée intégrée, Amazon Cognito permet de traiter et de gérer tous les utilisateurs authentifiés. Vos systèmes backend peuvent ainsi utiliser un ensemble de jetons de groupe d'utilisateurs standard.



Note

La connexion via un tiers (fédération) est disponible dans les groupes d'utilisateurs Amazon Cognito. Cette fonctionnalité est indépendante de la fédération via les groupes d'identités Amazon Cognito (identités fédérées).

Vous pouvez ajouter un IdP OIDC à votre groupe d'utilisateurs selon AWS Management Console la méthode API du groupe d'utilisateurs, via AWS CLI ou avec celle-ci. [CreatIdentityProvider](#)

Rubriques

- [Prérequis](#)
- [Étape 1 : Inscription avec un fournisseur d'identité OIDC](#)
- [Étape 2 : Ajout d'un fournisseur d'identité OIDC à votre groupe d'utilisateurs](#)
- [Étape 3 : Test de la configuration de votre IdP OIDC](#)
- [Flux d'authentification du fournisseur d'identité de groupe d'utilisateurs OIDC](#)

Prérequis

Avant de commencer, vous avez besoin de ce qui suit :

- Un groupe d'utilisateurs avec une application client et un domaine de groupe d'utilisateurs. Pour plus d'informations, consultez [Créer un groupe d'utilisateurs](#).
- Un fournisseur d'identité (IdP) OIDC avec la configuration suivante :
 - Prend en charge l'authentification client `client_secret_post`. Amazon Cognito ne vérifie pas la revendication `token_endpoint_auth_methods_supported` au point de terminaison de découverte OIDC pour votre fournisseur d'identité. Amazon Cognito ne prend pas en charge l'authentification client `client_secret_basic`. Pour plus d'informations sur l'authentification client, consultez [Client Authentication](#) dans la documentation sur OpenID Connect.

- Utilisez uniquement HTTPS pour les points de terminaison OIDC tels que `openid_configuration`, `userInfo` et `JWKS_URI` .
- Utilisez uniquement les ports TCP 80 et 443 pour les points de terminaison OIDC.
- Ne signez les jetons d'identification qu'avec les algorithmes HMAC-SHA, ECDSA ou RSA.
- Publiez un champ standard `kid` d'identifiant de clé dans son `JWKS_URI` et incluez un champ standard `kid` dans ses jetons.

Étape 1 : Inscription avec un fournisseur d'identité OIDC

Avant de créer un fournisseur d'identité OIDC avec Amazon Cognito, vous devez enregistrer votre application auprès du fournisseur d'identité OIDC pour recevoir un ID client et une clé secrète de client.


Pour vous inscrire avec un fournisseur d'identité OIDC

1. Créez un compte développeur avec le fournisseur d'identité OIDC.

Liens vers l'OIDC IdPs

IdP OIDC	Procédure d'installation	URL de découverte OIDC
Salesforce	Installer un fournisseur d'identité Salesforce	<code>https://login.salesforce.com</code>
Ping Identity	Installer un fournisseur d'identité Ping Identity	<p><code>https://<i>adresse de votre domaine Ping</i>:9031/idp/userinfo.openid</code></p> <p>Par exemple : <code>https://pf.company.com:9031/idp/userinfo.openid</code></p>
Okta	Installer un fournisseur d'identité Okta	<p><code>https://<i>votre sous-domaine Okta</i>.oktapreview.com</code></p> <p>ou <code>https://<i>Your Okta subdomain</i>.okta.com</code></p>

IdP OIDC	Procédure d'installation	URL de découverte OIDC
Microsoft Azure Active Directory (Azure AD)	Installer un fournisseur d'identité Microsoft Azure AD	<code>https://login.microsoftonline.com/{tenant}/v2.0</code>
Google	Installer un fournisseur d'identité Google	<code>https://accounts.google.com</code>

 **Note**

Amazon Cognito propose d'utiliser Google en tant que fournisseur intégré d'identité sociale pour la connexion. Nous vous recommandons d'utiliser un IdP intégré. veuillez consulter [Utilisation de fournisseurs d'identité sociale avec un pool d'utilisateurs](#).

- Enregistrez l'URL du domaine de votre groupe d'utilisateurs avec le point de terminaison / `oauth2/idpresponse` auprès de votre fournisseur d'identité (IdP) OIDC. Cela garantit que l'IdP OIDC l'acceptera d'Amazon Cognito lors de l'authentification des utilisateurs.

`https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse`

- Enregistrez votre URL de rappel avec votre groupe d'utilisateurs Amazon Cognito. Il s'agit de l'URL de la page vers laquelle Amazon Cognito redirige l'utilisateur après une authentification réussie.

`https://www.example.com`

- Sélectionnez vos [règles](#). La règle `openid` est obligatoire. La règle `email` est requise pour accorder l'accès aux demandes `email` et `email_verified` https://openid.net/specs/openid-connect-basic-1_0.html#StandardClaims.

5. L'IdP OIDC vous fournit un ID client et un clé secrète de client. Vous les utiliserez pour configurer un IdP OIDC dans votre groupe d'utilisateurs.

Exemple : Utilisation de Salesforce en tant qu'IdP OIDC avec votre groupe d'utilisateurs

Vous utilisez un fournisseur d'identité OIDC lorsque vous souhaitez établir une approbation entre un fournisseur d'identité compatible OIDC, tel que Salesforce, et votre groupe d'utilisateurs.

1. [Créez un compte](#) sur le site web Salesforce Developers.
2. [Connectez-vous](#) avec le compte développeur que vous avez créé à l'étape précédente.
3. Dans votre page Salesforce, procédez de l'une des manières suivantes :
 - Si vous utilisez Lightning Experience, choisissez l'icône d'engrenage de configuration, puis choisissez Setup Home (Page d'accueil de configuration).
 - Si vous utilisez Salesforce et que vous voyez Setup (Configurer) dans l'en-tête de l'interface utilisateur, choisissez cette option.
 - Si vous utilisez Salesforce Classic et que vous ne voyez pas Setup (Configurer) dans l'en-tête, choisissez votre nom dans la barre de navigation supérieure, puis choisissez Setup (Configurer) dans la liste déroulante.
4. Dans la barre de navigation de gauche, sélectionnez Company Settings (Paramètres de l'entreprise).
5. Dans la barre de navigation, choisissez Domaine, saisissez un domaine, puis choisissez Créer.
6. Dans la barre de navigation de gauche, sous Outils de plateforme puis choisissez Applications.
7. Choisissez App Manager (Gestionnaire d'applications).
8.
 - a. Choisissez Nouvelle application connectée.
 - b. Renseignez les champs obligatoires.

Sous Start URL (URL de lancement), saisissez une URL au niveau du point de terminaison `/authorize` pour le domaine de groupe d'utilisateurs qui se connecte avec votre fournisseur d'identité Salesforce. Quand vos utilisateurs accèdent à votre application connectée, Salesforce les dirige vers cette URL pour terminer la connexion. Salesforce redirige ensuite les utilisateurs vers l'URL de rappel que vous avez associée à votre client d'application.

```
https://mydomain.us-east-1.amazoncognito.com/authorize?  
response_type=code&client_id=<your_client_id>&redirect_uri=https://  
www.example.com&identity_provider=CorpSalesforce
```

- c. Activez OAuth settings (Paramètres OAuth) et saisissez l'URL du point de terminaison / oauth2/idpresponse pour votre domaine de groupe d'utilisateurs dans Callback URL (URL de rappel). Il s'agit de l'URL de la page où Salesforce émet le code d'autorisation qu'Amazon Cognito échange contre un jeton OAuth.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

9. Sélectionnez vos [règles](#). Vous devez inclure l'étendue openid. Pour accorder l'accès aux [demandes](#) email et email_verified, ajoutez l'étendue email. Séparez les étendues par des espaces.
10. Choisissez Créer.

Dans Salesforce, l'ID client est appelé Consumer Key (Clé consommateur) et la clé secrète du client est appelée Consumer Secret (Clé secrète du consommateur). Notez l'ID client et la clé secrète du client. Vous les utiliserez dans la section suivante.

Étape 2 : Ajout d'un fournisseur d'identité OIDC à votre groupe d'utilisateurs

Dans cette section, vous configurez votre groupe d'utilisateurs pour traiter les demandes d'authentification basées sur OIDC à partir d'un IdP OIDC.

Pour ajouter un IdP OIDC (console Amazon Cognito)

Ajouter un fournisseur d'identité OIDC

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs dans le menu de navigation.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet Expérience de connexion. Localisez Ouverture de session fédérée et sélectionnez Ajout d'un fournisseur d'identité.
5. Choisissez un fournisseur d'identité OpenID Connect.
6. Saisissez un nom unique dans le champ Nom du fournisseur.

7. Saisissez l'ID client que vous avez reçu de votre fournisseur dans ID du client.
8. Saisissez le secret client que vous avez reçu de votre fournisseur dans Secret du client.
9. Saisissez les périmètres d'autorisation pour ce fournisseur. Les périmètres définissent les groupes d'attributs utilisateur (tels que name et email) que votre demande demandera à votre fournisseur. Les périmètres doivent être séparés par des espaces, suivant la spécification [OAuth 2.0](#).

Votre utilisateur est invité à accepter de fournir ces attributs à votre application.

10. Choisissez une méthode de demande d'attribut pour fournir à Amazon Cognito la méthode HTTP (GET ou POST) qu'il doit utiliser pour récupérer les détails de l'utilisateur à partir du point de terminaison userInfo exploité par votre fournisseur.
11. Choisissez une Méthode de configuration pour récupérer les points de terminaison OpenID Connect soit par Remplissage automatique via l'URL du diffuseur soit par Saisie manuel. Utilisez Auto fill through issuer URL (Remplissage automatique via l'URL de l'émetteur) lorsque votre fournisseur dispose d'un point de terminaison .well-known/openid-configuration public sur lequel Amazon Cognito peut récupérer les URL des points de terminaison authorization, token, userInfo et jwks_uri.
12. Saisissez l'URL du diffuseur ou les URL des points de terminaison authorization, token, userInfo, et jwks_uri de votre IdP.

Note

L'URL doit commencer par `https://` et ne doit pas se terminer par une barre oblique `/`. Seuls les numéros de port 443 et 80 peuvent être utilisés avec cette URL. Par exemple, Salesforce utilise cette URL :

```
https://login.salesforce.com
```

Si vous choisissez le remplissage automatique, le document de découverte doit utiliser HTTPS pour les valeurs suivantes : `authorization_endpoint`, `token_endpoint`, `userinfo_endpoint`, et `jwks_uri`. Sinon, la connexion échoue.

13. Par défaut, la demande d'OIDC sub est mappée à l'attribut de groupe d'utilisateurs Nom d'utilisateur. Vous pouvez mapper d'autres [demandes](#) d'OIDC aux attributs de groupe d'utilisateurs. Saisissez la demande OIDC et choisissez l'attribut du groupe d'utilisateurs correspondant dans la liste déroulante. Par exemple, l'e-mail de demande est souvent mappé à l'e-mail de l'attribut de groupe d'utilisateurs.

14. Mappez les attributs de votre fournisseur d'identité dans votre groupe d'utilisateurs. Pour de plus amples informations, veuillez consulter [Spécification des mappages d'attribut du fournisseur d'identité pour votre groupe d'utilisateurs](#).
15. Choisissez Créer.
16. De l'onglet Intégration du client d'application, choisissez l'un des Clients d'application dans la liste et Modifier les paramètres d'interface utilisateur hébergée. Ajoutez le nouveau fournisseur d'identité OIDC au client d'application sous Identity providers (Fournisseurs d'identité).
17. Sélectionnez Enregistrer les modifications.

Pour ajouter un IdP OIDC (AWS CLI)

- Consultez les descriptions des paramètres de la méthode [CreateIdentityProviderAPI](#).

```
aws cognito-idp create-identity-provider
--user-pool-id string
--provider-name string
--provider-type OIDC
--provider-details map

--attribute-mapping string
--idp-identifiers (list)
--cli-input-json string
--generate-cli-skeleton string
```

Utilisez ce schéma pour les détails du fournisseur :

```
{
  "client_id": "string",
  "client_secret": "string",
  "authorize_scopes": "string",
  "attributes_request_method": "string",
  "oidc_issuer": "string",

  "authorize_url": "string",
```

```
"token_url": "string",  
"attributes_url": "string",  
"jwks_uri": "string"  
}
```

Étape 3 : Test de la configuration de votre IdP OIDC

Vous pouvez créer l'URL d'autorisation en utilisant les éléments des deux sections précédentes, et les utiliser pour tester votre configuration d'IdP OIDC.

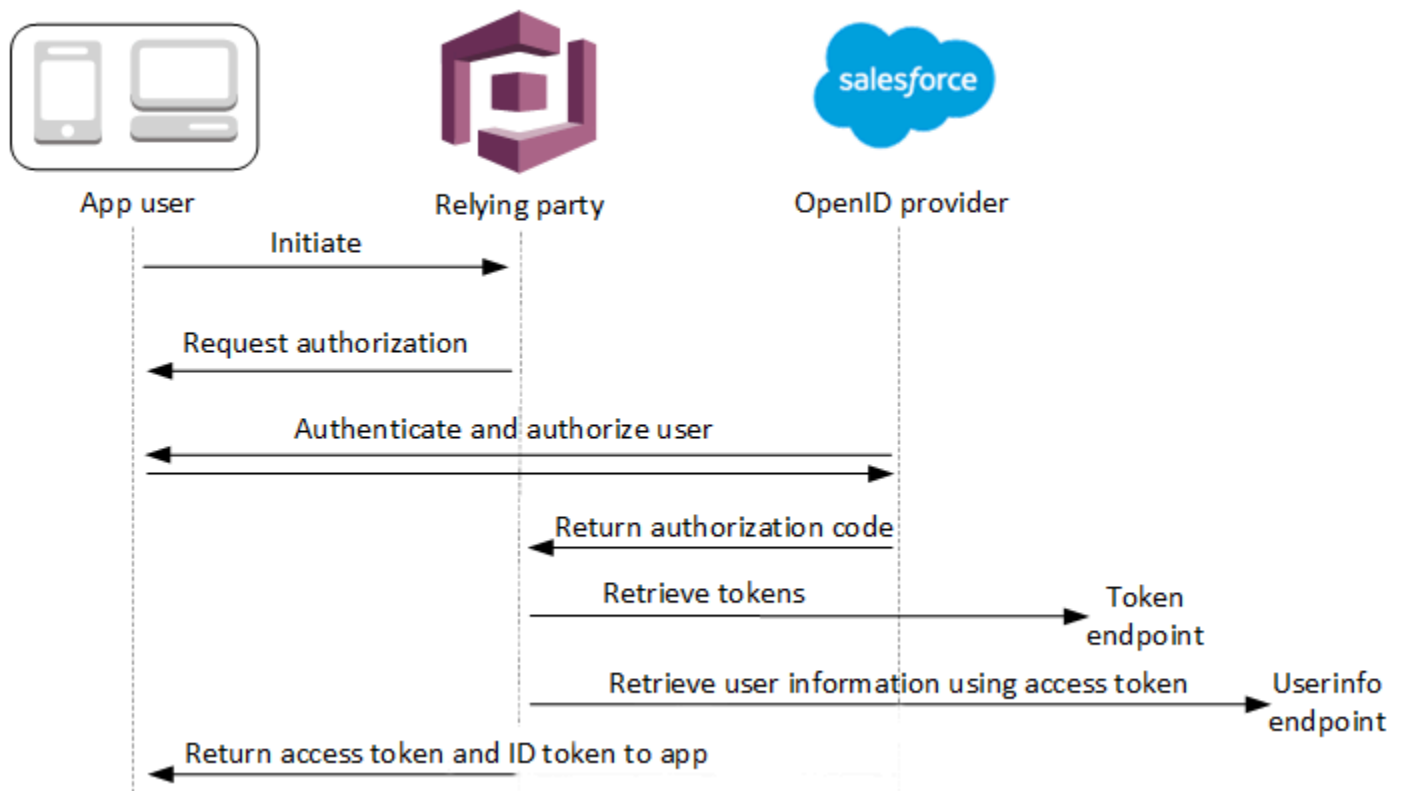
```
https://mydomain.us-east-1.amazoncognito.com/oauth2/authorize?  
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Votre domaine se trouve sur la page de la console répertoriant le nom de domaine du groupe d'utilisateurs. Le `client_id` se trouve sur la page Paramètres généraux. Utilisez votre URL de rappel pour le paramètre `redirect_uri`. Il s'agit de l'URL de la page vers laquelle l'utilisateur est redirigé après une authentification réussie.

Flux d'authentification du fournisseur d'identité de groupe d'utilisateurs OIDC

Lorsque l'utilisateur se connecte à votre application à l'aide d'un IdP OIDC, il passe par le flux d'authentification suivant.

1. L'utilisateur est dirigé vers la page de connexion intégrée Amazon Cognito, et invité à se connecter via un IdP OIDC tel que Salesforce.
2. L'utilisateur est redirigé vers le point de terminaison `authorization` du fournisseur d'identité OIDC.
3. Une fois que l'utilisateur est authentifié, l'IdP OIDC le redirige vers Amazon Cognito avec un code d'autorisation.
4. Amazon Cognito échange le code d'autorisation avec l'IdP OIDC pour obtenir un jeton d'accès.
5. Amazon Cognito crée ou met à jour le compte utilisateur dans votre groupe d'utilisateurs.
6. Amazon Cognito émet les jetons de porteur à votre application, lesquels peuvent inclure des jetons d'identité, d'accès et d'actualisation.



Note

Amazon Cognito annule les demandes d'authentification qui ne se terminent pas dans les 5 minutes et redirige l'utilisateur vers l'interface utilisateur hébergée. La page affiche un message d'erreur `Something went wrong`.

L'OIDC est une couche d'identité qui s'ajoute à OAuth 2.0, qui spécifie les jetons d'identité au format JSON (JWT) émis par les applications clientes OIDC (parties IdPs utilisatrices). Consultez la documentation de votre IdP OIDC pour plus d'informations sur l'ajout d'Amazon Cognito en tant que partie utilisatrice d'OIDC.

Lorsqu'un utilisateur s'authentifie avec un octroi de code d'autorisation, le groupe d'utilisateurs retourne les jetons d'identification, d'accès et d'actualisation. Le jeton d'identification est un jeton [OIDC](#) standard pour la gestion des identités et le jeton d'accès est un jeton [OAuth 2.0](#) standard. Pour plus d'informations sur les types d'octroi que le client de votre application de groupe d'utilisateurs peut prendre en charge, consultez [Point de terminaison d'autorisation](#).

Comment un groupe d'utilisateurs traite les demandes d'un fournisseur OIDC

Lorsque votre utilisateur effectue une connexion avec un fournisseur OIDC tiers, l'interface utilisateur hébergée Amazon Cognito récupère un code d'autorisation auprès de l'IdP. Votre groupe d'utilisateurs échange le code d'autorisation contre des jetons d'accès et d'identification avec le point de terminaison `token` de votre IdP. Votre groupe d'utilisateurs ne transmet pas ces jetons à votre utilisateur ou application, mais s'en sert pour créer un profil utilisateur avec les données qu'il présente dans les demandes de ses propres jetons.

Amazon Cognito ne valide pas le jeton d'accès de manière indépendante. En effet, il demande des informations d'attribut utilisateur au point de terminaison `userInfo` du fournisseur et s'attend à ce que la demande soit refusée si le jeton n'est pas valide.

Amazon Cognito valide le jeton d'identification du fournisseur en effectuant les vérifications suivantes :

1. Vérifie que le fournisseur a signé le jeton avec un algorithme issu de l'ensemble suivant : RSA, HMAC, Elliptic Curve.
2. Si le fournisseur a signé le jeton avec un algorithme de signature asymétrique, vérifie que l'ID de la clé de signature figurant dans la demande `kid` du jeton est répertorié sur le point de terminaison `JWKS_URI` du fournisseur.
3. Compare la signature du jeton d'identification à la signature qu'il attend en fonction des métadonnées du fournisseur.
4. Compare la demande `iss` à l'émetteur OIDC configuré pour l'IdP.
5. Vérifie que la demande `aud` correspond à l'ID client configuré sur l'IdP ou qu'elle contient l'ID client configuré si la demande `aud` contient plusieurs valeurs.
6. Vérifie que l'horodatage figurant dans la demande `exp` n'est pas antérieur à l'heure actuelle.

Votre groupe d'utilisateurs valide le jeton d'identification, puis tente d'envoyer une demande au point de terminaison `userInfo` du fournisseur avec le jeton d'accès du fournisseur. Il récupère les informations de profil utilisateur que les portées du jeton d'accès l'autorisent à lire. Votre groupe d'utilisateurs recherche ensuite les attributs utilisateur obligatoires que vous avez définis dans votre groupe d'utilisateurs. Vous devez créer des mappages d'attributs dans la configuration de votre fournisseur pour les attributs obligatoires. Votre groupe d'utilisateurs vérifie le jeton d'identification du fournisseur et la réponse `userInfo`. Votre groupe d'utilisateurs écrit toutes les demandes qui correspondent aux règles de mappage dans les attributs utilisateur du profil utilisateur du groupe d'utilisateurs. Votre groupe d'utilisateurs ignore les attributs qui correspondent à une règle

de mappage, mais qui ne sont pas obligatoires et qui ne se trouvent pas dans les demandes du fournisseur.

Spécification des mappages d'attributs du fournisseur d'identité pour votre groupe d'utilisateurs

Vous pouvez utiliser l' AWS Management Console API ou ou pour spécifier les AWS CLI mappages d'attributs pour le fournisseur d'identité (IdP) de votre groupe d'utilisateurs.

Choses à savoir sur les mappages

Avant de commencer à configurer le mappage des attributs utilisateur, passez en revue les informations importantes suivantes.

- Quand un utilisateur fédéré se connecte à votre application, un mappage doit être présent pour chaque attribut de groupe d'utilisateurs requis par votre groupe d'utilisateurs. Par exemple, si votre groupe d'utilisateurs nécessite un attribut `email` pour l'inscription, mappez cet attribut à son équivalent à partir du fournisseur d'identité.
- Par défaut, les adresses e-mail mappées ne sont pas vérifiées. Vous ne pouvez pas vérifier une adresse e-mail mappée à l'aide d'un code à usage unique. Au lieu de cela, mappez un attribut de votre fournisseur d'identité pour obtenir l'état de vérification. Par exemple, Google et la plupart des fournisseurs OIDC incluent l'attribut `email_verified`.
- Vous pouvez mapper les jetons du fournisseur d'identité (IdP) à des attributs personnalisés de votre groupe d'utilisateurs. Les fournisseurs sociaux présentent un jeton d'accès, et les fournisseurs OIDC présentent un jeton d'accès et d'identification. Pour mapper un jeton, ajoutez un attribut personnalisé d'une longueur maximale de 2 048 caractères, accordez à votre client d'application un accès en écriture à l'attribut, puis mappez `access_token` ou `id_token` de l'IdP à l'attribut personnalisé.
- Pour chaque attribut du groupe d'utilisateurs mappé, la longueur de valeur maximale de 2 048 caractères doit être suffisamment grande pour la valeur qu'Amazon Cognito obtient du fournisseur d'identité. Sinon, Amazon Cognito signale une erreur lorsque les utilisateurs se connectent à votre application. Amazon Cognito ne prend pas en charge le mappage des jetons IdP à des attributs personnalisés lorsque la longueur des jetons est supérieure à 2 048 caractères.
- Amazon Cognito calcule l'`username` attribut du profil d'un utilisateur fédéré à partir de demandes spécifiques transmises par votre IdP fédéré, comme indiqué dans le tableau suivant. Amazon Cognito ajoute à cette valeur d'attribut le nom de votre IdP, par exemple. `MyOIDCIdP_[sub]` Si vous souhaitez que vos utilisateurs fédérés aient un attribut correspondant exactement à un attribut

de votre répertoire d'utilisateurs externe, associez cet attribut à un attribut de connexion Amazon Cognito tel que `preferred_username`

Fournisseur d'identité	Attribut source <code>username</code>
Facebook	<code>id</code>
Google	<code>sub</code>
Login with Amazon	<code>user_id</code>
Se connecter avec Apple	<code>sub</code>
Fournisseurs SAML	<code>NameID</code>
Fournisseurs OpenID Connect (OIDC)	<code>sub</code>

- Amazon Cognito doit être en mesure de mettre à jour vos attributs mappés de groupe d'utilisateurs lorsque les utilisateurs se connectent à votre application. Quand un utilisateur se connecte via un fournisseur d'identité, Amazon Cognito met à jour les attributs mappés avec les informations les plus récentes du fournisseur d'identité. Amazon Cognito met à jour chaque attribut mappé, même si sa valeur actuelle correspond déjà aux dernières informations. Pour vous assurer qu'Amazon Cognito peut mettre à jour les attributs, vérifiez les exigences suivantes :
 - Tous les attributs personnalisés du groupe d'utilisateurs que vous mappez à partir de votre fournisseur d'identité doivent être mutables. Vous pouvez mettre à jour à tout moment des attributs personnalisés mutables. En revanche, pour un attribut personnalisé immuable, vous pouvez définir une valeur uniquement la première fois que vous créez le profil utilisateur. Pour créer un attribut personnalisé mutable dans la console Amazon Cognito, cochez la case `Mutable` pour l'attribut que vous ajoutez quand vous sélectionnez `Add custom attributes` (Ajouter des attributs personnalisés) dans l'onglet `Sign-up experience` (Expérience d'inscription). Ou, si vous créez votre groupe d'utilisateurs à l'aide de l'opération [CreateUserPoolAPI](#), vous pouvez définir le `Mutable` paramètre de chacun de ces attributs sur `true`. Si votre IdP envoie une valeur pour un attribut immuable mappé, Amazon Cognito renvoie une erreur et la connexion échoue.
 - Dans les paramètres du client d'application pour votre application, les attributs mappés doivent être accessibles en écriture. Vous pouvez définir des attributs accessibles en écriture dans la page `Clients d'application` de la console Amazon Cognito. Si vous créez le client d'application au moyen de l'opération d'API [CreateUserPoolClient](#), vous pouvez également ajouter ces attributs à la grappe `WriteAttributes`. Si votre IdP envoie une valeur pour un attribut

mappé non inscriptible, Amazon Cognito ne définit pas la valeur de l'attribut et procède à l'authentification.

- Lorsque les attributs IdP contiennent plusieurs valeurs, Amazon Cognito ajuste toutes les valeurs en une seule chaîne séparée par des virgules et code sous forme d'URL les valeurs contenant des caractères non alphanumériques (à l'exception des caractères « », « », « » et . « »). - * _ Vous devez décoder et analyser ces valeurs individuelles avant de les utiliser dans votre application.

Spécification des mappages d'attributs du fournisseur d'identité pour votre groupe d'utilisateurs (AWS Management Console)

Vous pouvez utiliser le AWS Management Console pour spécifier des mappages d'attributs pour l'IdP de votre groupe d'utilisateurs.

Note

Amazon Cognito mappera les revendications entrantes aux attributs de groupe d'utilisateurs uniquement si ces revendications existent dans le jeton entrant. Si une demande précédemment mappée n'existe plus dans le jeton entrant, elle ne sera ni supprimée ni modifiée. Si votre application nécessite le mappage de champs standard supprimés, vous pouvez utiliser le déclencheur Lambda de pré-authentification pour supprimer l'attribut personnalisé pendant l'authentification et permettre à ces attributs de se remplir à nouveau à partir du jeton entrant.

Pour spécifier un mappage d'attributs de fournisseur d'identité social

1. Connectez-vous à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Dans le volet de navigation, choisissez Groupes d'utilisateurs, puis choisissez le groupe d'utilisateurs que vous souhaitez modifier.
3. Choisissez l'onglet Sign-in experience (Expérience de connexion) et localisez Federated sign-in (Connexion fédérée).
4. Choisissez Add an identity provider (Ajouter un fournisseur d'identité) ou choisissez le fournisseur d'identité Facebook, Google, Amazon ou Apple que vous avez configuré. Localisez Mappage d'attribut et choisissez Modifier.

Pour plus d'informations sur l'ajout d'un fournisseur d'identité social, consultez [Utilisation de fournisseurs d'identité sociale avec un pool d'utilisateurs](#).

5. Pour chaque attribut que vous devez mapper, effectuez les étapes suivantes :
 - a. Sélectionnez un attribut dans la colonne Attribut du groupe d'utilisateurs Il s'agit de l'attribut affecté au profil utilisateur dans votre groupe d'utilisateurs. Les attributs personnalisés sont répertoriés après les attributs standard.
 - b. Sélectionnez un attribut dans la colonne **attribut du <fournisseur>**. Il s'agit de l'attribut transmis à partir du répertoire du fournisseur. Les attributs connus du fournisseur social sont fournis dans une liste déroulante.
 - c. Pour mapper des attributs supplémentaires entre votre IdP et Amazon Cognito, choisissez Ajout d'un autre attribut.
6. Sélectionnez Enregistrer les modifications.

Pour spécifier un mappage d'attribut de fournisseur SAML

1. Connectez-vous à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Dans le volet de navigation, choisissez Groupes d'utilisateurs, puis choisissez le groupe d'utilisateurs que vous souhaitez modifier.
3. Choisissez l'onglet Sign-in experience (Expérience de connexion) et localisez Federated sign-in (Connexion fédérée).
4. Choisissez Add an identity provider (Ajouter un fournisseur d'identité) ou choisissez le fournisseur d'identité SAML que vous avez configuré. Localisez Mappage d'attribut, et choisissez Modifier. Pour plus d'informations sur l'ajout d'un fournisseur d'identité SAML, consultez [Utilisation de fournisseurs d'identité SAML avec un groupe d'utilisateurs](#).
5. Pour chaque attribut que vous devez mapper, effectuez les étapes suivantes :
 - a. Sélectionnez un attribut dans la colonne Attribut du groupe d'utilisateurs Il s'agit de l'attribut affecté au profil utilisateur dans votre groupe d'utilisateurs. Les attributs personnalisés sont répertoriés après les attributs standard.
 - b. Sélectionnez un attribut dans la colonne Attribut SAML. Il s'agit de l'attribut transmis à partir du répertoire du fournisseur.

Votre fournisseur d'identité peut proposer des exemples d'assertions SAML à titre de référence. Certains IdPs utilisent des noms simples, tels que `email`, tandis que d'autres utilisent des noms d'attributs au format URL similaires à :

```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

- c. Pour mapper des attributs supplémentaires entre votre IdP et Amazon Cognito, choisissez Ajout d'un autre attribut.

6. Sélectionnez Enregistrer les modifications.

Spécifier les mappages d'attributs des fournisseurs d'identité pour votre groupe d'utilisateurs (AWS CLI et votre AWS API)

Utilisez les commandes suivantes pour spécifier les mappages d'attributs de fournisseur d'identité pour votre groupe d'utilisateurs.

Pour spécifier des mappages d'attributs au moment de la création du fournisseur

- AWS CLI: `aws cognito-idp create-identity-provider`

Exemple avec fichier de métadonnées : `aws cognito-idp create-identity-provider --user-pool-id <user_pool_id> --provider-name=SAML_provider_1 --provider-type SAML --provider-details file:///details.json --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

Où `details.json` contient :

```
{
  "MetadataFile": "<SAML metadata XML>"
}
```

Note

Si `<SAML metadata XML>` contient des guillemets (`"`), ils doivent être placés dans une séquence d'échappement (`\`).

Exemple avec URL de métadonnées :

```
aws cognito-idp create-identity-provider \  
--user-pool-id us-east-1_EXAMPLE \  
--provider-name=SAML_provider_1 \  
--provider-type SAML \  
--provider-details MetadataURL=https://myidp.example.com/saml/metadata \  
--attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/  
emailaddress
```

- AWS API : [CreateIdentityProvider](#)

Pour spécifier des mappages d'attributs pour un fournisseur d'identité existant

- AWS CLI: `aws cognito-idp update-identity-provider`

Exemple : `aws cognito-idp update-identity-provider --user-pool-id
<user_pool_id> --provider-name <provider_name> --attribute-mapping
email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

- AWS API : [UpdateIdentityProvider](#)

Pour obtenir des informations sur le mappage d'attributs pour un fournisseur d'identité spécifique

- AWS CLI: `aws cognito-idp describe-identity-provider`

Exemple : `aws cognito-idp describe-identity-provider --user-pool-id
<user_pool_id> --provider-name <provider_name>`

- AWS API : [DescribeIdentityProvider](#)

Liaison d'utilisateurs fédérés à un profil utilisateur existant

Souvent, le même utilisateur possède un profil auprès de plusieurs fournisseurs d'identité (IdPs) que vous avez connectés à votre groupe d'utilisateurs. Amazon Cognito peut lier chaque occurrence d'un utilisateur au même profil utilisateur dans votre répertoire. De cette façon, une personne ayant plusieurs utilisateurs de fournisseur d'identité peut vivre une expérience cohérente dans votre application. [AdminLinkProviderForUser](#) indique à Amazon Cognito de reconnaître l'identifiant unique d'un utilisateur dans votre annuaire fédéré en tant qu'utilisateur du groupe d'utilisateurs. Un utilisateur de votre groupe d'utilisateurs compte comme un utilisateur actif mensuel dans le cadre

de la [facturation](#) lorsque vous avez zéro, une ou plusieurs identités fédérées associées au profil utilisateur.

Lorsqu'un utilisateur fédéré se connecte à votre groupe d'utilisateurs pour la première fois, Amazon Cognito recherche un profil local que vous avez associé à son identité. S'il n'existe aucun profil lié, votre groupe d'utilisateurs crée un nouveau profil. Vous pouvez créer un profil local et le lier à votre utilisateur fédéré à tout moment avant sa première connexion, dans une demande d'`AdminLinkProviderForUserAPI`, soit dans le cadre d'une tâche de prépréparation planifiée, soit dans un [Déclencheur Lambda Avant l'inscription](#). Une fois que votre utilisateur s'est connecté et qu'Amazon Cognito a détecté un profil local associé, votre groupe d'utilisateurs lit les enregistrements de vos utilisateurs et les compare aux règles de mappage pour l'IdP. Votre groupe d'utilisateurs met ensuite à jour le profil local associé avec les enregistrements mappés à partir de leur connexion. De cette façon, vous pouvez configurer le profil local avec les demandes d'accès et conserver leurs demandes d'identité up-to-date auprès de votre fournisseur. Une fois qu'Amazon Cognito a associé votre utilisateur fédéré à un profil associé, il se connecte toujours à ce profil. Vous pouvez ensuite associer plusieurs identités de fournisseurs de vos utilisateurs au même profil, afin d'offrir à un client une expérience cohérente dans votre application. Pour associer un utilisateur fédéré qui s'est déjà connecté, vous devez d'abord supprimer son profil existant. Vous pouvez identifier les profils existants par leur format : `[Provider name]_identifiant`. Par exemple, `LoginWithAmazon_amzn1.account.AFAEXAMPLE`. Un utilisateur que vous avez créé puis associé à une identité d'utilisateur tiers possède le nom d'utilisateur avec lequel il a été créé et un `identities` attribut contenant les détails de ses identités associées.

Important

Étant donné qu'il `AdminLinkProviderForUser` permet à un utilisateur doté d'une identité fédérée externe de se connecter en tant qu'utilisateur existant dans le groupe d'utilisateurs, il est essentiel de ne l'utiliser qu'avec des attributs externes IdPs et de fournisseur approuvés par le propriétaire de l'application.

Par exemple, si vous êtes un fournisseur de services gérés (MSP) avec une application que vous partagez avec plusieurs clients. Chacun des clients se connecte à votre application via Active Directory Federation Services (ADFS). Votre administrateur informatique, Carlos, possède un compte dans chaque domaine de vos clients. Vous voulez que Carlos soit reconnu comme administrateur d'application chaque fois qu'il se connecte, quel que soit le fournisseur d'identité.

Votre ADFS IdPs présente l'adresse e-mail de Carlos `mcp_carlos@example.com` dans la email réclamation des assertions SAML de Carlos auprès d'Amazon Cognito. Vous créez un utilisateur dans votre groupe d'utilisateurs avec le nom d'utilisateur Carlos. Les commandes AWS Command Line Interface (AWS CLI) suivantes relient les identités de Carlos à celles d' IdPs ADFS1, ADFS2 et ADFS3.

Note

Vous pouvez lier un utilisateur en fonction de revendications d'attributs spécifiques. Cette capacité est propre à OIDC et IdPs SAML. Pour les autres types de fournisseurs, vous devez créer une liaison basée sur un attribut source fixe. Pour plus d'informations, consultez [AdminLinkProviderForUser](#). Vous devez définir `ProviderAttributeName` sur `Cognito_Subject` lorsque vous liez un fournisseur d'identité social à un profil utilisateur. `ProviderAttributeValue` doit être l'identifiant unique de l'utilisateur avec votre fournisseur d'identité.

```
aws cognito-idp admin-link-provider-for-user \  
--user-pool-id us-east-1_EXAMPLE \  
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \  
--source-user  
ProviderName=ADFS1,ProviderAttributeName=email,ProviderAttributeValue=mcp_carlos@example.com  
  
aws cognito-idp admin-link-provider-for-user \  
--user-pool-id us-east-1_EXAMPLE \  
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \  
--source-user  
ProviderName=ADFS2,ProviderAttributeName=email,ProviderAttributeValue=mcp_carlos@example.com  
  
aws cognito-idp admin-link-provider-for-user \  
--user-pool-id us-east-1_EXAMPLE \  
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \  
--source-user  
ProviderName=ADFS3,ProviderAttributeName=email,ProviderAttributeValue=mcp_carlos@example.com
```

Le profil utilisateur Carlos dans votre groupe d'utilisateurs possède désormais l'attribut `identities` suivant.

```
[{  
  "userId": "mcp_carlos@example.com",
```

```
"providerName": "ADFS1",
"providerType": "SAML",
"issuer": "http://auth.example.com",
"primary": false,
"dateCreated": 1111111111111111
}, {
  "userId": "msp_carlos@example.com",
  "providerName": "ADFS2",
  "providerType": "SAML",
  "issuer": "http://auth2.example.com",
  "primary": false,
  "dateCreated": 1111111111111111
}, {
  "userId": "msp_carlos@example.com",
  "providerName": "ADFS3",
  "providerType": "SAML",
  "issuer": "http://auth3.example.com",
  "primary": false,
  "dateCreated": 1111111111111111
}]
```

Points à savoir sur la liaison d'utilisateurs fédérés

- Vous pouvez lier jusqu'à cinq utilisateurs fédérés à chaque profil utilisateur.
- Vous pouvez lier des utilisateurs fédérés à un profil utilisateur fédéré existant ou à un utilisateur local.
- Vous ne pouvez pas associer des fournisseurs à des profils d'utilisateurs dans le AWS Management Console.
- Le jeton d'identification de votre utilisateur contient tous les fournisseurs associés dans la revendication `identities`.
- Vous pouvez définir un mot de passe pour le profil utilisateur fédéré créé automatiquement dans une demande d'API. [AdminSetUserPassword](#) Le statut de cet utilisateur passe alors de `EXTERNAL_PROVIDER` à `CONFIRMED`. Un utilisateur dans cet état peut se connecter en tant qu'utilisateur fédéré et lancer des flux d'authentification dans l'API comme un utilisateur local lié. Ils peuvent également modifier leur mot de passe et leurs attributs dans des requêtes d'API authentifiées par jeton, telles que et. [ChangePasswordUpdateUserAttributes](#) En tant que bonne pratique de sécurité et pour que les utilisateurs restent synchronisés avec votre fournisseur d'identité externe, ne définissez pas de mots de passe sur les profils utilisateur fédérés. Liez plutôt les utilisateurs à des profils locaux avec `AdminLinkProviderForUser`.

- Amazon Cognito renseigne les attributs utilisateur dans un profil utilisateur local lié lorsque l'utilisateur se connecte via son fournisseur d'identité. Amazon Cognito traite les champs standard d'identité contenus dans le jeton d'identification provenant d'un fournisseur d'identité OIDC, et vérifie également le point de terminaison `userInfo` des fournisseurs OAuth 2.0 et OIDC. Amazon Cognito donne la priorité aux informations contenues dans un jeton d'identification par rapport aux informations provenant de `userInfo`.

Lorsque vous apprenez que votre utilisateur n'utilise plus un compte utilisateur externe que vous avez associé à son profil, vous pouvez dissocier ce compte utilisateur de l'utilisateur de votre groupe d'utilisateurs. Lorsque vous avez associé votre utilisateur, vous avez fourni le nom de l'attribut, la valeur de l'attribut et le nom du fournisseur de l'utilisateur dans la demande. Pour supprimer un profil dont votre utilisateur n'a plus besoin, envoyez une demande d'[AdminDisableProviderForUserAPI](#) avec des paramètres équivalents.

[AdminLinkProviderForUser](#) Reportez-vous à la section Syntaxe de commande et exemples supplémentaires dans les AWS SDK.

Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda

Amazon Cognito utilise des fonctions AWS Lambda permettant de modifier le comportement d'authentification de votre groupe d'utilisateurs. Vous pouvez configurer votre groupe d'utilisateurs pour qu'il invoque automatiquement les fonctions Lambda avant la première inscription, une fois l'authentification terminée, et à plusieurs étapes intermédiaires. Vos fonctions peuvent modifier le comportement par défaut de votre flux d'authentification, effectuer des demandes d'API pour modifier votre groupe d'utilisateurs ou d'autres ressources AWS, et communiquer avec des systèmes externes. Le code de vos fonctions Lambda vous appartient. Amazon Cognito envoie des données d'événement à votre fonction, attend que la fonction traite les données et, dans la plupart des cas, anticipe un événement de réponse qui reflète les modifications que vous souhaitez apporter à la session.

Dans le système des événements de demande et de réponse, vous pouvez introduire vos propres défis d'authentification, faire migrer les utilisateurs entre votre groupe d'utilisateurs et un autre magasin d'identités, personnaliser les messages et modifier les jetons Web JSON (JWT).

Les déclencheurs Lambda peuvent personnaliser la réponse qu'Amazon Cognito fournit à l'utilisateur une fois qu'il a lancé une action dans votre groupe d'utilisateurs. Par exemple, vous pouvez

empêcher la connexion d'un utilisateur qui réussirait autrement à se connecter. Ils peuvent également exécuter des opérations d'exécution sur votre environnement AWS, les API externes, les bases de données et les magasins d'identités. Le déclencheur de migration d'utilisateur, par exemple, peut associer une action externe à une modification dans Amazon Cognito : vous pouvez rechercher les informations utilisateur dans un annuaire externe, puis définir les attributs d'un nouvel utilisateur en fonction de ces informations externes.

Quand un déclencheur Lambda est attribué à votre groupe d'utilisateurs, Amazon Cognito interrompt son flux par défaut pour demander des informations à votre fonction. Amazon Cognito génère un événement JSON et le transmet à votre fonction. Cet événement contient des informations concernant la demande de votre utilisateur de créer un compte d'utilisateur, de se connecter, de réinitialiser un mot de passe ou de mettre à jour un attribut. Votre fonction a alors la possibilité d'agir ou de renvoyer l'événement sans le modifier.

Le tableau suivant résume certaines des façons dont vous pouvez utiliser les déclencheurs Lambda pour personnaliser les opérations de groupe d'utilisateurs :

Flux de groupe d'utilisateurs	Opération	Description
Flux d'authentification personnalisé	Définition de la stimulation d'authentification	Détermine la question de sécurité (stimulation) dans un flux d'authentification personnalisé
	Création d'une stimulation d'authentification	Crée une question de sécurité (stimulation) dans un flux d'authentification personnalisé
	Vérification de la réponse à la stimulation d'authentification	Détermine si une réponse est correcte dans un flux d'authentification personnalisé
Événements d'authentification	the section called “Déclencheur Lambda avant l'authentification.”	Validation personnalisée pour accepter ou refuser la demande de connexion

Flux de groupe d'utilisateurs	Opération	Description
	the section called “Déclencheur Lambda après l'authentification.”	Enregistre les événements pour une analytique personnalisée
	the section called “Déclencheur Lambda avant la génération de jeton”	Augmente ou supprime les revendications de jeton
Inscription	the section called “Déclencheur Lambda Avant l'inscription”	Effectue une validation personnalisée qui accepte ou refuse la demande d'inscription
	the section called “Déclencheur Lambda après confirmation.”	Ajoute des messages personnalisés de bienvenue ou une journalisation des événements pour une analytique personnalisée
	the section called “Déclencheur Lambda de migration d'utilisateur”	Effectue la migration d'un utilisateur à partir d'un annuaire d'utilisateurs existant vers des groupes d'utilisateurs
Messages	the section called “Déclencheur Lambda message personnalisé”	Effectue une personnalisation et une localisation avancées de messages
Création de jeton	the section called “Déclencheur Lambda avant la génération de jeton”	Ajoute ou supprime des attributs dans les jetons d'ID
Fournisseurs tiers d'e-mails et de SMS	the section called “Déclencheurs Lambda Expéditeur personnalisé”	Utilise un fournisseur tiers pour envoyer des SMS et des e-mails

Rubriques

- [Considérations Importantes](#)
- [Ajout d'un déclencheur Lambda à un groupe d'utilisateurs](#)
- [Événement déclencheur Lambda d'un groupe d'utilisateurs](#)
- [Paramètres communs des déclencheurs Lambda de groupe d'utilisateurs](#)
- [Connexion d'opérations d'API aux déclencheurs Lambda](#)
- [Connexion de déclencheurs Lambda aux opérations fonctionnelles du groupe d'utilisateurs](#)
- [Déclencheur Lambda Avant l'inscription](#)
- [Déclencheur Lambda après confirmation.](#)
- [Déclencheur Lambda avant authentification](#)
- [Déclencheur Lambda après l'authentification](#)
- [Déclencheurs Lambda création d'une stimulation d'authentification personnalisée](#)
- [Déclencheur Lambda avant génération de jeton](#)
- [Déclencheur Lambda de migration d'utilisateur](#)
- [Déclencheur Lambda message personnalisé](#)
- [Déclencheurs Lambda Expéditeur personnalisé](#)

Considérations Importantes

Lorsque vous préparez vos groupes d'utilisateurs pour les fonctions Lambda, prenez en compte les éléments suivants :

- Les événements qu'Amazon Cognito envoie aux déclencheurs Lambda peuvent changer en fonction des nouvelles fonctionnalités. Les positions des éléments de réponse et de demande dans la hiérarchie JSON peuvent changer, ou des noms d'éléments peuvent être ajoutés. Dans votre fonction Lambda, vous pouvez vous attendre à recevoir les paires clé-valeur d'entrée décrites dans ce guide, mais une validation d'entrée plus stricte peut entraîner l'échec de vos fonctions.
- Vous pouvez choisir l'une des nombreuses versions des événements qu'Amazon Cognito envoie à certains déclencheurs. Certaines versions peuvent nécessiter que vous acceptiez une modification de la tarification d'Amazon Cognito. Pour de plus amples informations sur la tarification, consultez [Tarification d'Amazon Cognito](#). Pour personnaliser les jetons d'accès dans une [Déclencheur Lambda avant génération de jeton](#), vous devez configurer votre groupe d'utilisateurs avec des

[fonctionnalités de sécurité avancées](#) et mettre à jour la configuration de votre déclencheur Lambda pour utiliser la version 2 de l'événement.

- À l'exception des [Déclencheurs Lambda Expéditeur personnalisé](#), Amazon Cognito appelle les fonctions Lambda de manière synchrone. Quand Amazon Cognito appelle votre fonction Lambda, elle doit répondre dans un délai de 5 secondes. Si elle ne le fait pas et si l'appel peut être retenté, Amazon Cognito retente l'appel. Après trois tentatives infructueuses, la fonction expire. Vous ne pouvez pas modifier ce délai d'attente de cinq secondes. Pour plus d'informations, consultez [Modèle de programmation Lambda](#) dans le Guide du développeur AWS Lambda.

Amazon Cognito ne réessaie pas les appels de fonction qui renvoient une [erreur Invoke](#) avec un code d'état HTTP compris entre 500 et 599. Ces codes indiquent un problème de configuration qui empêche Lambda de lancer la fonction. Pour plus d'informations, consultez [Gestion des erreurs et tentatives automatiques dans AWS Lambda](#).

- Vous ne pouvez pas déclarer une version de fonction dans la configuration de votre déclencheur Lambda. Les groupes d'utilisateurs Amazon Cognito invoquent la dernière version de votre fonction par défaut. Cependant, vous pouvez associer une version de fonction à un alias et définir votre déclencheur LambdaArn sur l'ARN d'alias dans une demande d'API [CreateUserPool](#) ou [UpdateUserPool](#). Cette option n'est pas disponible dans la AWS Management Console. Pour plus d'informations sur l'utilisation des alias, consultez [Alias de fonction Lambda](#) dans le Guide du développeur AWS Lambda.
- Si vous supprimez un déclencheur Lambda, vous devez mettre à jour le déclencheur correspondant dans le groupe d'utilisateurs. Par exemple, si vous supprimez le déclencheur d'authentification, vous devez définir le déclencheur Post authentication (Après l'authentification) dans le groupe d'utilisateurs correspondant sur none.
- Si votre fonction Lambda ne renvoie pas les paramètres de demande et de réponse à Amazon Cognito, ou renvoie une erreur, l'événement d'authentification échoue. Vous pouvez renvoyer une erreur dans votre fonction pour empêcher l'inscription d'un utilisateur, son authentification, la génération de jetons ou toute autre phase de son flux d'authentification qui invoque le déclencheur Lambda.

L'interface utilisateur hébergée par Amazon Cognito renvoie les erreurs générées par les déclencheurs Lambda sous forme de texte d'erreur au-dessus de l'invite de connexion.

L'API des groupes d'utilisateurs Amazon Cognito renvoie les erreurs de déclenchement au format `[trigger] failed with error [error text from response]`. Conformément à la bonne pratique, générez uniquement les erreurs dans vos fonctions Lambda que vous voulez que les utilisateurs voient. Utilisez des méthodes de sortie telles que `print()` pour consigner

toutes les informations sensibles ou de débogage dans CloudWatch Logs. Pour voir un exemple, consultez [Exemple de pré-inscription : Refuser l'inscription si le nom d'utilisateur comporte moins de cinq caractères](#).

- Vous pouvez ajouter une fonction Lambda dans un autre Compte AWS en tant que déclencheur pour votre groupe d'utilisateurs. Vous devez ajouter des déclencheurs entre comptes avec les opérations d'API [CreateUserPool](#) et [UpdateUserPool](#), ou leurs équivalents dans AWS CloudFormation et l'AWS CLI. Vous ne pouvez pas ajouter de fonctions entre comptes dans la AWS Management Console.
- Lorsque vous ajoutez un déclencheur Lambda dans la console Amazon Cognito, Amazon Cognito ajoute une politique basée sur les ressources à votre fonction qui permet à votre groupe d'utilisateurs d'appeler la fonction. Lorsque vous créez un déclencheur Lambda en dehors de la console Amazon Cognito, y compris une fonction entre comptes, vous devez ajouter des autorisations à la politique basée sur les ressources de la fonction Lambda. Les autorisations que vous avez ajoutées doivent autoriser Amazon Cognito à appeler la fonction au nom de votre groupe d'utilisateurs. Vous pouvez [ajouter des autorisations à partir de la console Lambda](#) ou utiliser l'opération d'API Lambda [AddPermission](#).

Exemple de politique basée sur les ressources Lambda

L'exemple suivant de politique basée sur les ressources Lambda accorde une capacité limitée à Amazon Cognito pour appeler une fonction Lambda. Amazon Cognito ne peut invoquer la fonction que lorsqu'elle le fait au nom du groupe d'utilisateurs dans le `aws:SourceArn` et le compte dans la Condition `aws:SourceAccount`.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "lambda-allow-cognito",
      "Effect": "Allow",
      "Principal": {
        "Service": "cognito-idp.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "<your Lambda function ARN>",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "<your account number>"
        }
      }
    }
  ]
}
```

```
    },
    "ArnLike": {
      "AWS:SourceArn": "<your user pool ARN>"
    }
  }
]
}
```

Ajout d'un déclencheur Lambda à un groupe d'utilisateurs

Pour ajouter un déclencheur Lambda à un groupe d'utilisateurs à l'aide de la console

1. Utilisez la [console Lambda](#) pour créer une fonction Lambda. Pour plus d'informations sur les fonctions Lambda, consultez le [Guide du développeur AWS Lambda](#).
2. Accédez à la [console Amazon Cognito](#), puis choisissez User Pools (Groupes d'utilisateurs).
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
4. Choisissez l'onglet User pool properties (Propriétés du groupe d'utilisateurs) et localisez Lambda triggers (Déclencheurs Lambda).
5. Choisissez Ajouter un déclencheur Lambda.
6. Sélectionnez un déclencheur Lambda Category (Catégorie) en fonction de l'étape de l'authentification que vous souhaitez personnaliser.
7. Sélectionnez Affecter une fonction Lambda, puis une fonction dans la même Région AWS comme votre groupe d'utilisateurs.

Note

Si vos informations d'identification AWS Identity and Access Management (IAM) sont autorisées à mettre à jour la fonction Lambda, Amazon Cognito ajoute une politique basée sur les ressources Lambda. Avec cette politique, Amazon Cognito peut invoquer la fonction que vous sélectionnez. Si les informations d'identification connectées ne disposent pas d'autorisations IAM suffisantes, vous devez mettre à jour la politique basée sur les ressources séparément. Pour de plus amples informations, veuillez consulter [the section called "Considérations Importantes"](#).

8. Choisissez Enregistrer les modifications.

9. Vous pouvez utiliser CloudWatch dans la console Lambda pour consigner votre fonction Lambda. Pour plus d'informations, consultez [Accès à CloudWatch Logs pour Lambda](#).

Événement déclencheur Lambda d'un groupe d'utilisateurs

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. La fonction Lambda renvoie à Amazon Cognito le même objet d'événement, avec des modifications dans la réponse. Cet événement affiche les paramètres communs des déclencheurs Lambda :

JSON

```
{
  "version": "string",
  "triggerSource": "string",
  "region": AWSRegion,
  "userPoolId": "string",
  "userName": "string",
  "callerContext":
    {
      "awsSdkVersion": "string",
      "clientId": "string"
    },
  "request":
    {
      "userAttributes": {
        "string": "string",
        ....
      }
    },
  "response": {}
}
```

Paramètres communs des déclencheurs Lambda de groupe d'utilisateurs

version ;

Numéro de version de votre fonction Lambda.

triggerSource

Nom de l'événement qui a déclenché la fonction Lambda. Pour obtenir une description de chaque triggerSource, consultez [Connexion de déclencheurs Lambda aux opérations fonctionnelles du groupe d'utilisateurs](#).

region

La Région AWS en tant qu'instance `AWSRegion`.

userPoolId

ID du groupe d'utilisateurs.

userName

Nom d'utilisateur de l'utilisateur actuel.

callerContext

Métadonnées relatives à la demande et à l'environnement de code. Elles contiennent les champs `awsSdkVersion` et `clientId`.

awsSdkVersion

Version du kit AWS SDK qui a généré la demande.

clientId

Identifiant du client d'application du groupe d'utilisateurs.

de la demande

Détails de la demande d'API de l'utilisateur. Cela inclut les champs suivants, ainsi que tous les paramètres de demande spécifiques au déclencheur. Par exemple, un événement envoyé par Amazon Cognito à un déclencheur avant authentification contient également un paramètre `userNotFound`. Vous pouvez traiter la valeur de ce paramètre pour effectuer une action personnalisée lorsque votre utilisateur essaie de se connecter avec un nom d'utilisateur non enregistré.

userAttributes

Une ou plusieurs paires clé-valeur de noms et de valeurs d'attributs utilisateur, par exemple `"email": "john@example.com"`.

réponse

Ce paramètre ne contient aucune information dans la demande d'origine. Votre fonction Lambda doit renvoyer l'intégralité de l'événement à Amazon Cognito et ajouter tous les paramètres de retour au paramètre `response`. Pour voir quels paramètres de retour votre fonction peut inclure, consultez la documentation du déclencheur que vous souhaitez utiliser.

Connexion d'opérations d'API aux déclencheurs Lambda

Les sections suivantes décrivent les déclencheurs Lambda qu'Amazon Cognito invoque à partir de l'activité dans votre groupe d'utilisateurs.

Quand votre application connecte des utilisateurs via l'API des groupes d'utilisateurs Amazon Cognito, l'interface utilisateur hébergée ou les points de terminaison du groupe d'utilisateurs, Amazon Cognito invoque vos fonctions Lambda en fonction du contexte de session. Pour plus d'informations sur l'API des groupes d'utilisateurs Amazon Cognito et les points de terminaison du groupe d'utilisateurs, consultez [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#). Les tableaux des sections suivantes décrivent les événements qui amènent Amazon Cognito à invoquer une fonction, ainsi que la chaîne `triggerSource` qu'Amazon Cognito inclut dans la demande.

Rubriques

- [Déclencheurs Lambda dans l'API Amazon Cognito](#)
- [Déclencheurs Lambda pour les utilisateurs locaux Amazon Cognito dans l'interface utilisateur hébergée](#)
- [Déclencheurs Lambda pour utilisateurs fédérés](#)

Déclencheurs Lambda dans l'API Amazon Cognito

Le tableau suivant décrit les chaînes sources des déclencheurs Lambda qu'Amazon Cognito peut invoquer quand votre application crée, connecte ou met à jour un utilisateur local.

Sources des déclencheurs d'utilisateur local dans l'API Amazon Cognito

Opération API	Déclencheur Lambda	Source du déclencheur
AdminCreateUser	Avant l'inscription	PreSignUp_AdminCreateUser

Opération API	Déclencheur Lambda	Source du déclencheur
	Pre token generation	TokenGeneration_NewPasswordChallenge
	Message personnalisé	CustomMessage_AdminCreateUser
	Expéditeur d'e-mail personnalisé	CustomEmailSender_AdminCreateUser
	Expéditeur de SMS personnalisé	CustomSMSSender_AdminCreateUser
SignUp	Avant l'inscription	PreSignUp_SignUp
	Message personnalisé	CustomMessage_SignUp
	Expéditeur d'e-mail personnalisé	CustomEmailSender_SignUp
	Expéditeur de SMS personnalisé	CustomSMSSender_SignUp
ConfirmSignUp AdminConfirmSignUp	Après la confirmation	PostConfirmation_ConfirmSignUp
InitiateAuth AdminInitiateAuth	Avant l'authentification	PreAuthentication_Authentication
	Define auth challenge	DefineAuthChallenge_Authentication
	Create auth challenge	CreateAuthChallenge_Authentication

Opération API	Déclencheur Lambda	Source du déclencheur
	Pre token generation	TokenGeneration_Authentication TokenGeneration_AuthenticateDevice TokenGeneration_RefreshTokens
	Migration de l'utilisateur (Migrate user)	UserMigration_Authentication
	Message personnalisé	CustomMessage_Authentication
	Expéditeur d'e-mail personnalisé	CustomEmailSender_AccountTakeOverNotification
	Expéditeur de SMS personnalisé	CustomSMSSender_Authentication
ForgotPassword	Migration de l'utilisateur (Migrate user)	UserMigration_ForgotPassword
	Message personnalisé	CustomMessage_ForgotPassword
	Expéditeur d'e-mail personnalisé	CustomEmailSender_ForgotPassword
	Expéditeur de SMS personnalisé	CustomSMSSender_ForgotPassword
ConfirmForgotPassword	Après la confirmation	PostConfirmation_ConfirmForgotPassword

Opération API	Déclencheur Lambda	Source du déclencheur
UpdateUserAttributes AdminUpdateUserAttributes	Message personnalisé	CustomMessage_UpdateUserAttribute
	Expéditeur d'e-mail personnalisé	CustomEmailSender_UpdateUserAttribute
	Expéditeur de SMS personnalisé	CustomSMSSender_UpdateUserAttribute
VerifyUserAttributes	Message personnalisé	CustomMessage_VerifyUserAttribute
	Expéditeur d'e-mail personnalisé	CustomEmailSender_VerifyUserAttribute
	Expéditeur de SMS personnalisé	CustomSMSSender_VerifyUserAttribute

Déclencheurs Lambda pour les utilisateurs locaux Amazon Cognito dans l'interface utilisateur hébergée

Le tableau suivant décrit les chaînes sources des déclencheurs Lambda qu'Amazon Cognito peut invoquer quand un utilisateur local se connecte à votre groupe d'utilisateurs avec l'interface utilisateur hébergée.

Sources des déclencheurs d'utilisateur locaux dans l'interface utilisateur hébergée

URI d'interface utilisateur hébergée	Déclencheur Lambda	Source du déclencheur
/signup	Avant l'inscription	PreSignUp_SignUp
	Message personnalisé	CustomMessage_SignUp
	Expéditeur d'e-mail personnalisé	CustomEmailSender_SignUp

URI d'interface utilisateur hébergée	Déclencheur Lambda	Source du déclencheur	
	Expéditeur de SMS personnalisé	CustomSMSSender_SignUp	
/confirmuser	Après la confirmation	PostConfirmation_ConfirmSignUp	
/login	Avant l'authentification	PreAuthentication_Authentication	
	Define auth challenge	DefineAuthChallenge_Authentication	
	Create auth challenge	CreateAuthChallenge_Authentication	
	Pre token generation		TokenGeneration_Authentication
			TokenGeneration_AuthenticateDevice
			TokenGeneration_RefreshTokens
	Migration de l'utilisateur (Migrate user)	UserMigration_Authentication	
	Message personnalisé	CustomMessage_Authentication	
Expéditeur d'e-mail personnalisé	CustomEmailSender_AccountTakeOverNotification		
Expéditeur de SMS personnalisé	CustomSMSSender_Authentication		

URI d'interface utilisateur hébergée	Déclencheur Lambda	Source du déclencheur
/forgotpassword	Migration de l'utilisateur (Migrate user)	UserMigration_ForgotPassword
	Message personnalisé	CustomMessage_ForgotPassword
	Expéditeur d'e-mail personnalisé	CustomEmailSender_ForgotPassword
	Expéditeur de SMS personnalisé	CustomSMSSender_ForgotPassword
/confirmforgotpassword	Après la confirmation	PostConfirmation_ConfirmForgotPassword

Déclencheurs Lambda pour utilisateurs fédérés

Vous pouvez utiliser les déclencheurs Lambda suivants pour personnaliser les flux de travail de votre groupe d'utilisateurs pour les utilisateurs qui se connectent avec un fournisseur fédéré.

Note

Les utilisateurs fédérés peuvent utiliser l'interface utilisateur hébergée par Amazon Cognito pour se connecter, ou vous pouvez générer une demande auprès du [Point de terminaison d'autorisation](#), qui les redirige silencieusement vers la page de connexion de leur fournisseur d'identité. Vous ne pouvez pas connecter des utilisateurs fédérés avec l'API des groupes d'utilisateurs Amazon Cognito.

Sources des déclencheurs d'utilisateurs fédérés

Événement de connexion	Déclencheur Lambda	Source du déclencheur
Première connexion	Avant l'inscription	PreSignUp_ExternalProvider
	Après la confirmation	PostConfirmation_ConfirmSignUp
	Pre token generation	TokenGeneration_HostedAuth
Connexions suivantes	Avant l'authentification	PreAuthentication_Authentication
	Après l'authentification	PostAuthentication_Authentication
	Pre token generation	TokenGeneration_HostedAuth

La connexion fédérée n'appelle aucun [Déclencheurs Lambda création d'une stimulation d'authentification personnalisée](#), [Déclencheur Lambda de migration d'utilisateur](#), [Déclencheur Lambda message personnalisé](#) ou [Déclencheurs Lambda Expéditeur personnalisé](#) dans votre groupe d'utilisateurs.

Connexion de déclencheurs Lambda aux opérations fonctionnelles du groupe d'utilisateurs

Chaque déclencheur Lambda joue un rôle fonctionnel dans votre groupe d'utilisateurs. Par exemple, un déclencheur peut modifier votre flux d'inscription ou ajouter une stimulation d'authentification personnalisée. L'événement qu'Amazon Cognito envoie à une fonction Lambda peut refléter l'une des actions qui constituent ce rôle fonctionnel. Par exemple, Amazon Cognito invoque un déclencheur avant inscription lorsque votre utilisateur s'inscrit et quand vous créez un utilisateur. Chacun de ces différents cas pour le même rôle fonctionnel a sa propre valeur `triggerSource`. Votre fonction Lambda peut traiter les événements entrants différemment en fonction de l'opération qui l'a invoquée.

Amazon Cognito invoque également toutes les fonctions affectées quand un événement correspond à une source de déclencheur. Par exemple, quand un utilisateur se connecte à un groupe d'utilisateurs dans lequel vous avez attribué des déclencheurs de migration d'utilisateur et avant authentification, il active les deux.

Déclencheurs d'inscription, de confirmation et de connexion (authentification)

Déclencheur	Valeur triggerSource	Événement
Avant l'inscription	PreSignUp_SignUp	Avant l'inscription.
Avant l'inscription	PreSignUp_AdminCreateUser	Pré-inscription quand un administrateur crée un nouvel utilisateur.
Avant l'inscription	PreSignUp_ExternalProvider	Avant l'inscription pour les fournisseurs d'identité externes.
Après la confirmation	PostConfirmation_ConfirmSignUp	Confirmation après l'inscription.
Après la confirmation	PostConfirmation_ConfirmForgotPassword	Confirmation après l'oubli de mot de passe de confirmation.
Avant l'authentification	PreAuthentication_Authentication	Avant l'authentification.
Après l'authentification	PostAuthentication_Authentication	Après l'authentification.

Déclencheurs de stimulation d'authentification personnalisés

Déclencheur	Valeur triggerSource	Événement
Define auth challenge	DefineAuthChallenge_Authentication	Définition de la stimulation d'authentification.

Déclencheur	Valeur triggerSource	Événement
Create auth challenge	CreateAuthChallenge_Authentication	Création d'une stimulation d'authentification.
Vérification de la stimulation d'authentification	VerifyAuthChallengeResponse_Authentication	Vérification de la réponse à la stimulation d'authentification.

Déclencheurs avant génération de jeton

Déclencheur	Valeur triggerSource	Événement
Pre token generation	TokenGeneration_HostedAuth	Amazon Cognito authentifie l'utilisateur à partir de votre page de connexion de l'interface utilisateur hébergée.
Pre token generation	TokenGeneration_Authentication	Les flux d'authentification utilisateur sont terminés.
Pre token generation	TokenGeneration_NewPasswordChallenge	L'administrateur crée l'utilisateur. Amazon Cognito appelle cela quand l'utilisateur doit modifier un mot de passe temporaire.
Pre token generation	TokenGeneration_AuthenticateDevice	Fin de l'authentification d'un appareil utilisateur.
Pre token generation	TokenGeneration_RefreshTokens	L'utilisateur tente d'actualiser les jetons d'identité et d'accès.

Déclencheur de migration d'utilisateur

Déclencheur	Valeur triggerSource	Événement
Migration d'utilisateur	UserMigration_Authentication	Migration de l'utilisateur au moment de la connexion.
Migration d'utilisateur	UserMigration_ForgotPassword	Migration de l'utilisateur dans le cadre du flux de mot de passe oublié.

Déclencheurs de message personnalisé

Déclencheur	Valeur triggerSource	Événement
Message personnalisé	CustomMessage_SignUp	Message personnalisé lorsqu'un utilisateur s'inscrit dans votre groupe d'utilisateurs.
Message personnalisé	CustomMessage_AdminCreateUser	Message personnalisé lorsque vous créez un utilisateur en tant qu'administrateur et qu'Amazon Cognito lui envoie un mot de passe temporaire.
Message personnalisé	CustomMessage_ResendCode	Message personnalisé lorsque l'utilisateur existant demande un nouveau code de confirmation.
Message personnalisé	CustomMessage_ForgotPassword	Message personnalisé lorsque l'utilisateur demande la réinitialisation d'un mot de passe.
Message personnalisé	CustomMessage_UpdateUserAttribute	Message personnalisé lorsqu'un utilisateur change son adresse e-mail ou son numéro de téléphone et

Déclencheur	Valeur triggerSource	Événement
		qu'Amazon Cognito envoie un code de vérification.
Message personnalisé	CustomMessage_VerifyUserAttribute	Message personnalisé lorsqu'un utilisateur ajoute une adresse e-mail ou un numéro de téléphone et qu'Amazon Cognito envoie un code de vérification.
Message personnalisé	CustomMessage_Authentication	Message personnalisé lorsqu'un utilisateur ayant configuré l'authentification MFA basée sur SMS se connecte.

Déclencheur Lambda Avant l'inscription

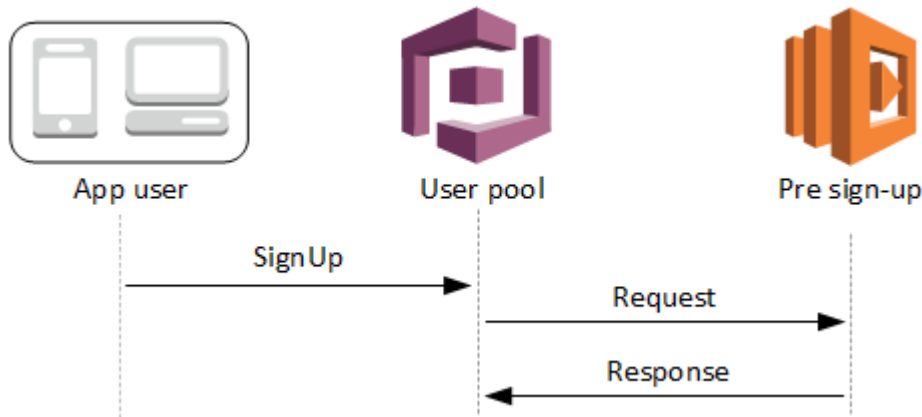
Peu avant qu'Amazon Cognito inscrive un nouvel utilisateur, le service active la fonction AWS Lambda de pré-inscription. Dans le cadre du processus d'inscription, vous pouvez utiliser cette fonction pour effectuer une validation personnalisée et, en fonction des résultats de cette validation, accepter ou rejeter la demande d'enregistrement.

Rubriques

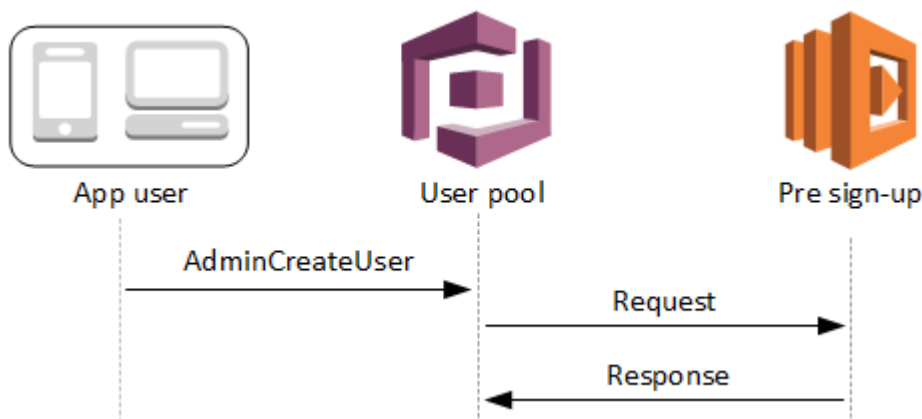
- [Flux Lambda Avant l'inscription](#)
- [Paramètres du déclencheur Lambda avant l'inscription](#)
- [Didacticiels relatifs à l'inscription](#)
- [Exemple d'avant inscription : confirmation automatique d'utilisateurs à partir d'un domaine enregistré](#)
- [Exemple d'avant inscription : confirmation et vérification automatiques de tous les utilisateurs](#)
- [Exemple de pré-inscription : Refuser l'inscription si le nom d'utilisateur comporte moins de cinq caractères](#)

Flux Lambda Avant l'inscription

Flux d'inscription client



Flux d'inscription serveur



La demande inclut les données de validation provenant du client. Ces données proviennent des `ValidationData` valeurs transmises au pool d'utilisateurs `SignUp` et aux méthodes de `AdminCreateUser` l'API.

Paramètres du déclencheur Lambda avant l'inscription

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```

{
  "request": {
    "userAttributes": {
  
```

```
        "string": "string",
        . . .
    },
    "validationData": {
        "string": "string",
        . . .
    },
    "clientMetadata": {
        "string": "string",
        . . .
    }
},

"response": {
    "autoConfirmUser": "boolean",
    "autoVerifyPhone": "boolean",
    "autoVerifyEmail": "boolean"
}
}
```

Paramètres de demande avant l'inscription

userAttributes

Une ou plusieurs paires nom-valeur représentant des attributs utilisateur. Les noms d'attributs sont les clés.

validationData

Une ou plusieurs paires clé-valeur avec les données d'attribut utilisateur que votre application a transmises à Amazon Cognito dans la demande de création d'un utilisateur. Envoyez ces informations à votre fonction Lambda dans les ValidationData paramètres de votre demande [AdminCreateUser](#) ou de votre demande d'[SignUpAPI](#).

Amazon Cognito ne définit pas vos ValidationData données en tant qu'attributs de l'utilisateur que vous créez. ValidationData sont des informations utilisateur temporaires que vous fournissez aux fins de votre déclencheur Lambda préalable à l'inscription.

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée pour la fonction Lambda que vous spécifiez pour le déclencheur Avant l'inscription. Vous pouvez transmettre ces données à votre fonction Lambda en utilisant le ClientMetadata paramètre dans

les actions d'API suivantes : [AdminCreateUser](#), [AdminRespondToAuthChallengeForgotPassword](#), et [SignUp](#)

Paramètres de réponse avant inscription

Dans la réponse, vous pouvez définir `autoConfirmUser` sur `true` si vous voulez confirmer automatiquement l'utilisateur. Vous pouvez définir `autoVerifyEmail` sur `true` pour vérifier automatiquement l'e-mail de l'utilisateur. Vous pouvez définir `autoVerifyPhone` sur `true` pour vérifier automatiquement le numéro de téléphone de l'utilisateur.

Note

Les paramètres de réponse `autoVerifyPhone`, `autoVerifyEmail` et `autoConfirmUser` sont ignorés par Amazon Cognito quand l'API `AdminCreateUser` déclenche la fonction Lambda avant l'inscription.

`autoConfirmUser`

Défini sur `true` pour confirmer automatiquement l'utilisateur, ou sur `false` dans le cas contraire.

`autoVerifyEmail`

Affectez-lui la valeur `true` pour que l'adresse e-mail d'un utilisateur qui s'inscrit soit définie comme vérifiée, sinon affectez-lui la valeur `false`. Si `autoVerifyEmail` est défini sur `true`, l'attribut `email` doit comporter une valeur valide et non nulle. Dans le cas contraire, une erreur survient et l'utilisateur ne pourra pas finaliser l'inscription.

Si l'attribut `email` est sélectionné en tant qu'alias, un alias est créé pour l'adresse e-mail de l'utilisateur quand le paramètre `autoVerifyEmail` est défini. S'il existe déjà un alias avec cette adresse e-mail, l'alias est déplacé vers le nouvel utilisateur et l'adresse e-mail de l'utilisateur précédent est marquée comme non vérifiée. Pour plus d'informations, consultez [Personnalisation des attributs de connexion](#).

`autoVerifyPhone`

Définir sur `true` pour que le numéro de téléphone d'un utilisateur qui s'inscrit soit défini comme vérifié, sinon définir sur `false`. Si `autoVerifyPhone` est défini sur `true`, l'attribut `phone_number` doit comporter une valeur valide et non nulle. Dans le cas contraire, une erreur survient et l'utilisateur ne pourra pas finaliser l'inscription.

Si l'attribut `phone_number` est sélectionné en tant qu'alias, un alias est créé pour le numéro de téléphone de l'utilisateur lorsque le paramètre `autoVerifyPhone` est défini. Si un alias existe déjà avec ce numéro de téléphone, l'alias sera déplacé vers le nouvel utilisateur et le numéro de téléphone de l'utilisateur précédent sera marqué comme non vérifié. Pour de plus amples informations, veuillez consulter [Personnalisation des attributs de connexion](#).

Didacticiels relatifs à l'inscription

La fonction Lambda Avant l'inscription est déclenchée juste avant l'inscription d'un utilisateur. Consultez ces didacticiels d'inscription à Amazon Cognito pour JavaScript Android et iOS.

Plateforme	Didacticiel
JavaScript SDK d'identité	Inscrivez les utilisateurs avec JavaScript
Kit SDK d'identité pour Android	Inscrivez des utilisateurs avec Android
Kit SDK d'identité pour iOS	Inscrivez des utilisateurs avec iOS

Exemple d'avant inscription : confirmation automatique d'utilisateurs à partir d'un domaine enregistré

Vous pouvez utiliser le déclencheur Lambda de pré-inscription afin d'ajouter une logique personnalisée pour valider les nouveaux utilisateurs qui s'inscrivent pour votre groupe d'utilisateurs. Il s'agit d'un exemple de JavaScript programme qui montre comment inscrire un nouvel utilisateur. Il appelle un déclencheur Lambda de pré-inscription dans le cadre de l'authentification.

JavaScript

```
var attributeList = [];  
var dataEmail = {  
  Name: "email",  
  Value: "...", // your email here  
};  
var dataPhoneNumber = {  
  Name: "phone_number",
```

```
    Value: "...", // your phone number here with +country code and no delimiters in
    front
  };

  var dataEmailDomain = {
    Name: "custom:domain",
    Value: "example.com",
  };
  var attributeEmail = new AmazonCognitoIdentity.CognitoUserAttribute(dataEmail);
  var attributePhoneNumber = new AmazonCognitoIdentity.CognitoUserAttribute(
    dataPhoneNumber
  );
  var attributeEmailDomain = new AmazonCognitoIdentity.CognitoUserAttribute(
    dataEmailDomain
  );

  attributeList.push(attributeEmail);
  attributeList.push(attributePhoneNumber);
  attributeList.push(attributeEmailDomain);

  var cognitoUser;
  userPool.signUp(
    "username",
    "password",
    attributeList,
    null,
    function (err, result) {
      if (err) {
        alert(err);
        return;
      }
      cognitoUser = result.user;
      console.log("user name is " + cognitoUser.getUsername());
    }
  );
```

Voici un exemple de déclencheur Lambda appelé juste avant l'inscription avec le déclencheur Lambda Avant l'inscription du groupe d'utilisateurs. Il utilise un attribut personnalisé `custom:domain` pour confirmer automatiquement les nouveaux utilisateurs à partir d'un domaine de messagerie en particulier. Tous les nouveaux utilisateurs ne figurant pas dans le domaine personnalisé seront ajoutés au groupe d'utilisateurs, mais ne seront pas automatiquement confirmés.

Node.js

```
exports.handler = (event, context, callback) => {
  // Set the user pool autoConfirmUser flag after validating the email domain
  event.response.autoConfirmUser = false;

  // Split the email address so we can compare domains
  var address = event.request.userAttributes.email.split("@");

  // This example uses a custom attribute "custom:domain"
  if (event.request.userAttributes.hasOwnProperty("custom:domain")) {
    if (event.request.userAttributes["custom:domain"] === address[1]) {
      event.response.autoConfirmUser = true;
    }
  }

  // Return to Amazon Cognito
  callback(null, event);
};
```

Python

```
def lambda_handler(event, context):
    # It sets the user pool autoConfirmUser flag after validating the email domain
    event['response']['autoConfirmUser'] = False

    # Split the email address so we can compare domains
    address = event['request']['userAttributes']['email'].split('@')

    # This example uses a custom attribute 'custom:domain'
    if 'custom:domain' in event['request']['userAttributes']:
        if event['request']['userAttributes']['custom:domain'] == address[1]:
            event['response']['autoConfirmUser'] = True

    # Return to Amazon Cognito
    return event
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des

données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "request": {
    "userAttributes": {
      "email": "testuser@example.com",
      "custom:domain": "example.com"
    }
  },
  "response": {}
}
```

Exemple d'avant inscription : confirmation et vérification automatiques de tous les utilisateurs

Cet exemple confirme tous les utilisateurs et définit les attributs `email` et `phone_number` sur « verified » si l'attribut est présent. De même, si l'option d'attribution d'un alias est activée, des alias sont créés pour `phone_number` et `email` lorsque la vérification automatique est définie.

Note

S'il existe déjà un alias avec ce numéro de téléphone, l'alias est déplacé vers le nouvel utilisateur et l'attribut `phone_number` de l'utilisateur précédent est marqué comme non vérifié. Il en va de même pour les adresses e-mail. Pour éviter cela, vous pouvez utiliser [l'ListUsers API](#) des groupes d'utilisateurs pour voir s'il existe un utilisateur existant qui utilise déjà le numéro de téléphone ou l'adresse e-mail du nouvel utilisateur comme alias.

Node.js

```
const handler = async (event) => {
  // Confirm the user
  event.response.autoConfirmUser = true;
  // Set the email as verified if it is in the request
  if (event.request.userAttributes.hasOwnProperty("email")) {
    event.response.autoVerifyEmail = true;
  }
}
```



```
// Set the phone number as verified if it is in the request
if (event.request.userAttributes.hasOwnProperty("phone_number")) {
    event.response.autoVerifyPhone = true;
}

return event;
};

export { handler };
```

Python

```
def lambda_handler(event, context):
    # Confirm the user
    event['response']['autoConfirmUser'] = True

    # Set the email as verified if it is in the request
    if 'email' in event['request']['userAttributes']:
        event['response']['autoVerifyEmail'] = True

    # Set the phone number as verified if it is in the request
    if 'phone_number' in event['request']['userAttributes']:
        event['response']['autoVerifyPhone'] = True

    # Return to Amazon Cognito
    return event
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "request": {
    "userAttributes": {
      "email": "user@example.com",
      "phone_number": "+12065550100"
    }
  }
}
```

```
    }  
  },  
  "response": {}  
}
```

Exemple de pré-inscription : Refuser l'inscription si le nom d'utilisateur comporte moins de cinq caractères

Cet exemple vérifie la longueur du nom d'utilisateur dans une demande d'inscription. L'exemple renvoie une erreur si l'utilisateur a saisi un nom de moins de cinq caractères.

Node.js

```
exports.handler = (event, context, callback) => {  
  // Impose a condition that the minimum length of the username is 5 is imposed on  
  all user pools.  
  if (event.userName.length < 5) {  
    var error = new Error("Cannot register users with username less than the  
minimum length of 5");  
    // Return error to Amazon Cognito  
    callback(error, event);  
  }  
  // Return to Amazon Cognito  
  callback(null, event);  
};
```

Python

```
def lambda_handler(event, context):  
    if len(event['userName']) < 5:  
        raise Exception("Cannot register users with username less than the minimum  
length of 5")  
    # Return to Amazon Cognito  
    return event
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des

données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "userName": "rroe",
  "response": {}
}
```

Déclencheur Lambda après confirmation.

Amazon Cognito invoque ce déclencheur après qu'un utilisateur inscrit a confirmé son compte d'utilisateur. Dans votre fonction Lambda après confirmation, vous pouvez envoyer des messages personnalisés ou ajouter des demandes d'API personnalisées. Par exemple, vous pouvez interroger un système externe et renseigner des attributs supplémentaires pour l'utilisateur. Amazon Cognito invoque ce déclencheur uniquement pour les utilisateurs qui s'inscrivent dans votre groupe d'utilisateurs, et non pour les comptes d'utilisateur que vous créez avec vos informations d'identification d'administrateur.

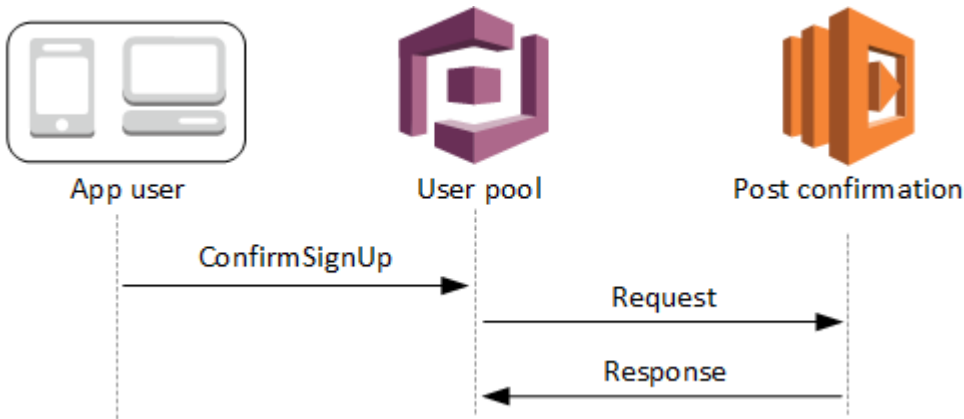
La demande contient les attributs actuels pour l'utilisateur confirmé.

Rubriques

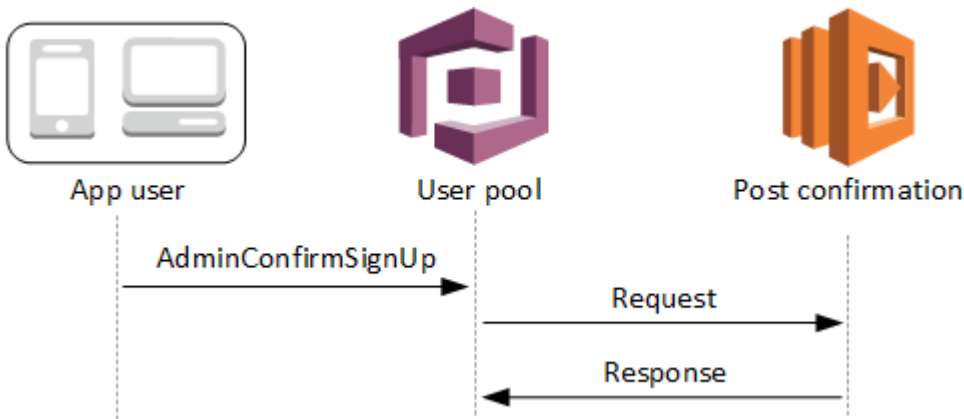
- [Flux Lambda après confirmation](#)
- [Paramètres du déclencheur Lambda après confirmation](#)
- [Didacticiels relatifs à la confirmation de l'utilisateur](#)
- [Exemple de stade après confirmation](#)

Flux Lambda après confirmation

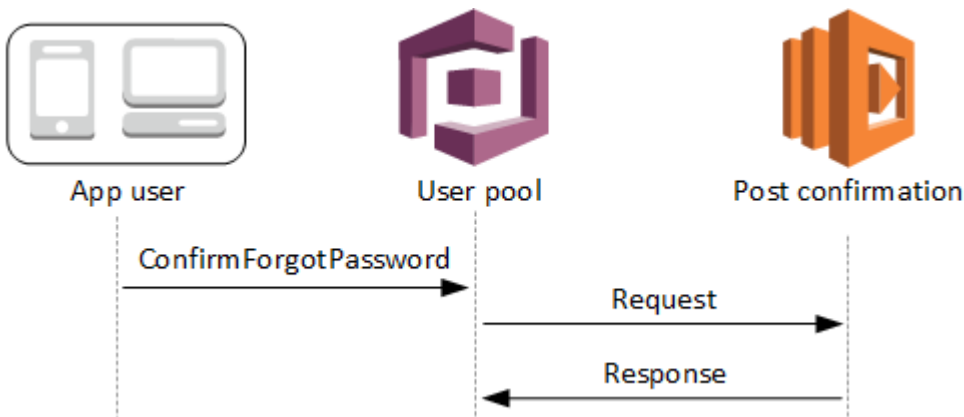
Flux d'inscription de confirmation client



Flux d'inscription de confirmation serveur



Flux de confirmation pour l'oubli de mot de passe



Paramètres du déclencheur Lambda après confirmation

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
    "userAttributes": {
      "string": "string",
      . . .
    },
    "clientMetadata": {
      "string": "string",
      . . .
    }
  },
  "response": {}
}
```

Paramètres de demande après confirmation

userAttributes

Une ou plusieurs paires clé-valeur représentant des attributs utilisateur.

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée pour la fonction Lambda que vous spécifiez pour le déclencheur Après confirmation. Vous pouvez transmettre ces données à votre fonction Lambda à l'aide du paramètre ClientMetadata dans les actions d'API suivantes : [AdminConfirmSignUp](#), [ConfirmForgotPassword](#), [ConfirmSignUp](#) et [SignUp](#).

Paramètres de réponse après confirmation

Aucune information de retour supplémentaire n'est prévue dans la réponse.

Didacticiels relatifs à la confirmation de l'utilisateur

La fonction Lambda Après confirmation est déclenchée juste après qu'Amazon Cognito confirme un nouvel utilisateur. Consultez ces didacticiels relatifs à la confirmation d'un utilisateur pour JavaScript, Android et iOS.

Plateforme	Didacticiel
Kit SDK d'identité pour JavaScript	Confirmez des utilisateurs avec JavaScript
Kit SDK d'identité pour Android	Confirmez des utilisateurs avec Android
Kit SDK d'identité pour iOS	Confirmez des utilisateurs avec iOS

Exemple de stade après confirmation

Cet exemple de fonction Lambda envoie un e-mail de confirmation à votre utilisateur à l'aide d'Amazon SES. Pour plus d'informations, consultez le [Manuel du développeur Amazon Simple Storage Service](#).

Node.js

```
// Import required AWS SDK clients and commands for Node.js. Note that this requires
// the `@aws-sdk/client-ses` module to be either bundled with this code or included
// as a Lambda layer.
import { SES, SendEmailCommand } from "@aws-sdk/client-ses";
const ses = new SES();

const handler = async (event) => {
  if (event.request.userAttributes.email) {
    await sendTheEmail(
      event.request.userAttributes.email,
      `Congratulations ${event.userName}, you have been confirmed.`
    );
  }
  return event;
};
```

```
const sendTheEmail = async (to, body) => {
  const eParams = {
    Destination: {
      ToAddresses: [to],
    },
    Message: {
      Body: {
        Text: {
          Data: body,
        },
      },
      Subject: {
        Data: "Cognito Identity Provider registration completed",
      },
    },
    // Replace source_email with your SES validated email address
    Source: "<source_email>",
  };
  try {
    await ses.send(new SendEmailCommand(eParams));
  } catch (err) {
    console.log(err);
  }
};

export { handler };
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "request": {
    "userAttributes": {
      "email": "user@example.com",
      "email_verified": true
    }
  }
}
```

```
    }  
  },  
  "response": {}  
}
```

Déclencheur Lambda avant authentification

Amazon Cognito appelle ce déclencheur quand un utilisateur tente de se connecter, ce qui vous permet d'effectuer une validation personnalisée qui réalise des actions préparatoires. Par exemple, vous pouvez refuser la demande d'authentification ou enregistrer les données de session au niveau d'un système externe.

Note

Ce déclencheur Lambda ne s'active pas lorsqu'un utilisateur n'existe pas ou qu'il possède déjà une session dans votre groupe d'utilisateurs. Si le paramètre `PreventUserExistenceErrors` d'un client d'application de groupe d'utilisateurs est défini sur `ENABLED`, le déclencheur Lambda s'active.

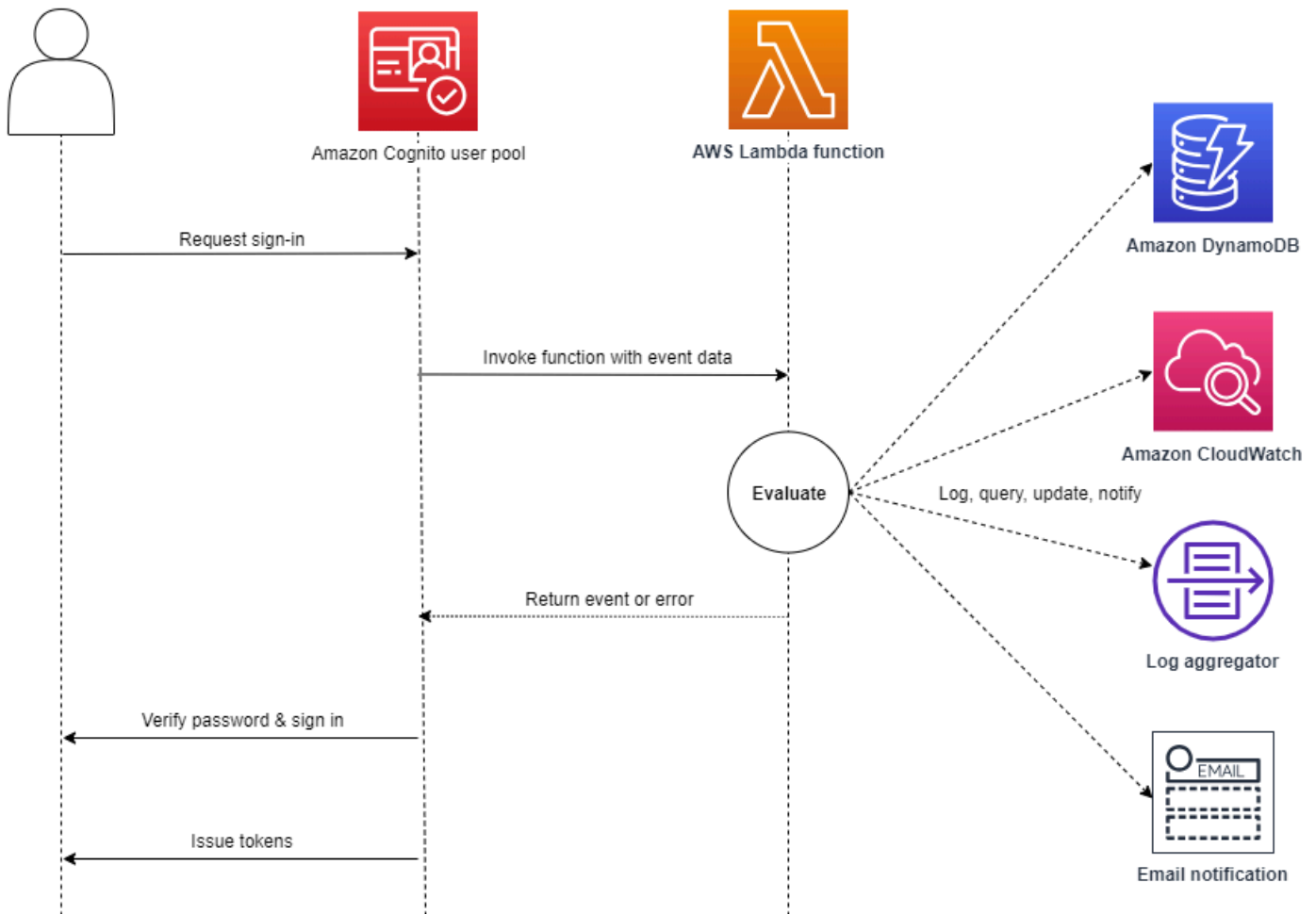
Rubriques

- [Présentation du flux d'authentification](#)
- [Paramètres du déclencheur Lambda avant l'authentification](#)
- [Exemple de stade avant l'authentification](#)

Présentation du flux d'authentification

Amazon Cognito pre authentication trigger

Evaluate and authorize user sign-in



La demande inclut les données de validation du client, provenant des valeurs `ClientMetadata` que votre application transmet aux opérations d'API `InitiateAuth` et `AdminInitiateAuth` du groupe d'utilisateurs.

Pour de plus amples informations, veuillez consulter [Flux d'authentification de groupe d'utilisateurs](#).

Paramètres du déclencheur Lambda avant l'authentification

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
    "userAttributes": {
      "string": "string",
      . . .
    },
    "validationData": {
      "string": "string",
      . . .
    },
    "userNotFound": boolean
  },
  "response": {}
}
```

Paramètres de demande avant l'authentification

userAttributes

Une ou plusieurs paires nom-valeur représentant les attributs utilisateur.

userNotFound

Quand vous définissez `PreventUserExistenceErrors` sur `ENABLED` pour votre client de groupe d'utilisateurs, Amazon Cognito renseigne ce booléen.

validationData

Une ou plusieurs paires clé-valeur contenant les données de validation dans la demande de connexion de l'utilisateur. Pour transmettre ces données à votre fonction Lambda, utilisez le paramètre `ClientMetadata` dans les actions d'API [InitiateAuth](#) et [AdminInitiateAuth](#).

Paramètres de réponse avant l'authentification

Amazon Cognito n'attend aucune information de retour supplémentaire dans la réponse. Votre fonction peut renvoyer une erreur pour rejeter la tentative de connexion ou utiliser des opérations d'API pour interroger et modifier vos ressources.

Exemple de stade avant l'authentification

Cet exemple de fonction empêche les utilisateurs de se connecter à votre groupe d'utilisateurs avec un client d'application spécifique. Puisque la fonction Lambda avant l'authentification n'appelle pas lorsque votre utilisateur possède une session existante, elle empêche uniquement les nouvelles sessions avec l'ID client de l'application que vous souhaitez bloquer.

Node.js

```
const handler = async (event) => {
  if (
    event.callerContext.clientId === "user-pool-app-client-id-to-be-blocked"
  ) {
    throw new Error("Cannot authenticate users from this user pool app client");
  }

  return event;
};

export { handler };
```

Python

```
def lambda_handler(event, context):
    if event['callerContext']['clientId'] == "<user pool app client id to be
    blocked>":
        raise Exception("Cannot authenticate users from this user pool app client")

    # Return to Amazon Cognito
    return event
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des

données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "callerContext": {
    "clientId": "<user pool app client id to be blocked>"
  },
  "response": {}
}
```

Déclencheur Lambda après l'authentification

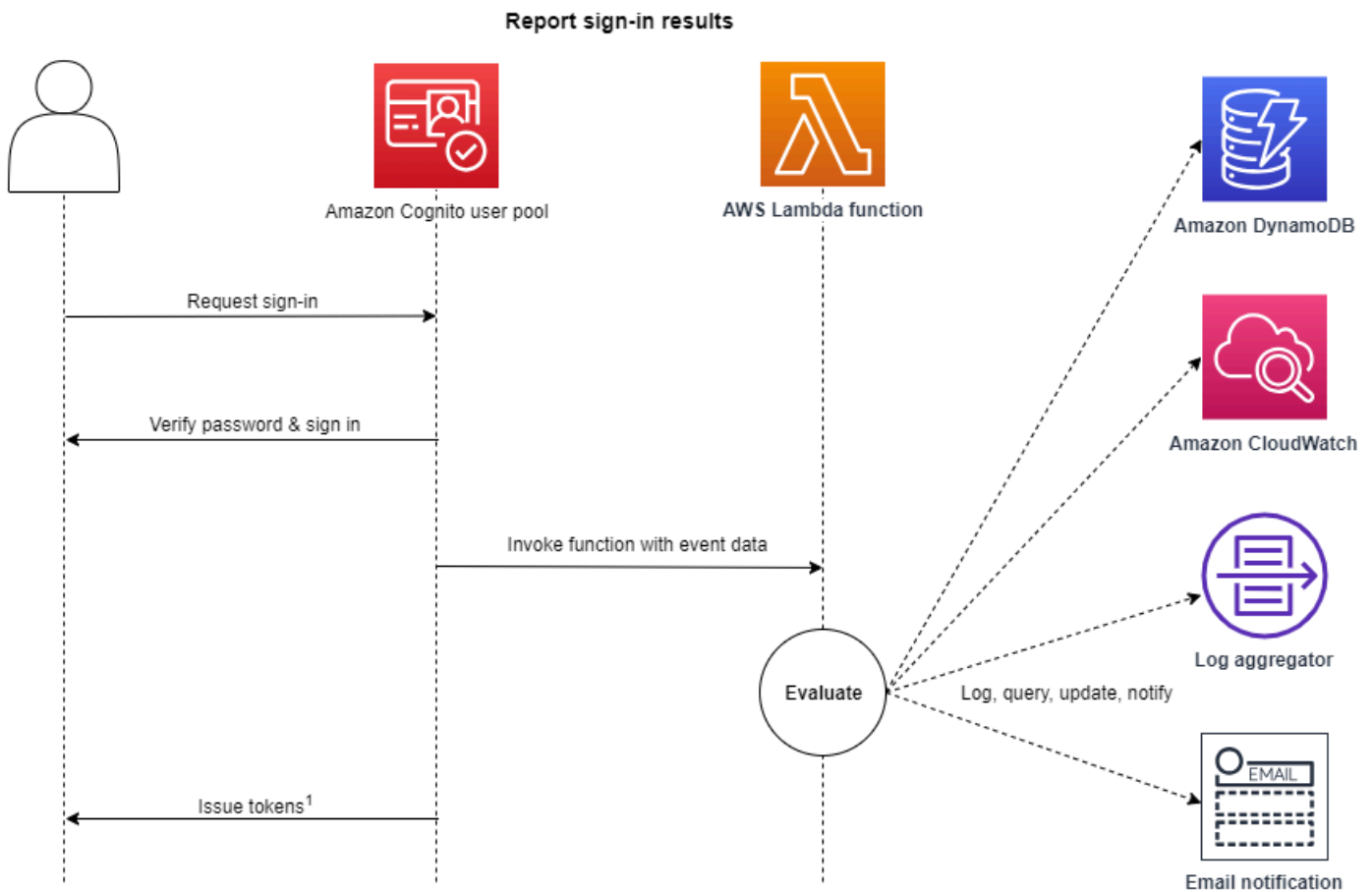
Comme Amazon Cognito appelle ce déclencheur après la connexion d'un utilisateur, vous pouvez ajouter une logique personnalisée après l'authentification de l'utilisateur par Amazon Cognito.

Rubriques

- [Présentation du flux d'authentification](#)
- [Paramètres du déclencheur Lambda après l'authentification](#)
- [Didacticiels relatifs à l'authentification](#)
- [Exemple de stade après l'authentification](#)

Présentation du flux d'authentification

Amazon Cognito post authentication trigger



Pour de plus amples informations, veuillez consulter [Flux d'authentification de groupe d'utilisateurs](#).

Paramètres du déclencheur Lambda après l'authentification

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
```

```
    "userAttributes": {
      "string": "string",
      . . .
    },
    "newDeviceUsed": boolean,
    "clientMetadata": {
      "string": "string",
      . . .
    }
  },
  "response": {}
}
```

Paramètres de demande après l'authentification

newDeviceUsed

Cet indicateur permet de savoir si l'utilisateur s'est connecté sur un nouvel appareil. Amazon Cognito définit cet indicateur seulement si la valeur des appareils mémorisés du groupe d'utilisateurs est `Always` ou `User Opt-In`.

userAttributes

Une ou plusieurs paires nom-valeur représentant des attributs utilisateur.

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée à la fonction Lambda que vous spécifiez pour le déclencheur Après l'authentification. Pour transmettre ces données à votre fonction Lambda, vous pouvez utiliser le paramètre `ClientMetadata` dans les actions d'API [AdminRespondToAuthChallenge](#) et [RespondToAuthChallenge](#). Amazon Cognito n'inclut pas les données provenant du paramètre `ClientMetadata` dans les opérations d'API [AdminInitiateAuth](#) et [InitiateAuth](#) dans la demande transmise à la fonction après authentification.

Paramètres de réponse après l'authentification

Amazon Cognito n'attend aucune information de retour supplémentaire dans la réponse. Votre fonction peut utiliser les opérations d'API pour interroger et modifier vos ressources, ou enregistrer des métadonnées d'événements dans un système externe.

Didacticiels relatifs à l'authentification

Immédiatement après qu'Amazon Cognito connecte un utilisateur, il active la fonction Lambda après authentification. Consultez ces didacticiels relatifs à la connexion pour JavaScript, Android et iOS.

Plateforme	Didacticiel
Kit SDK d'identité pour JavaScript	Connectez des utilisateurs avec JavaScript
Kit SDK d'identité pour Android	Connectez des utilisateurs avec Android
Kit SDK d'identité pour iOS	Connectez des utilisateurs avec iOS

Exemple de stade après l'authentification

Cet exemple de fonction Lambda Après l'authentification envoie les données d'une connexion à CloudWatch Logs réussie.

Node.js

```
const handler = async (event) => {
  // Send post authentication data to Amazon CloudWatch logs
  console.log("Authentication successful");
  console.log("Trigger function =", event.triggerSource);
  console.log("User pool = ", event.userPoolId);
  console.log("App client ID = ", event.callerContext.clientId);
  console.log("User ID = ", event.userName);

  return event;
};

export { handler }
```

Python

```
import os
```

```
def lambda_handler(event, context):

    # Send post authentication data to Cloudwatch logs
    print ("Authentication successful")
    print ("Trigger function =", event['triggerSource'])
    print ("User pool = ", event['userPoolId'])
    print ("App client ID = ", event['callerContext']['clientId'])
    print ("User ID = ", event['userName'])

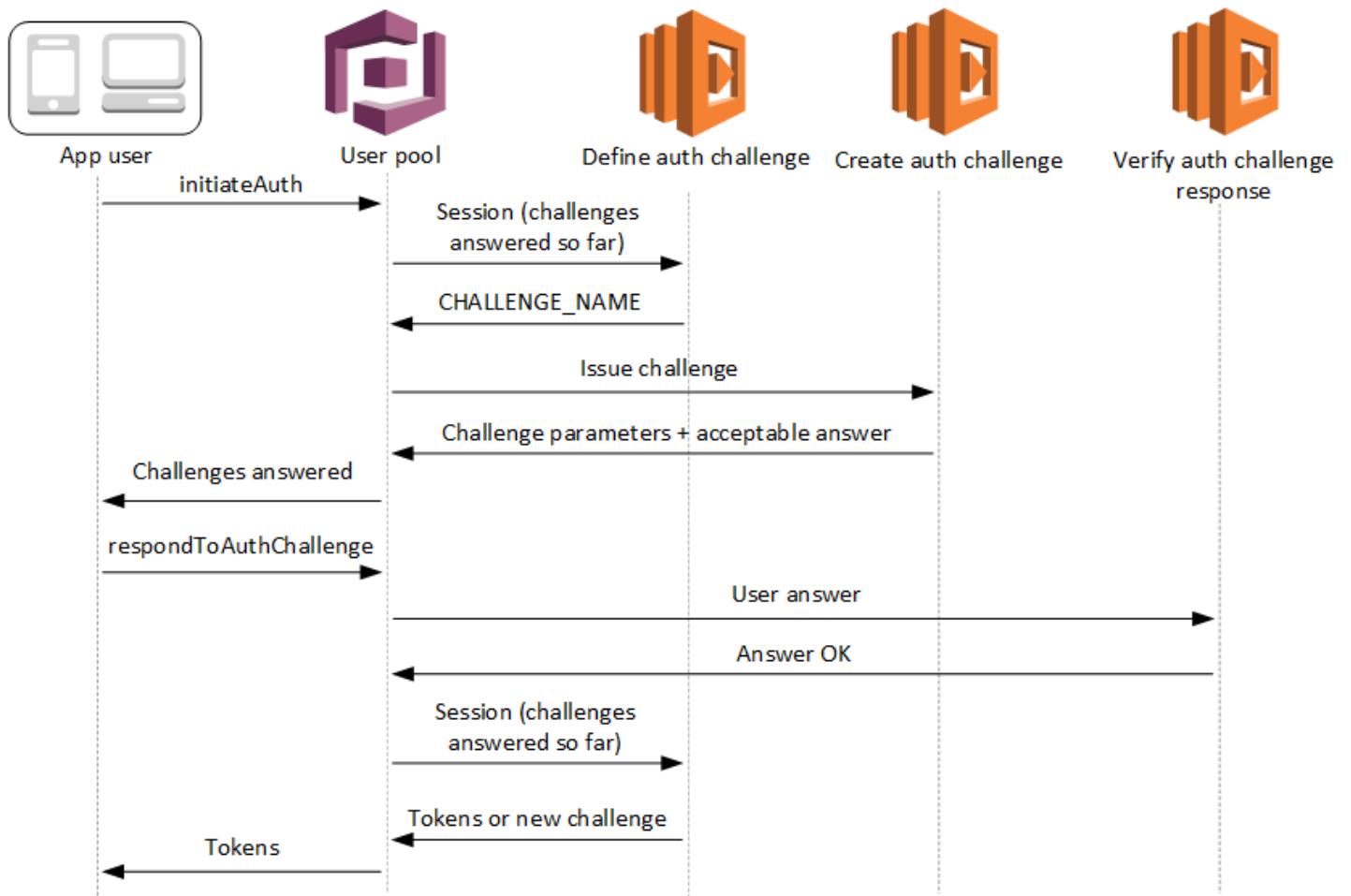
    # Return to Amazon Cognito
    return event
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "triggerSource": "testTrigger",
  "userPoolId": "testPool",
  "userName": "testName",
  "callerContext": {
    "clientId": "12345"
  },
  "response": {}
}
```


Déclencheurs Lambda création d'une stimulation d'authentification personnalisée



Ces déclencheurs Lambda émettent et vérifient leurs propres stimulations dans le cadre du [flux d'authentification personnalisé](#) d'un groupe d'utilisateurs.

Define auth challenge

Amazon Cognito appelle ce déclencheur pour initier le flux d'authentification personnalisé.

Create auth challenge

Amazon Cognito appelle ce déclencheur après la commande Définition d'une stimulation d'authentification pour créer une stimulation personnalisée.

Verify auth challenge response

Amazon Cognito appelle ce déclencheur pour vérifier si la réponse de l'utilisateur final à une stimulation personnalisée est valide ou non.

Vous pouvez intégrer de nouveaux types de stimulations avec ces déclencheurs Lambda de stimulation. Par exemple, ces types de stimulations peuvent inclure des CAPTCHAs ou des questions dynamiques.

Vous pouvez généraliser l'authentification en deux étapes courantes avec les méthodes d'API `InitiateAuth` et `RespondToAuthChallenge` du groupe d'utilisateurs.

Dans ce flux, un utilisateur s'authentifie en répondant à des défis successifs jusqu'à ce que l'authentification échoue ou que l'utilisateur reçoive les jetons. Ces deux appels d'API peuvent être répétés de façon à inclure différents défis.

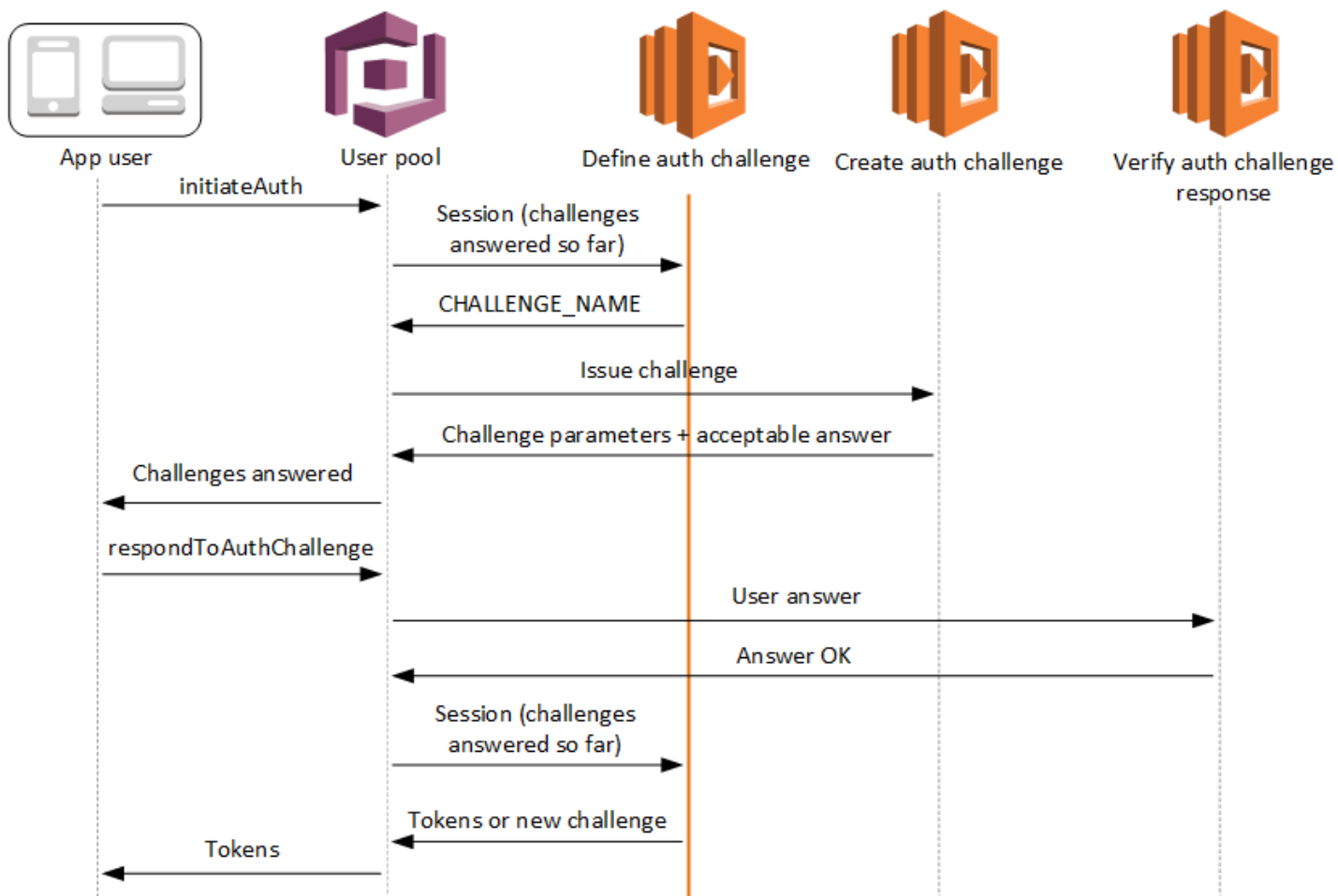
Note

L'interface utilisateur hébergée d'Amazon Cognito ne prend pas en charge l'authentification personnalisée avec [les déclencheurs Lambda de stimulation d'authentification personnalisée](#).

Rubriques

- [Déclencheur Lambda Définition d'une stimulation d'authentification](#)
- [Déclencheur Lambda création d'une stimulation d'authentification](#)
- [Déclencheur Lambda Vérification de la réponse à la stimulation d'authentification](#)

Déclencheur Lambda Définition d'une stimulation d'authentification



Define auth challenge

Amazon Cognito appelle ce déclencheur pour initier le [flux d'authentification personnalisé](#).

La demande de ce déclencheur Lambda contient `session`. Le paramètre `session` est un tableau contenant toutes les demandes de vérification présentées à l'utilisateur dans le processus d'authentification en cours. La demande inclut également le résultat correspondant. Le tableau `session` stocke les détails des demandes de vérification (`ChallengeResult`) dans l'ordre chronologique. La demande de vérification `session[0]` représente la première demande de vérification que l'utilisateur reçoit.

Vous pouvez demander à Amazon Cognito de vérifier les mots de passe utilisateur avant d'émettre vos stimulations personnalisées. Tous les déclencheurs Lambda associés à la catégorie Authentification des [quotas de taux de demande](#) s'exécutent lorsque vous effectuez une authentification SRP dans un flux de stimulation personnalisé. Voici un aperçu du processus :

1. Votre application initie la connexion en appelant `InitiateAuth` ou `AdminInitiateAuth` avec le mappage `AuthParameters`. Les paramètres doivent inclure `CHALLENGE_NAME: SRP_A`, et les valeurs pour `SRP_A` et `USERNAME`.
2. Amazon Cognito appelle votre déclencheur Lambda Définition de la question de sécurité d'authentification avec une session initiale qui contient `challengeName: SRP_A` et `challengeResult: true`.
3. Après réception de ces entrées, votre fonction Lambda répond avec `challengeName: PASSWORD_VERIFIER`, `issueTokens: false`, `failAuthentication: false`.
4. Si la vérification du mot de passe réussit, Amazon Cognito appelle votre fonction Lambda à nouveau avec une nouvelle session contenant `challengeName: PASSWORD_VERIFIER` et `challengeResult: true`.
5. Pour initier vos demandes de vérification personnalisées, votre fonction Lambda répond avec `challengeName: CUSTOM_CHALLENGE`, `issueTokens: false` et `failAuthentication: false`. Si vous ne souhaitez pas démarrer votre flux d'authentification personnalisé avec la vérification du mot de passe, vous pouvez initier une connexion avec le mappage `AuthParameters` en incluant `CHALLENGE_NAME: CUSTOM_CHALLENGE`.
6. La boucle de stimulation se répète jusqu'à ce que toutes les réponses soient apportées à la stimulation.

Rubriques

- [Paramètres du déclencheur Lambda Définition d'une stimulation d'authentification](#)
- [Exemple de définition de la stimulation d'authentification](#)

Paramètres du déclencheur Lambda Définition d'une stimulation d'authentification

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
    "userAttributes": {
      "string": "string",
```

```

    . . .
  },
  "session": [
    ChallengeResult,
    . . .
  ],
  "clientMetadata": {
    "string": "string",
    . . .
  },
  "userNotFound": boolean
},
"response": {
  "challengeName": "string",
  "issueTokens": boolean,
  "failAuthentication": boolean
}
}

```

Paramètres de demande de définition de la stimulation d'authentification

Quand Amazon Cognito appelle votre fonction Lambda, Amazon Cognito fournit les paramètres suivants :

userAttributes

Une ou plusieurs paires nom-valeur représentant les attributs utilisateur.

userNotFound

Booléen renseigné par Amazon Cognito quand `PreventUserExistenceErrors` est défini sur `ENABLED` pour votre client de groupe d'utilisateurs. Une valeur `true` signifie que l'ID utilisateur (nom d'utilisateur, adresse e-mail, etc.) ne correspond à aucun utilisateur existant. Quand `PreventUserExistenceErrors` a pour valeur `ENABLED`, le service n'informe pas l'application des utilisateurs inexistant. Nous vous recommandons de faire en sorte que vos fonctions Lambda conservent la même expérience utilisateur et tiennent compte de la latence. De cette façon, l'appelant ne peut pas détecter un comportement différent quand l'utilisateur existe ou n'existe pas.

séance

Tableau d'éléments `ChallengeResult`. Chacun contient les éléments suivants :

challengeName

Un des types de demande de vérification suivants : `CUSTOM_CHALLENGE`, `SRP_A`, `PASSWORD_VERIFIER`, `SMS_MFA`, `DEVICE_SRP_AUTH`, `DEVICE_PASSWORD_VERIFIER` ou `ADMIN_NO_SRP_AUTH`.

Lorsque votre fonction de définition de la stimulation d'authentification lance une stimulation `PASSWORD_VERIFIER` à un utilisateur qui a configuré l'authentification multifactorielle, Amazon Cognito lance ensuite une stimulation `SMS_MFA`. Dans votre fonction, incluez la gestion des événements d'entrée provenant des stimulations `SMS_MFA`. Vous n'avez pas besoin d'invoquer la stimulation `SMS_MFA` à partir de votre fonction de définition de la stimulation d'authentification.

Important

Quand votre fonction détermine si un utilisateur s'est authentifié correctement et doit recevoir des jetons, vérifiez toujours `challengeName` dans votre fonction de définition de la stimulation d'authentification et s'il correspond à la valeur attendue.

challengeResult

Défini sur `true` si l'utilisateur a répondu à la demande de vérification avec succès, ou sur `false` dans le cas contraire.

challengeMetadata

Votre nom pour la demande de vérification personnalisée. Utilisé uniquement si `challengeName` est `CUSTOM_CHALLENGE`.

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée à la fonction Lambda que vous spécifiez pour le déclencheur Définition d'une stimulation d'authentification. Pour transmettre ces données à votre fonction Lambda, vous pouvez utiliser le paramètre `ClientMetadata` dans les opérations d'API [AdminRespondToAuthChallenge](#) et [RespondToAuthChallenge](#). La demande qui appelle la fonction de définition d'une demande de vérification d'authentification n'inclut pas les données transmises dans le paramètre `ClientMetadata` dans les opérations d'API [AdminInitiateAuth](#) et [InitiateAuth](#).

Paramètres de réponse de définition de la stimulation d'authentification

Dans la réponse, vous pouvez renvoyer l'étape suivante du processus d'authentification.

challengeName

Chaîne contenant le nom de la prochaine demande de vérification. Si vous souhaitez présenter une nouvelle demande de vérification pour votre utilisateur, spécifiez ici son nom.

issueTokens

Si vous déterminez que l'utilisateur a suffisamment répondu aux demandes de vérification d'authentification, définissez la valeur `true`. Si l'utilisateur n'a pas suffisamment répondu aux demandes de vérification, définissez la valeur `false`.

failAuthentication

Si vous souhaitez mettre fin au processus d'authentification actuel, définissez la valeur `true`. Pour poursuivre le processus d'authentification actuel, définissez la valeur `false`.

Exemple de définition de la stimulation d'authentification

Cet exemple définit une série de demandes de vérification d'authentification et émet des jetons seulement si l'utilisateur répond avec succès à toutes les demandes de vérification.

Node.js

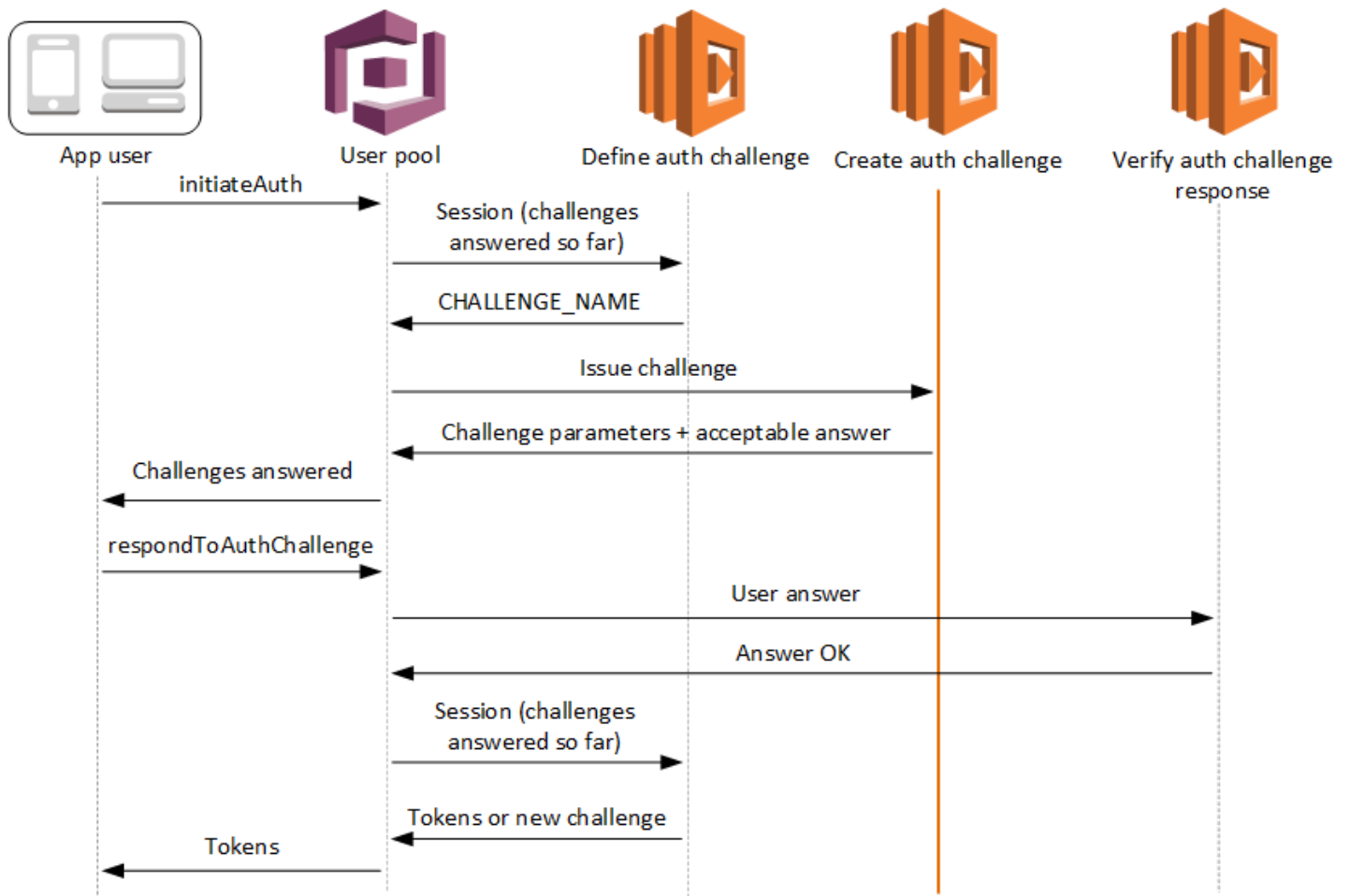
```
const handler = async (event) => {
  if (
    event.request.session.length == 1 &&
    event.request.session[0].challengeName == "SRP_A"
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "PASSWORD_VERIFIER";
  } else if (
    event.request.session.length == 2 &&
    event.request.session[1].challengeName == "PASSWORD_VERIFIER" &&
    event.request.session[1].challengeResult == true
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "CUSTOM_CHALLENGE";
  } else if (
```

```
    event.request.session.length == 3 &&
    event.request.session[2].challengeName == "CUSTOM_CHALLENGE" &&
    event.request.session[2].challengeResult == true
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "CUSTOM_CHALLENGE";
  } else if (
    event.request.session.length == 4 &&
    event.request.session[3].challengeName == "CUSTOM_CHALLENGE" &&
    event.request.session[3].challengeResult == true
  ) {
    event.response.issueTokens = true;
    event.response.failAuthentication = false;
  } else {
    event.response.issueTokens = false;
    event.response.failAuthentication = true;
  }

  return event;
};

export { handler }
```


Déclencheur Lambda création d'une stimulation d'authentification



Create auth challenge

Amazon Cognito appelle ce déclencheur après la Définition d'une stimulation d'authentification si une stimulation personnalisée a été spécifiée dans le déclencheur Définition d'une stimulation d'authentification. Il crée un [flux d'authentification personnalisé](#).

Ce déclencheur Lambda est appelé pour créer une stimulation à présenter à l'utilisateur. La demande de ce déclencheur Lambda inclut les paramètres `challengeName` et `session`. `challengeName` est une chaîne et constitue le nom de la prochaine demande de vérification pour l'utilisateur. La valeur de cet attribut est définie dans le déclencheur Lambda Définition d'une stimulation d'authentification.

La boucle de stimulation se répète jusqu'à ce que toutes les réponses soient apportées à la stimulation.

Rubriques

- [Paramètres du déclencheur Lambda création d'une stimulation d'authentification](#)
- [Exemple de création d'une stimulation d'authentification](#)

Paramètres du déclencheur Lambda création d'une stimulation d'authentification

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
    "userAttributes": {
      "string": "string",
      . . .
    },
    "challengeName": "string",
    "session": [
      ChallengeResult,
      . . .
    ],
    "clientMetadata": {
      "string": "string",
      . . .
    },
    "userNotFound": boolean
  },
  "response": {
    "publicChallengeParameters": {
      "string": "string",
      . . .
    },
    "privateChallengeParameters": {
      "string": "string",
      . . .
    },
    "challengeMetadata": "string"
  }
}
```

Paramètres de demande de création d'une stimulation d'authentification

userAttributes

Une ou plusieurs paires nom-valeur représentant des attributs utilisateur.

userNotFound

Cette valeur booléenne est utilisée lorsque `PreventUserExistenceErrors` est défini sur `ENABLED` pour votre client de groupe d'utilisateurs.

challengeName

Nom de la nouvelle demande de vérification.

séance

L'élément `session` est un tableau d'éléments `ChallengeResult`, chacun d'entre eux contenant les éléments suivants :

challengeName

Type de demande de vérification. Soit `"CUSTOM_CHALLENGE"`, `"PASSWORD_VERIFIER"`, `"SMS_MFA"`, `"DEVICE_SRP_AUTH"`, `"DEVICE_PASSWORD_VERIFIER"` ou `"ADMIN_NO_SRP_AUTH"`.

challengeResult

Défini sur `true` si l'utilisateur a répondu à la demande de vérification avec succès, ou sur `false` dans le cas contraire.

challengeMetadata

Votre nom pour la demande de vérification personnalisée. Utilisé uniquement si `challengeName` est `"CUSTOM_CHALLENGE"`.

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée à la fonction Lambda que vous spécifiez pour le déclencheur Création d'une stimulation d'authentification. Vous pouvez utiliser le paramètre `ClientMetadata` dans les actions d'API [AdminRespondToAuthChallenge](#) et [RespondToAuthChallenge](#) pour transmettre ces données à votre fonction Lambda. La demande qui appelle la fonction de définition de la demande de vérification d'authentification n'inclut pas les données transmises dans le paramètre `ClientMetadata` dans les opérations d'API [AdminInitiateAuth](#) et [InitiateAuth](#).

Paramètres de réponse de création d'une stimulation d'authentification

publicChallengeParameters

Une ou plusieurs paires clé-valeur que l'application client doit utiliser dans la demande de vérification à présenter à l'utilisateur. Ce paramètre doit contenir toutes les informations nécessaires pour présenter avec précision la demande de vérification à l'utilisateur.

privateChallengeParameters

Ce paramètre est utilisé uniquement par le déclencheur Lambda Vérification de la réponse à la stimulation d'authentification. Ce paramètre doit contenir toutes les informations qui sont nécessaire pour valider la réponse de l'utilisateur à la demande de vérification. En d'autres termes, le paramètre `publicChallengeParameters` contient la question qui est posée à l'utilisateur et `privateChallengeParameters` contient les réponses valides pour la question.

challengeMetadata

Votre nom pour la demande de vérification personnalisée, s'il s'agit d'une demande de vérification personnalisée.

Exemple de création d'une stimulation d'authentification

Une image CAPTCHA est créée en tant que demande de vérification pour l'utilisateur. L'URL de l'image CAPTCHA est ajoutée aux paramètres de demande de vérification publics sous la forme « `captchaUrl` », et la réponse attendue est ajoutée aux paramètres de demande de vérification privés.

Node.js

```
const handler = async (event) => {
  if (event.request.challengeName !== "CUSTOM_CHALLENGE") {
    return event;
  }

  if (event.request.session.length === 2) {
    event.response.publicChallengeParameters = {};
    event.response.privateChallengeParameters = {};
    event.response.publicChallengeParameters.captchaUrl = "url/123.jpg";
    event.response.privateChallengeParameters.answer = "5";
  }

  if (event.request.session.length === 3) {
```

```

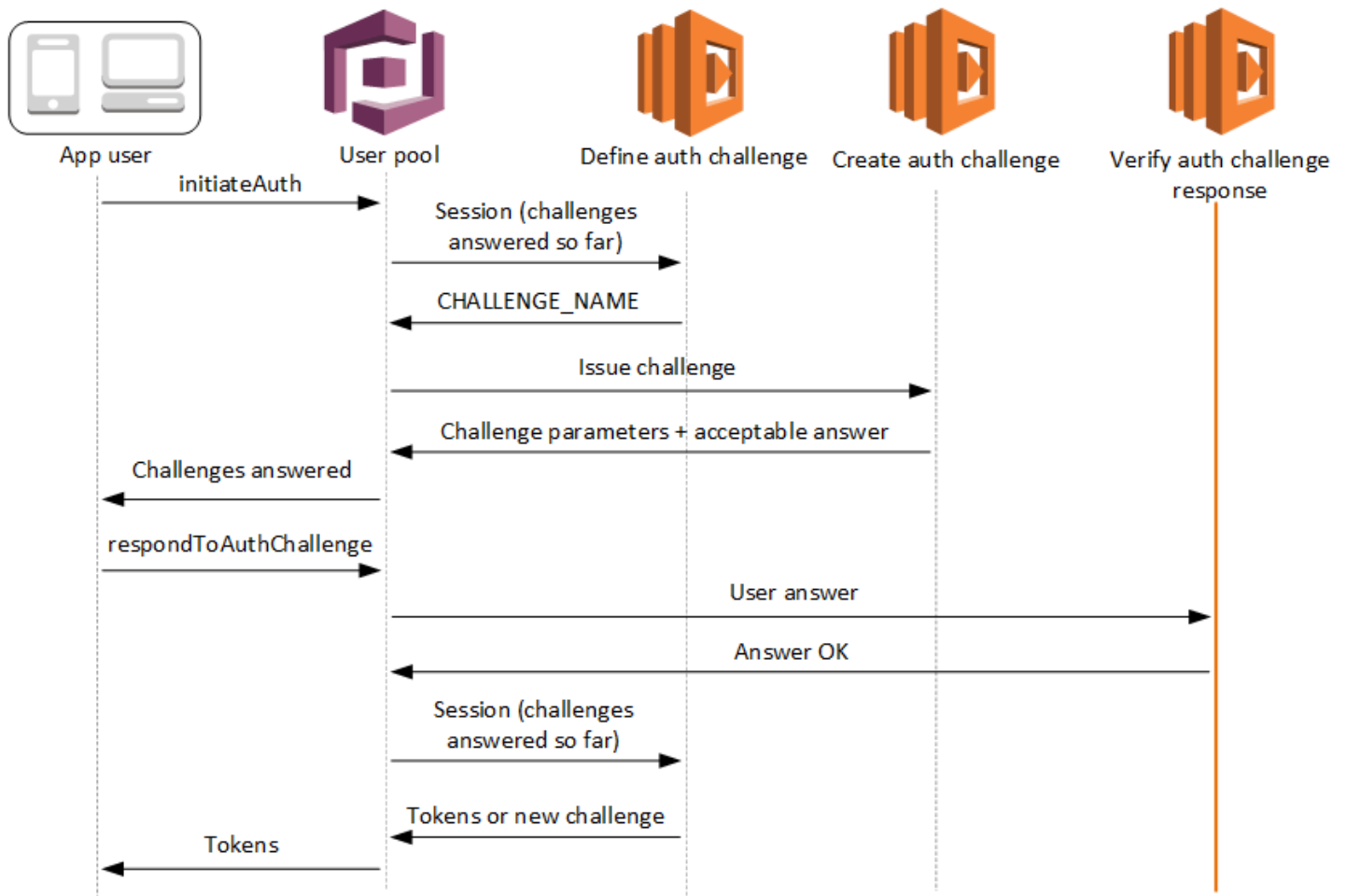
event.response.publicChallengeParameters = {};
event.response.privateChallengeParameters = {};
event.response.publicChallengeParameters.securityQuestion =
    "Who is your favorite team mascot?";
event.response.privateChallengeParameters.answer = "Peccy";
}

return event;
};

export { handler }

```

Déclencheur Lambda Vérification de la réponse à la stimulation d'authentification



Verify auth challenge response

Amazon Cognito appelle ce déclencheur pour vérifier si la réponse de l'utilisateur à une demande de vérification d'authentification personnalisée est valide ou non. Il fait partie du [flux d'authentification personnalisé](#) d'un groupe d'utilisateurs.

La demande pour ce déclencheur contient les paramètres `privateChallengeParameters` et `challengeAnswer`. Le déclencheur Lambda Création d'une demande de vérification d'authentification renvoie les valeurs de `privateChallengeParameters` et contient la réponse attendue de l'utilisateur. Le paramètre `challengeAnswer` contient la réponse de l'utilisateur pour la demande de vérification.

La réponse contient l'attribut `answerCorrect`. Si l'utilisateur répond à la demande de vérification, Amazon Cognito définit la valeur de l'attribut sur `true`. Si l'utilisateur ne répond pas correctement à la demande de vérification, Amazon Cognito définit la valeur sur `false`.

La boucle de demande de vérification se répète jusqu'à ce que les utilisateurs aient répondu à toutes les demandes de vérification.

Rubriques

- [Paramètres du déclencheur Lambda Vérification de la réponse à la stimulation d'authentification](#)
- [Exemple de vérification de la réponse à la stimulation d'authentification](#)

Paramètres du déclencheur Lambda Vérification de la réponse à la stimulation d'authentification

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
    "userAttributes": {
      "string": "string",
      . . .
    },
    "privateChallengeParameters": {
      "string": "string",
```

```
    . . .
  },
  "challengeAnswer": "string",
  "clientMetadata": {
    "string": "string",
    . . .
  },
  "userNotFound": boolean
},
"response": {
  "answerCorrect": boolean
}
}
```

Paramètres de demande de vérification de la réponse à la stimulation d'authentification

userAttributes

Ce paramètre contient une ou plusieurs paires nom-valeur représentant les attributs utilisateur.

userNotFound

Quand Amazon Cognito définit `PreventUserExistenceErrors` sur `ENABLED` pour votre client de groupe d'utilisateurs, Amazon Cognito renseigne ce booléen.

privateChallengeParameters

Ce paramètre provient du déclencheur Création d'une demande de vérification d'authentification. Pour déterminer si l'utilisateur a réussi un défi, Amazon Cognito compare ces paramètres au paramètre `challengeAnswer` d'un utilisateur.

Ce paramètre contient toutes les informations qui sont nécessaires pour valider la réponse de l'utilisateur à la demande de vérification. Ces informations incluent la question posée par Amazon Cognito à l'utilisateur (`publicChallengeParameters`), et les réponses valides à la question (`privateChallengeParameters`). Seul le déclencheur Lambda Vérification de la réponse à la demande de vérification d'authentification utilise ce paramètre.

challengeAnswer

La valeur de ce paramètre est la réponse de l'utilisateur à la demande de vérification.

clientMetadata

Ce paramètre contient une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée à la fonction Lambda pour le déclencheur Vérification de la réponse à la demande de vérification d'authentification. Pour transmettre ces données à votre fonction Lambda, utilisez le paramètre ClientMetadata dans les opérations d'API [AdminRespondToAuthChallenge](#) et [RespondToAuthChallenge](#). Amazon Cognito n'inclut pas les données provenant du paramètre ClientMetadata dans les opérations d'API [AdminInitiateAuth](#) et [InitiateAuth](#) dans la demande transmise à la fonction de vérification de la réponse à la demande de vérification d'authentification.

Paramètres de réponse de vérification de la réponse à la stimulation d'authentification

answerCorrect

Si l'utilisateur répond à la demande de vérification, Amazon Cognito définit ce paramètre sur `true`. Si l'utilisateur ne répond pas correctement à la demande de vérification, Amazon Cognito définit le paramètre sur `false`.

Exemple de vérification de la réponse à la stimulation d'authentification

Dans cet exemple, la fonction Lambda vérifie si la réponse de l'utilisateur à une stimulation correspond à la réponse attendue. Si la réponse de l'utilisateur correspond à la réponse prévue, Amazon Cognito définit le paramètre `answerCorrect` sur `true`.

Node.js

```
const handler = async (event) => {
  if (
    event.request.privateChallengeParameters.answer ==
    event.request.challengeAnswer
  ) {
    event.response.answerCorrect = true;
  } else {
    event.response.answerCorrect = false;
  }

  return event;
};
```



```
export { handler };
```

Déclencheur Lambda avant génération de jeton

Comme Amazon Cognito appelle ce déclencheur avant la génération de jeton, vous pouvez personnaliser les revendications dans les jetons du groupe d'utilisateurs. Les fonctionnalités de base de la première version ou l'événement déclencheur avant la génération du jeton V1_0 vous permettent de personnaliser le jeton d'identité (ID). Dans les groupes d'utilisateurs dotés de [fonctionnalités de sécurité avancées](#) actives, vous pouvez générer la version 2 ou l'événement déclencheur V2_0 en personnalisant le jeton d'accès.

Amazon Cognito envoie un événement V1_0 en tant que demande à votre fonction avec les données qu'il inscrirait dans le jeton d'identification. Un événement V2_0 est une demande unique contenant les données qu'Amazon Cognito inscrirait à la fois dans les jetons d'identité et d'accès. Pour personnaliser les deux jetons, vous devez mettre à jour votre fonction afin d'utiliser la version du déclencheur la plus récente et envoyer les données des deux jetons dans la même réponse.

Ce déclencheur Lambda permet d'ajouter, de supprimer et de modifier certaines demandes relatives aux jetons d'identité et d'accès avant qu'Amazon Cognito ne les envoie à votre application. Pour utiliser cette fonction, associez une fonction Lambda à partir de la console des groupes d'utilisateurs Amazon Cognito ou mettez à jour votre groupe d'utilisateurs LambdaConfig via l'AWS Command Line Interface (AWS CLI).

Versions de l'événement

Votre groupe d'utilisateurs peut fournir différentes versions d'un événement déclencheur antérieur à la génération du jeton à votre fonction Lambda. Un V1_0 déclencheur fournit les paramètres de modification des jetons d'identification. Un V2_0 déclencheur fournit les paramètres suivants.

1. Les fonctions d'un V1_0 déclencheur.
2. Possibilité de personnaliser les jetons d'accès.
3. Possibilité de transmettre des types de données complexes aux valeurs de réclamation des identifiants et des jetons d'accès :
 - Chaîne
 - Nombre
 - Booléen

- Tableau de chaînes, de nombres, de booléens ou d'une combinaison de ces éléments
- JSON

Note

Dans le jeton d'identification, vous pouvez renseigner des objets complexes avec les valeurs des revendications, à l'exception de `phone_number_verified`, `email_verifiedupdated_at`, et `address`.

Les groupes d'utilisateurs fournissent V1_0 des événements par défaut. Pour configurer votre groupe d'utilisateurs afin d'envoyer un V2_0 événement, choisissez une version d'événement déclencheur des fonctionnalités de base et personnalisation du jeton d'accès lorsque vous configurez votre déclencheur dans la console Amazon Cognito. Vous pouvez également définir la valeur de `LambdaVersion` dans les [LambdaConfig](#) paramètres d'une requête [UpdateUserPool](#) ou d'une demande d'[CreateUserPool](#) API. Des frais supplémentaires s'appliquent à la personnalisation des jetons d'accès V2_0 lors d'événements. Pour plus d'informations, consultez [Tarification d'Amazon Cognito](#).

Demandes et portées exclues

Amazon Cognito limite les demandes et les portées que vous pouvez ajouter, modifier ou supprimer dans les jetons d'accès et d'identité. Si votre fonction Lambda tente de définir une valeur pour l'une de ces demandes, Amazon Cognito émet un jeton contenant la valeur initiale de la réclamation, si elle figurait dans la demande.

Demandes partagées

- `acr`
- `amr`
- `at_hash`
- `auth_time`
- `azp`
- `exp`
- `iat`

- `iss`
- `jti`
- `nbf`
- `nonce`
- `origin_jti`
- `sub`
- `token_use`

Demandes de jetons d'identification

- `identities`
- `aud`
- `cognito:username`

Demandes relatives aux jetons d'accès

- `username`
- `client_id`
- `scope`

Note

Vous pouvez modifier les portées d'un jeton d'accès avec les valeurs de réponse `scopesToAdd` et `scopesToSuppress`, mais vous ne pouvez pas modifier directement la demande `scope`. Vous ne pouvez pas ajouter des portées commençant par `aws.cognito`, y compris la portée `aws.cognito.signin.user.admin` réservée aux groupes d'utilisateurs.

- `device_key`
- `event_id`
- `version`

Vous ne pouvez pas ajouter ou annuler les demandes comportant les préfixes suivants, mais vous pouvez les supprimer ou les empêcher d'apparaître dans le jeton.

- `dev`:
- `cognito`:

Vous pouvez ajouter une demande aux jetons d'accès, mais sa valeur doit correspondre à l'ID client de l'application de la session en cours. Vous pouvez obtenir l'ID du client dans l'événement de demande à partir de `event.callerContext.clientId`.

Personnalisation du jeton d'identité

Avec le déclencheur Lambda de la prégénération du jeton, vous pouvez personnaliser le contenu d'un jeton d'identité (ID) à partir de votre groupe d'utilisateurs. Le jeton d'identification fournit des attributs utilisateur provenant d'une source d'identité fiable pour la connexion à une application Web ou mobile. Pour plus d'informations sur les jetons d'identification, consultez [Utilisation du jeton d'identification](#).

Les utilisations du déclencheur Lambda de prégénération du jeton avec un jeton d'identification sont les suivantes.

- Lors de l'exécution, modifiez le rôle IAM que votre utilisateur demande à partir d'une réserve d'identités.
- Ajoutez des attributs utilisateur provenant d'une source externe.
- Ajoutez ou remplacez les valeurs d'attributs utilisateur existantes.
- Supprimez la divulgation des attributs utilisateur qui, en raison des portées autorisées de votre utilisateur et de l'accès en lecture aux attributs que vous avez accordé à votre client d'application, seraient autrement transmis à votre application.

Personnalisation du jeton d'accès

Avec le déclencheur Lambda de la prégénération du jeton, vous pouvez personnaliser le contenu d'un jeton d'identité (ID) à partir de votre groupe d'utilisateurs. Le jeton d'accès autorise les utilisateurs à récupérer des informations à partir de ressources dont l'accès est protégé, telles que les opérations d'API autorisées par les jetons Amazon Cognito et les API tierces. Bien que vous puissiez générer des jetons d'accès pour une autorisation machine-to-machine (M2M) avec Amazon Cognito en accordant des informations d'identification client, les demandes M2M n'invoquent pas la fonction de déclenchement préalable à la génération des jetons et ne peuvent pas émettre de jetons d'accès personnalisés. Pour plus d'informations sur les jetons d'accès, consultez [Utilisation du jeton d'accès](#).

Les utilisations du déclencheur Lambda de prégénération du jeton avec un jeton d'accès sont les suivantes.

- Ajoutez ou supprimez les portées OAuth 2.0 de la demande scope. Par exemple, vous pouvez ajouter des portées à un jeton d'accès issu de l'authentification de l'API des groupes d'utilisateurs Amazon Cognito, qui attribue uniquement la portée `aws.cognito.signin.user.admin`.
- Modifiez l'appartenance d'un utilisateur à des groupes de groupes d'utilisateurs.
- Ajoutez des demandes qui ne figurent pas déjà dans un jeton d'accès Amazon Cognito.
- Supprimez la divulgation de demandes qui seraient autrement transmises à votre application.

Pour prendre en charge la personnalisation de l'accès dans votre groupe d'utilisateurs, vous devez configurer le groupe d'utilisateurs pour générer une version mise à jour de la demande de déclenchement. Mettez à jour votre groupe d'utilisateurs comme indiqué dans la procédure suivante.

AWS Management Console

Pour prendre en charge la personnalisation du jeton d'accès dans un déclencheur Lambda de prégénération du jeton

1. Accédez à la [console Amazon Cognito](#), puis choisissez User Pools (Groupes d'utilisateurs).
2. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
3. Si ce n'est pas déjà fait, activez les [fonctionnalités de sécurité avancées](#) dans l'onglet Intégration d'application.
4. Choisissez l'onglet User pool properties (Propriétés du groupe d'utilisateurs) et localisez Lambda triggers (Déclencheurs Lambda).
5. Ajoutez ou modifiez un déclencheur de prégénération de jetons.
6. Choisissez une fonction Lambda sous Attribuer une fonction Lambda.
7. Choisissez une Version de l'événement déclencheur des Fonctionnalités de base et de la personnalisation des jetons d'accès. Ce paramètre met à jour les paramètres de demande qu'Amazon Cognito envoie à votre fonction afin d'inclure des champs pour la personnalisation des jetons d'accès.

User pools API

Pour prendre en charge la personnalisation du jeton d'accès dans un déclencheur Lambda de prégénération du jeton

Générez une demande [UpdateUserPool](#) d'API [CreateUserPool](#) ou. Vous devez spécifier une valeur pour tous les paramètres auxquels vous ne voulez pas attribuer une valeur par défaut. Pour plus d'informations, consultez [Mise à jour de la configuration d'un groupe d'utilisateurs](#).

Incluez le contenu suivant dans le paramètre `LambdaVersion` de votre demande. La valeur `LambdaVersion` de `V2_0` oblige votre groupe d'utilisateurs à ajouter des paramètres pour la personnalisation des jetons d'accès. Pour appeler une version de fonction spécifique, utilisez un ARN de fonction Lambda avec une version de fonction comme valeur de `LambdaArn`.

```
"PreTokenGenerationConfig": {
  "LambdaArn": "arn:aws:lambda:us-west-2:123456789012:function:MyFunction",
  "LambdaVersion": "V2_0"
},
```

Rubriques

- [Sources du déclencheur Lambda avant la génération de jeton](#)
- [Paramètres du déclencheur Lambda avant la génération de jeton](#)
- [Exemple de version 2 d'un événement déclencheur avant le jeton : ajout et suppression de demandes, de champs d'application et de groupes](#)
- [Exemple de version 2 d'un événement antérieur à la génération de jetons : ajout de revendications contenant des objets complexes](#)
- [Exemple de version 1 d'un événement avant la génération du jeton : ajout d'une nouvelle demande et suppression d'une demande existante](#)
- [Exemple de version 1 d'un événement avant la génération du jeton : modification de l'appartenance de l'utilisateur au groupe](#)

Sources du déclencheur Lambda avant la génération de jeton

Valeur triggerSource	Événement
TokenGeneration_HostedAuth	Appelé lors de l'authentification à partir de la page de connexion de l'interface utilisateur hébergée par Amazon Cognito.

Valeur triggerSource	Événement
TokenGeneration_Authentication	Appelé lorsque les flux d'authentification d'utilisateur sont terminés.
TokenGeneration_NewPassword Challenge	Appelé après la création de l'utilisateur par un administrateur. Ce flux est appelé lorsque l'utilisateur doit changer un mot de passe temporaire.
TokenGeneration_Authenticat eDevice	Appelé à la fin de l'authentification d'un dispositif d'utilisateur.
TokenGeneration_RefreshTokens	Appelé lorsqu'un utilisateur tente d'actualiser les jetons d'identité et d'accès.

Paramètres du déclencheur Lambda avant la génération de jeton

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes. Lorsque vous ajoutez un déclencheur Lambda de pré-génération de jetons à votre groupe d'utilisateurs, vous pouvez choisir la version du déclencheur. Cette version détermine si Amazon Cognito transmet une demande à votre fonction Lambda avec des paramètres supplémentaires pour la personnalisation des jetons d'accès.

Version 1

Le jeton de version 1 peut définir l'appartenance à un groupe, les rôles IAM et les nouvelles réclamations sous forme de jetons d'identification.

```
{
  "request": {
    "userAttributes": {"string": "string"},
    "groupConfiguration": {
      "groupsToOverride": [
        "string",
        "string"
      ],
      "iamRolesToOverride": [
        "string",
```

```

        "string"
      ],
      "preferredRole": "string"
    },
    "clientMetadata": {"string": "string"}
  },
  "response": {
    "claimsOverrideDetails": {
      "claimsToAddOrOverride": {"string": "string"},
      "claimsToSuppress": [
        "string",
        "string"
      ],
    },
    "groupOverrideDetails": {
      "groupsToOverride": [
        "string",
        "string"
      ],
      "iamRolesToOverride": [
        "string",
        "string"
      ],
      "preferredRole": "string"
    }
  }
}

```

Version 2

L'événement de demande de version 2 ajoute des champs qui personnalisent le jeton d'accès. Il ajoute également la prise en charge `claimsToOverride` des types de données complexes dans l'objet de réponse. Votre fonction Lambda peut renvoyer les types de données suivants sous la valeur de `claimsToOverride`

- Chaîne
- Nombre
- Booléen
- Tableau de chaînes, de nombres, de booléens ou d'une combinaison de ces éléments
- JSON


```
{
  "request": {
    "userAttributes": {
      "string": "string"
    },
    "scopes": ["string", "string"],
    "groupConfiguration": {
      "groupsToOverride": ["string", "string"],
      "iamRolesToOverride": ["string", "string"],
      "preferredRole": "string"
    },
    "clientMetadata": {
      "string": "string"
    }
  },
  "response": {
    "claimsAndScopeOverrideDetails": {
      "idTokenGeneration": {
        "claimsToAddOrOverride": {
          "string": [accepted datatype]
        },
        "claimsToSuppress": ["string", "string"]
      },
      "accessTokenGeneration": {
        "claimsToAddOrOverride": {
          "string": [accepted datatype]
        },
        "claimsToSuppress": ["string", "string"],
        "scopesToAdd": ["string", "string"],
        "scopesToSuppress": ["string", "string"]
      },
      "groupOverrideDetails": {
        "groupsToOverride": ["string", "string"],
        "iamRolesToOverride": ["string", "string"],
        "preferredRole": "string"
      }
    }
  }
}
```

Paramètres de demande avant la génération de jeton

Name (Nom)	Description	Version minimale de l'événement déclencheur
userAttributes	Attributs du profil de votre utilisateur dans votre groupe d'utilisateurs.	1
groupConfiguration	Objet en entrée qui contient la configuration de groupe actuelle. L'objet inclut <code>groupsToOverride</code> , <code>iamRolesToOverride</code> et <code>preferredRole</code> .	1
groupsToOverride	Groupes du groupe d'utilisateurs dont votre utilisateur est membre.	1
iamRolesToReplace	Vous pouvez associer un groupe d'utilisateurs à un rôle AWS Identity and Access Management (IAM). Cet élément est une liste de tous les rôles IAM des groupes dont votre utilisateur est membre.	1
preferredRole	Vous pouvez définir une priorité pour les groupes de groupes d'utilisateurs. Cet élément contient le nom du rôle IAM du groupe ayant la priorité la plus élevée dans l'élément <code>groupsToOverride</code> .	1
clientMetadata	Une ou plusieurs paires clé-valeur que vous pouvez spécifier et fournir en tant qu'entrée personnalisée à la fonction Lambda pour le déclencheur avant la génération de jeton.	1

Pour transmettre ces données à votre fonction Lambda, utilisez le `ClientMetadata` paramètre dans les opérations [AdminRespondToAuthChallenge](#) et [RespondToAuthChallengeAPI](#). Amazon Cognito n'inclut pas les données issues du `ClientMetadata` paramètre [AdminInitiateAuth](#) et des opérations d'[InitiateAuthAPI](#) dans la

Name (Nom)	Description	Version minimale de l'événement déclencheur
	demande transmise à la fonction de pré-génération du jeton.	
portées	Portées OAuth 2.0 de votre utilisateur. Les portées présentes dans un jeton d'accès sont les portées standard et personnalisées du groupe d'utilisateurs demandées par votre utilisateur et que vous avez autorisé votre client d'application à émettre.	2

Paramètres de réponse avant la génération de jeton

Name (Nom)	Description	Version minimale de l'événement déclencheur
claimsOverrideDetails	Un conteneur pour tous les éléments d'un événement déclencheur V1_0.	1
claimsAndScopeOverrideDetails	Un conteneur pour tous les éléments d'un événement déclencheur V2_0.	2
idTokenGeneration	Les demandes que vous souhaitez remplacer, ajouter ou supprimer dans le jeton d'identification de votre utilisateur. Les valeurs de personnalisation du parent du jeton d'identification apparaissent uniquement dans les événements de la version 2, mais les éléments enfants apparaissent dans les événements de la version 1.	2
accessTokenGeneration	Les demandes que vous souhaitez remplacer, ajouter ou supprimer dans le jeton d'accès de votre utilisateur. Les valeurs de personnalisation du parent du jeton d'accès n'apparaissent que dans les événements de la version 2.	2

Name (Nom)	Description	Version minimale de l'événement déclencheur
<code>claimsToAddOrOverride</code>	<p>Carte d'une ou de plusieurs demandes et de leurs valeurs que vous souhaitez ajouter ou modifier. Pour les demandes associées au groupe, utilisez plutôt <code>groupOverrideDetails</code> .</p> <p>Dans les événements de la version 2, cet élément apparaît à la fois sous <code>accessTokenGeneration</code> et <code>idTokenGeneration</code> .</p>	1 [*]
<code>claimsToSuppress</code>	<p>Liste des demandes que vous voulez qu'Amazon Cognito supprime. Si votre fonction supprime et remplace une valeur de revendication, Amazon Cognito supprime la revendication.</p> <p>Dans les événements de la version 2, cet élément apparaît à la fois sous <code>accessTokenGeneration</code> et <code>idTokenGeneration</code> .</p>	1

Name (Nom)	Description	Version minimale de l'événement déclencheur
groupOverrideDetails	<p>Objet en sortie qui contient la configuration de groupe actuelle. L'objet inclut <code>groupsToOverride</code> , <code>iamRolesToOverride</code> et <code>preferredRole</code> .</p> <p>Votre fonction remplace l'objet <code>groupOverrideDetails</code> par l'objet que vous fournissez. Si vous fournissez un objet vide ou null dans la réponse, Amazon Cognito supprime les groupes. Pour laisser la configuration de groupe existante telle quelle, copiez la valeur de l'objet <code>groupConfiguration</code> de la demande dans l'objet <code>groupOverrideDetails</code> de la réponse. Ensuite, retransmettez-la au service.</p> <p>Les jetons d'identification et d'accès Amazon Cognito contiennent tous les deux les demandes <code>cognito:groups</code> . Votre objet <code>groupOverrideDetails</code> remplace la demande <code>cognito:groups</code> dans les jetons d'accès et les jetons d'identification.</p>	1
scopesToAdd	Liste des portées OAuth 2.0 que vous souhaitez ajouter à la demande <code>scope</code> du jeton d'accès de votre utilisateur. Vous ne pouvez pas ajouter de valeurs de portée contenant un ou plusieurs espaces vides.	2
scopesToSuppress	Liste des portées OAuth 2.0 que vous souhaitez supprimer de la demande <code>scope</code> du jeton d'accès de votre utilisateur.	2

* Les objets de réponse aux événements de la version 1 peuvent renvoyer des chaînes. Les objets de réponse aux événements de la version 2 peuvent renvoyer [des objets complexes](#).

Exemple de version 2 d'un événement déclencheur avant le jeton : ajout et suppression de demandes, de champs d'application et de groupes

Cet exemple apporte les modifications suivantes aux jetons d'un utilisateur.

1. Définit leur valeur `family_name` comme `Doe` dans le jeton d'identification.
2. Évite que les demandes `email` et `phone_number` n'apparaissent dans le jeton d'identification.
3. Définit leur demande de jeton d'identification `cognito:roles` sur `"arn:aws:iam::123456789012:role\sns_callerA", "arn:aws:iam::123456789012:role\sns_callerC", "arn:aws:iam::123456789012:role\sns_callerB"`.
4. Définit leur demande de jeton d'identification `cognito:preferred_role` sur `arn:aws:iam::123456789012:role/sns_caller`.
5. Ajoute les portées `openid`, `email` et `solar-system-data/asteroids.add` au jeton d'accès.
6. Supprime les portées `phone_number` et `aws.cognito.signin.user.admin` du jeton d'accès. La suppression de `phone_number` empêche la récupération du numéro de téléphone de l'utilisateur à partir de `userInfo`. La suppression de `aws.cognito.signin.user.admin` empêche les demandes d'API de l'utilisateur de lire et de modifier son propre profil avec l'API des groupes d'utilisateurs Amazon Cognito.

Note

La suppression de `phone_number` des portées empêche uniquement la récupération du numéro de téléphone d'un utilisateur si les portées restantes du jeton d'accès incluent `openid` et au moins une autre portée standard. Pour plus d'informations, consultez [À propos des portées](#).

7. Définit leur demande de jeton d'identification et d'accès `cognito:groups` sur `"new-group-A", "new-group-B", "new-group-C"`.

JavaScript

```
export const handler = function(event, context) {
  event.response = {
    "claimsAndScopeOverrideDetails": {
      "idTokenGeneration": {
        "claimsToAddOrOverride": {
```

```
        "family_name": "Doe"
    },
    "claimsToSuppress": [
        "email",
        "phone_number"
    ]
},
"accessTokenGeneration": {
    "scopesToAdd": [
        "openid",
        "email",
        "solar-system-data/asteroids.add"
    ],
    "scopesToSuppress": [
        "phone_number",
        "aws.cognito.signin.user.admin"
    ]
},
"groupOverrideDetails": {
    "groupsToOverride": [
        "new-group-A",
        "new-group-B",
        "new-group-C"
    ],
    "iamRolesToOverride": [
        "arn:aws:iam::123456789012:role/new_roleA",
        "arn:aws:iam::123456789012:role/new_roleB",
        "arn:aws:iam::123456789012:role/new_roleC"
    ],
    "preferredRole": "arn:aws:iam::123456789012:role/new_role",
}
}
};
// Return to Amazon Cognito
context.done(null, event);
};
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "version": "2",
  "triggerSource": "TokenGeneration_Authentication",
  "region": "us-east-1",
  "userPoolId": "us-east-1_EXAMPLE",
  "userName": "JaneDoe",
  "callerContext": {
    "awsSdkVersion": "aws-sdk-unknown-unknown",
    "clientId": "1example23456789"
  },
  "request": {
    "userAttributes": {
      "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "cognito:user_status": "CONFIRMED",
      "email_verified": "true",
      "phone_number_verified": "true",
      "phone_number": "+12065551212",
      "family_name": "Zoe",
      "email": "Jane.Doe@example.com"
    },
    "groupConfiguration": {
      "groupsToOverride": ["group-1", "group-2", "group-3"],
      "iamRolesToOverride": ["arn:aws:iam::123456789012:role/sns_caller1",
"arn:aws:iam::123456789012:role/sns_caller2", "arn:aws:iam::123456789012:role/
sns_caller3"],
      "preferredRole": ["arn:aws:iam::123456789012:role/sns_caller"]
    },
    "scopes": [
      "aws.cognito.signin.user.admin", "openid", "email", "phone"
    ]
  },
  "response": {
    "claimsAndScopeOverrideDetails": []
  }
}
```

Exemple de version 2 d'un événement antérieur à la génération de jetons : ajout de revendications contenant des objets complexes

Cet exemple apporte les modifications suivantes aux jetons d'un utilisateur.

1. Ajoute des revendications de types numérique, chaîne, booléen et JSON au jeton d'identification. Il s'agit de la seule modification que les événements déclencheurs de la version 2 mettent à la disposition du jeton d'identification.
2. Ajoute des revendications de types numérique, chaîne, booléen et JSON au jeton d'accès.
3. Ajoute trois étendues au jeton d'accès.
4. Supprime les sub demandes email et contenues dans les jetons d'identification et d'accès.
5. Supprime l'aws.cognito.signin.user.admin étendue du jeton d'accès.

JavaScript

```
export const handler = function(event, context) {

    var scopes = ["MyAPI.read", "MyAPI.write", "MyAPI.admin"]
    var claims = {}
    claims["aud"]= event.callerContext.clientId;
    claims["booleanTest"] = false;
    claims["longTest"] = 9223372036854775807;
    claims["exponentTest"] = 1.7976931348623157E308;
    claims["ArrayTest"] = ["test", 9223372036854775807, 1.7976931348623157E308,
true];
    claims["longStringTest"] = "\\{\\
    \\\"first_json_block\\\": \\{\\
        \\\"key_A\\\": \\\"value_A\\\",\\
        \\\"key_B\\\": \\\"value_B\\\"\\
    \\},\\
    \\\"second_json_block\\\": \\{\\
        \\\"key_C\\\": \\{\\
            \\\"subkey_D\\\": [\\
                \\\"value_D\\\",\\
                \\\"value_E\\\"\\
            ],\\
            \\\"subkey_F\\\": \\\"value_F\\\"\\
        \\},\\
        \\\"key_G\\\": \\\"value_G\\\"\\
    \\}\\
    \\}\";
    claims["jsonTest"] = {
        "first_json_block": {
            "key_A": "value_A",
            "key_B": "value_B"
        },
    },
```

```

    "second_json_block": {
      "key_C": {
        "subkey_D": [
          "value_D",
          "value_E"
        ],
        "subkey_F": "value_F"
      },
      "key_G": "value_G"
    }
  };
  event.response = {
    "claimsAndScopeOverrideDetails": {
      "idTokenGeneration": {
        "claimsToAddOrOverride": claims,
        "claimsToSuppress": ["email","sub"]
      },
      "accessTokenGeneration": {
        "claimsToAddOrOverride": claims,
        "claimsToSuppress": ["email","sub"],
        "scopesToAdd": scopes,
        "scopesToSuppress": ["aws.cognito.signin.user.admin"]
      }
    }
  };
  console.info("EVENT response\n" + JSON.stringify(event, (_, v) => typeof v ===
'bigint' ? v.toString() : v, 2))
  console.info("EVENT response size\n" + JSON.stringify(event, (_, v) => typeof v
=== 'bigint' ? v.toString() : v).length)
  // Return to Amazon Cognito
  context.done(null, event);
};

```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
```

```
"version": "2",
"triggerSource": "TokenGeneration_HostedAuth",
"region": "us-west-2",
"userPoolId": "us-west-2_EXAMPLE",
"userName": "JaneDoe",
"callerContext": {
  "awsSdkVersion": "aws-sdk-unknown-unknown",
  "clientId": "1example23456789"
},
"request": {
  "userAttributes": {
    "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "cognito:user_status": "CONFIRMED"
    "email_verified": "true",
    "phone_number_verified": "true",
    "phone_number": "+12065551212",
    "email": "Jane.Doe@example.com"
  },
  "groupConfiguration": {
    "groupsToOverride": ["group-1", "group-2", "group-3"],
    "iamRolesToOverride": ["arn:aws:iam::123456789012:role/sns_caller1"],
    "preferredRole": ["arn:aws:iam::123456789012:role/sns_caller1"]
  },
  "scopes": [
    "aws.cognito.signin.user.admin",
    "phone",
    "openid",
    "profile",
    "email"
  ]
},
"response": {
  "claimsAndScopeOverrideDetails": []
}
}
```

Exemple de version 1 d'un événement avant la génération du jeton : ajout d'une nouvelle demande et suppression d'une demande existante

Cet exemple utilise l'événement déclencheur version 1 avec une fonction Lambda de prégénération du jeton pour ajouter une nouvelle demande et en supprimer une existante.

Node.js

```
const handler = async (event) => {
  event.response = {
    claimsOverrideDetails: {
      claimsToAddOrOverride: {
        my_first_attribute: "first_value",
        my_second_attribute: "second_value",
      },
      claimsToSuppress: ["email"],
    },
  };

  return event;
};

export { handler };
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des données pertinentes pour votre déclencheur Lambda. Un événement de test pour cet exemple de code est fourni ci-dessous. Comme l'exemple de code ne traite pas de paramètres de demande, vous pouvez utiliser un événement de test avec une demande vide. Pour plus d'informations sur les paramètres de demande communs, consultez [Événement déclencheur Lambda d'un groupe d'utilisateurs](#).

JSON

```
{
  "request": {},
  "response": {}
}
```

Exemple de version 1 d'un événement avant la génération du jeton : modification de l'appartenance de l'utilisateur au groupe

Cet exemple utilise l'événement déclencheur version 1 avec une fonction Lambda avant la génération du jeton pour modifier l'appartenance au groupe de l'utilisateur.

Node.js

```
const handler = async (event) => {
  event.response = {
    claimsOverrideDetails: {
      groupOverrideDetails: {
        groupsToOverride: ["group-A", "group-B", "group-C"],
        iamRolesToOverride: [
          "arn:aws:iam::XXXXXXXXXXXX:role/sns_callerA",
          "arn:aws:iam::XXXXXXXXXXXX:role/sns_callerB",
          "arn:aws:iam::XXXXXXXXXXXX:role/sns_callerC",
        ],
        preferredRole: "arn:aws:iam::XXXXXXXXXXXX:role/sns_caller",
      },
    },
  },
};

return event;
};

export { handler };
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "request": {},
  "response": {}
}
```

Déclencheur Lambda de migration d'utilisateur

Quand un utilisateur n'existe pas dans le groupe d'utilisateurs au moment de la connexion avec un mot de passe, ou dans le flux de mot de passe oublié, Amazon Cognito appelle ce déclencheur.

Lorsque la fonction Lambda renvoie une réponse positive, Amazon Cognito crée l'utilisateur dans le groupe d'utilisateurs. Pour en savoir plus sur le flux d'authentification avec le déclencheur Lambda de migration d'utilisateur, consultez [Importation d'utilisateurs dans des groupes d'utilisateurs avec un déclencheur Lambda de migration d'utilisateur](#).

Pour migrer des utilisateurs de votre annuaire d'utilisateurs existant vers des groupes d'utilisateurs Amazon Cognito au moment de la connexion ou pendant le flux de mot de passe oublié, utilisez ce déclencheur Lambda.

Rubriques

- [Sources du déclencheur Lambda Migration d'utilisateur](#)
- [Paramètres du déclencheur Lambda Migration d'utilisateur](#)
- [Exemple de migration de l'utilisateur : Migration d'un utilisateur avec un mot de passe existant](#)

Sources du déclencheur Lambda Migration d'utilisateur

Valeur triggerSource	Événement
UserMigration_Authentication	Migration des utilisateurs lors de la connexion.
UserMigration_ForgotPassword	Migration de l'utilisateur dans le cadre du flux de mot de passe oublié.

Paramètres du déclencheur Lambda Migration d'utilisateur

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "userName": "string",
  "request": {
    "password": "string",
    "validationData": {
      "string": "string",
```

```

    . . .
  },
  "clientMetadata": {
    "string": "string",
    . . .
  }
},
"response": {
  "userAttributes": {
    "string": "string",
    . . .
  },
  "finalUserStatus": "string",
  "messageAction": "string",
  "desiredDeliveryMediums": [ "string", . . . ],
  "forceAliasCreation": boolean,
  "enableSMSMFA": boolean
}
}

```

Paramètres de demande de migration d'utilisateur

userName

Nom d'utilisateur que l'utilisateur saisit lors de la connexion.

mot de passe

Mot de passe que l'utilisateur saisit lors de la connexion. Amazon Cognito n'envoie pas cette valeur dans une demande initiée par un flux de mot de passe oublié.

validationData

Une ou plusieurs paires clé-valeur contenant les données de validation dans la demande de connexion de l'utilisateur. Pour transmettre ces données à votre fonction Lambda, vous pouvez utiliser le paramètre ClientMetadata dans les actions d'API [InitiateAuth](#) et [AdminInitiateAuth](#).

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée à la fonction Lambda pour le déclencheur de migration d'utilisateur. Pour transmettre ces données à votre fonction Lambda, vous pouvez utiliser le paramètre ClientMetadata dans les actions d'API [AdminRespondToAuthChallenge](#) et [ForgotPassword](#).

Paramètres de réponse de migration d'utilisateur

userAttributes


Ce champ est obligatoire.

Ce champ doit contenir une ou plusieurs paires nom-valeur qu'Amazon Cognito stocke dans le profil utilisateur dans votre groupe d'utilisateurs et utilise comme attributs utilisateur. Vous pouvez inclure les attributs utilisateur standard et personnalisés. Les attributs personnalisés ont besoin du préfixe `custom:` pour se distinguer des attributs standard. Pour plus d'informations, consultez [Attributs personnalisés](#).

Note

Pour réinitialiser son mot de passe dans le flux de mot de passe oublié, un utilisateur doit avoir une adresse e-mail vérifiée ou un numéro de téléphone vérifié. Amazon Cognito envoie un message contenant un code de réinitialisation de mot de passe à l'adresse e-mail ou au numéro de téléphone spécifiés dans les attributs utilisateur.

Attributs	Exigence
Tous les attributs signalés comme obligatoires lorsque vous avez créé le groupe d'utilisateurs	Si des attributs requis sont manquants au cours de la migration, Amazon Cognito utilise les valeurs par défaut.
<code>username</code>	<p>Requis si vous avez configuré votre groupe d'utilisateurs avec des attributs d'alias en plus du nom d'utilisateur pour la connexion, et que l'utilisateur a entré une valeur d'alias valide comme nom d'utilisateur. Cette valeur d'alias peut être une adresse e-mail, un nom d'utilisateur préféré ou un numéro de téléphone.</p> <p>Si la demande et le groupe d'utilisateurs répondent aux exigences relatives aux alias, la réponse de votre fonction doit attribuer le paramètre <code>username</code> qu'elle a reçu à un attribut d'alias. De plus, la réponse doit attribuer votre propre valeur à l'attribut <code>username</code>.</p>

Attributs	Exigence
	<p>Si votre groupe d'utilisateurs ne répond pas aux conditions requises pour mapper le paramètre <code>username</code> reçu sur un alias, le paramètre <code>username</code> figurant dans la réponse doit correspondre exactement à la demande ou être omis.</p> <div data-bbox="553 432 1507 604"><p> Note</p><p><code>username</code> doit être unique dans le groupe d'utilisateurs.</p></div>

`finalUserStatus`

Vous pouvez définir ce paramètre sur `CONFIRMED` pour confirmer automatiquement vos utilisateurs afin qu'ils puissent se connecter avec leurs mots de passe précédents. Lorsque vous définissez un utilisateur sur `CONFIRMED`, il n'a pas besoin de prendre des mesures supplémentaires avant de pouvoir se connecter. Si vous ne définissez pas cet attribut sur `CONFIRMED`, il est réglé sur `RESET_REQUIRED`.

Un paramètre `finalUserStatus` égal à `RESET_REQUIRED` signifie que l'utilisateur doit modifier son mot de passe immédiatement après la migration lors de la connexion, et que votre application cliente doit gérer l'exception `PasswordResetRequiredException` pendant le flux d'authentification.

Note

Amazon Cognito n'applique pas la politique de complexité de mot de passe que vous avez configurée pour le groupe d'utilisateurs lors de la migration à l'aide du déclencheur Lambda. Si le mot de passe ne répond pas à la politique de mot de passe que vous avez configurée, Amazon Cognito accepte toujours le mot de passe pour pouvoir continuer à migrer l'utilisateur. Pour appliquer la politique de sécurité du mot de passe et rejeter les mots de passe non conformes à celle-ci, validez la sécurité du mot de passe dans votre code. Ensuite, si le mot de passe n'est pas conforme à la politique, définissez `finalUserStatus` sur `RESET_REQUIRED`.

messageAction

Vous pouvez définir ce paramètre sur `SUPPRESS` pour refuser d'envoyer le message de bienvenue qu'Amazon Cognito envoie habituellement aux nouveaux utilisateurs. Si votre fonction ne renvoie pas ce paramètre, Amazon Cognito envoie le message de bienvenue.

desiredDeliveryMediums

Vous pouvez définir ce paramètre sur `EMAIL` pour envoyer le message de bienvenue par e-mail, ou sur `SMS` pour l'envoyer par SMS. Si votre fonction ne renvoie pas ce paramètre, Amazon Cognito envoie le message de bienvenue par SMS.

forceAliasCreation

Si vous définissez ce paramètre sur `TRUE` et que le numéro de téléphone ou l'adresse e-mail spécifiés dans le paramètre `UserAttributes` existe déjà en tant qu'alias pour un autre utilisateur, l'appel d'API migre l'alias de cet autre utilisateur vers l'utilisateur nouvellement créé. L'utilisateur précédent ne peut plus se connecter à l'aide de cet alias.

Si vous définissez ce paramètre sur `FALSE` et que l'alias existe, Amazon Cognito ne migre pas l'utilisateur et renvoie une erreur à l'application cliente.

Si vous ne renvoyez pas ce paramètre, Amazon Cognito suppose que sa valeur est « `false` ».

enableSMSMFA

Définissez ce paramètre sur `true` pour exiger de votre utilisateur migré qu'il effectue l'authentification multifactorielle (MFA) par SMS pour se connecter. L'authentification MFA doit être activée pour votre groupe d'utilisateurs. Les attributs de votre utilisateur dans les paramètres de demande doivent inclure un numéro de téléphone, sans quoi la migration de cet utilisateur échouera.

Exemple de migration de l'utilisateur : Migration d'un utilisateur avec un mot de passe existant

Cet exemple de fonction Lambda migre l'utilisateur avec un mot de passe existant et supprime le message de bienvenue d'Amazon Cognito.

Node.js

```
const validUsers = {
  belladonna: { password: "Test123", emailAddress: "bella@example.com" },
```

```
};

// Replace this mock with a call to a real authentication service.
const authenticateUser = (username, password) => {
  if (validUsers[username] && validUsers[username].password === password) {
    return validUsers[username];
  } else {
    return null;
  }
};

const lookupUser = (username) => {
  const user = validUsers[username];

  if (user) {
    return { emailAddress: user.emailAddress };
  } else {
    return null;
  }
};

const handler = async (event) => {
  if (event.triggerSource == "UserMigration_Authentication") {
    // Authenticate the user with your existing user directory service
    const user = authenticateUser(event.userName, event.request.password);
    if (user) {
      event.response.userAttributes = {
        email: user.emailAddress,
        email_verified: "true",
      };
      event.response.finalUserStatus = "CONFIRMED";
      event.response.messageAction = "SUPPRESS";
    }
  } else if (event.triggerSource == "UserMigration_ForgotPassword") {
    // Look up the user in your existing user directory service
    const user = lookupUser(event.userName);
    if (user) {
      event.response.userAttributes = {
        email: user.emailAddress,
        // Required to enable password-reset code to be sent to user
        email_verified: "true",
      };
      event.response.messageAction = "SUPPRESS";
    }
  }
}
```

```
}  
  
    return event;  
};  
  
export { handler };
```

Déclencheur Lambda message personnalisé

Amazon Cognito appelle ce déclencheur avant d'envoyer un message de vérification ou un code d'authentification multifactor (MFA) par e-mail ou par téléphone. Vous pouvez personnaliser le message de manière dynamique avec votre déclencheur de message personnalisé. Vous pouvez modifier les messages personnalisés statiques sous l'onglet Message Customizations (Personnalisation des messages) de la console [Amazon Cognito](#).

La demande inclut `codeParameter`. Il s'agit d'une chaîne qui sert d'espace réservé pour le code remis par Amazon Cognito à l'utilisateur. Insérez la chaîne `codeParameter` dans le corps du message, à l'emplacement où vous souhaitez que le code de vérification apparaisse. Quand il reçoit cette réponse, Amazon Cognito remplace la chaîne `codeParameter` par le code de vérification réel.

Note

Une fonction Lambda de message personnalisé avec la source du déclencheur `CustomMessage_AdminCreateUser` renvoie un nom d'utilisateur et un code de vérification. Étant donné qu'un utilisateur créé par l'administrateur doit recevoir à la fois son nom d'utilisateur et son code, la réponse de votre fonction doit inclure à la fois `request.usernameParameter` et `request.codeParameter`.

Rubriques

- [Sources du déclencheur Lambda message personnalisé](#)
- [Paramètres du déclencheur Lambda message personnalisé](#)
- [Exemple de message personnalisé pour l'inscription](#)
- [Exemple de message personnalisé pour la création d'utilisateurs par les administrateurs](#)

Sources du déclencheur Lambda message personnalisé

Valeur triggerSource	Événement
CustomMessage_SignUp	Message personnalisé – Pour envoyer le code de confirmation après l'inscription.
CustomMessage_AdminCreateUser	Message personnalisé – Pour envoyer le mot de passe temporaire à un nouvel utilisateur.
CustomMessage_ResendCode	Message personnalisé – Pour renvoyer le code de confirmation à un utilisateur existant.
CustomMessage_ForgotPassword	Message personnalisé – Pour envoyer le code de confirmation suite à une demande de mot de passe oublié.
CustomMessage_UpdateUserAttribute	Message personnalisé – En cas de modification de l'e-mail ou du numéro de téléphone, ce déclencheur envoie automatiquement un code de vérification à l'utilisateur. Impossible à utiliser pour d'autres attributs.
CustomMessage_VerifyUserAttribute	Message personnalisé – Ce déclencheur envoie un code de vérification à l'utilisateur lorsque ce dernier en fait manuellement la demande concernant une nouvelle adresse e-mail ou un nouveau numéro de téléphone.
CustomMessage_Authentication	Message personnalisé – Pour envoyer le code MFA lors de l'authentification.

Paramètres du déclencheur Lambda message personnalisé

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
    "userAttributes": {
      "string": "string",
      . . .
    }
    "codeParameter": "####",
    "usernameParameter": "string",
    "clientMetadata": {
      "string": "string",
      . . .
    }
  },
  "response": {
    "smsMessage": "string",
    "emailMessage": "string",
    "emailSubject": "string"
  }
}
```

Paramètres de demande de message personnalisé

userAttributes

Une ou plusieurs paires nom-valeur représentant des attributs utilisateur.

codeParameter

Chaîne que vous pouvez utiliser comme espace réservé pour le code de vérification dans le message personnalisé.

usernameParameter

Nom de l'utilisateur. Amazon Cognito inclut ce paramètre dans les demandes provenant d'utilisateurs créés par l'administrateur.

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée à la fonction Lambda que vous spécifiez pour le déclencheur de message personnalisé. La demande qui invoque une fonction de message personnalisé n'inclut pas les données transmises dans

le `ClientMetadata` paramètre ni dans [AdminInitiateAuth](#) les opérations [InitiateAuth](#) d'API. Pour transmettre ces données à votre fonction Lambda, vous pouvez utiliser le `ClientMetadata` paramètre dans les actions d'API suivantes :

- [AdminResetUserPassword](#)
- [AdminRespondToAuthChallenge](#)
- [AdminUpdateUserAttributes](#)
- [ForgotPassword](#)
- [GetUserAttributeVerificationCode](#)
- [ResendConfirmationCode](#)
- [SignUp](#)
- [UpdateUserAttributes](#)

Paramètres de réponse de message personnalisé

Dans la réponse, spécifiez le texte personnalisé à utiliser dans les messages à vos utilisateurs. Pour connaître les contraintes de chaîne appliquées par Amazon Cognito à ces paramètres, consultez.

[MessageTemplateType](#)

smsMessage

Message SMS personnalisé à envoyer à vos utilisateurs. Il doit inclure la valeur `codeParameter` que vous avez reçue dans la demande.

emailMessage

Message électronique personnalisé à envoyer à vos utilisateurs. Vous pouvez utiliser le formatage HTML dans le paramètre `emailMessage`. Il doit inclure la valeur `codeParameter` que vous avez reçue dans la demande en tant que variable `{####}`. Amazon Cognito peut utiliser le paramètre `emailMessage` seulement si l'attribut `EmailSendingAccount` du groupe d'utilisateurs est `DEVELOPER`. Si l'attribut `EmailSendingAccount` du groupe d'utilisateurs n'est pas `DEVELOPER` et qu'un paramètre `emailMessage` est renvoyé, Amazon Cognito génère un code d'erreur `400 com.amazonaws.cognito.identity.idp.model.InvalidLambdaResponseException`. Lorsque vous choisissez d'utiliser Amazon Simple Email Service (Amazon SES) pour envoyer des messages électroniques, l'attribut `EmailSendingAccount` d'un groupe d'utilisateurs est `DEVELOPER`. Sinon, la valeur est `COGNITO_DEFAULT`.

emailSubject

Ligne d'objet pour le message personnalisé. Vous ne pouvez utiliser le `emailSubject` paramètre que si l'attribut `EmailSendingAccount` du groupe d'utilisateurs est `DEVELOPER`. Si l'attribut `EmailSendingAccount` du groupe d'utilisateurs n'est pas `DEVELOPER` et qu'Amazon Cognito renvoie un paramètre `emailSubject`, Amazon Cognito génère un code d'erreur `400 com.amazonaws.cognito.idp.model.InvalidLambdaResponseException`. L'attribut `EmailSendingAccount` d'un groupe d'utilisateurs est `DEVELOPER` quand vous choisissez d'utiliser Amazon Simple Email Service (Amazon SES) pour envoyer des messages électroniques. Sinon, la valeur est `COGNITO_DEFAULT`.

Exemple de message personnalisé pour l'inscription

Cette exemple de fonction Lambda personnalise un e-mail ou un SMS quand le service nécessite qu'une application envoie un code de vérification à l'utilisateur.

Amazon Cognito peut appeler un déclencheur Lambda lors de plusieurs événements : après l'inscription, lors du renvoi d'un code de vérification, lors de la récupération d'un mot de passe oublié ou lors de la vérification d'un attribut utilisateur. La réponse inclut des messages par SMS et e-mail. Le message doit inclure le paramètre de code "####". Ce paramètre est l'espace réservé pour le code de vérification que l'utilisateur reçoit.

La longueur maximale d'un message électronique est de 20 000 caractères UTF-8. Cette longueur inclut le code de vérification. Vous pouvez utiliser des balises HTML dans ces messages électroniques.

La longueur maximale d'un SMS est de 140 caractères UTF-8. Cette longueur inclut le code de vérification.

Node.js

```
const handler = async (event) => {
  if (event.triggerSource === "CustomMessage_SignUp") {
    const message = `Thank you for signing up. Your confirmation code is
    ${event.request.codeParameter}.`;
    event.response.smsMessage = message;
    event.response.emailMessage = message;
    event.response.emailSubject = "Welcome to the service.";
  }
}
```



```
    return event;
};

export { handler };
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "version": 1,
  "triggerSource": "CustomMessage_SignUp/CustomMessage_ResendCode/CustomMessage_ForgotPassword/CustomMessage_VerifyUserAttribute",
  "region": "<region>",
  "userPoolId": "<userPoolId>",
  "userName": "<userName>",
  "callerContext": {
    "awsSdk": "<calling aws sdk with version>",
    "clientId": "<apps client id>",
    ...
  },
  "request": {
    "userAttributes": {
      "phone_number_verified": false,
      "email_verified": true,
      ...
    },
    "codeParameter": "####"
  },
  "response": {
    "smsMessage": "<custom message to be sent in the message with code parameter>"
    "emailMessage": "<custom message to be sent in the message with code parameter>"
    "emailSubject": "<custom email subject>"
  }
}
```

Exemple de message personnalisé pour la création d'utilisateurs par les administrateurs

La demande envoyée par Amazon Cognito à cet exemple de fonction Lambda de message personnalisé comporte une `triggerSource` valeur, un nom d'utilisateur `CustomMessage_AdminCreateUser` et un mot de passe temporaire. La fonction est renseignée `${event.request.codeParameter}` à partir du mot de passe temporaire de la demande et `${event.request.usernameParameter}` du nom d'utilisateur de la demande.

Vos messages personnalisés doivent insérer les valeurs de `codeParameter` et `usernameParameter` dans `smsMessage` et `emailMessage` dans l'objet de réponse. Dans cet exemple, la fonction écrit le même message dans les champs de réponse `event.response.smsMessage` et `event.response.emailMessage`.

La longueur maximale d'un message électronique est de 20 000 caractères UTF-8. Cette longueur inclut le code de vérification. Vous pouvez utiliser des balises HTML dans ces e-mails. La longueur maximale d'un SMS est de 140 caractères UTF-8. Cette longueur inclut le code de vérification.

La réponse inclut des messages par SMS et e-mail.

Node.js

```
const handler = async (event) => {
  if (event.triggerSource === "CustomMessage_AdminCreateUser") {
    const message = `Welcome to the service. Your user name is
${event.request.usernameParameter}. Your temporary password is
${event.request.codeParameter}`;
    event.response.smsMessage = message;
    event.response.emailMessage = message;
    event.response.emailSubject = "Welcome to the service";
  }
  return event;
};

export { handler }
```

Amazon Cognito transmet les informations d'événement à votre fonction Lambda. Cette fonction renvoie alors le même objet d'événement à Amazon Cognito, avec d'éventuelles modifications dans la réponse. Dans la console Lambda, vous pouvez configurer un événement de test avec des

données pertinentes pour votre déclencheur Lambda. Voici un événement de test pour cet exemple de code :

JSON

```
{
  "version": 1,
  "triggerSource": "CustomMessage_AdminCreateUser",
  "region": "<region>",
  "userPoolId": "<userPoolId>",
  "userName": "<userName>",
  "callerContext": {
    "awsSdk": "<calling aws sdk with version>",
    "clientId": "<apps client id>",
    ...
  },
  "request": {
    "userAttributes": {
      "phone_number_verified": false,
      "email_verified": true,
      ...
    },
    "codeParameter": "####",
    "usernameParameter": "username"
  },
  "response": {
    "smsMessage": "<custom message to be sent in the message with code parameter and username parameter>"
    "emailMessage": "<custom message to be sent in the message with code parameter and username parameter>"
    "emailSubject": "<custom email subject>"
  }
}
```

Déclencheurs Lambda Expéditeur personnalisé

Les groupes d'utilisateurs Amazon Cognito fournissent les déclencheurs Lambda `CustomEmailSender` et `CustomSMSSender` pour activer les notifications tierces par e-mail et par SMS. Vous pouvez choisir des fournisseurs de SMS et d'e-mail pour envoyer des notifications aux utilisateurs à partir du code de votre fonction Lambda. Quand Amazon Cognito doit envoyer des notifications telles que des codes de confirmation, des codes de vérification ou des mots de

se passe temporaires à des utilisateurs, les événements activent vos fonctions Lambda configurées. Amazon Cognito envoie le code et les mots de passe temporaires (secrets) à vos fonctions Lambda activées. Amazon Cognito chiffre ces secrets avec une clé AWS KMS gérée par le client et le kit AWS Encryption SDK. Le kit AWS Encryption SDK est une bibliothèque de chiffrement côté serveur qui vous aide à chiffrer et déchiffrer les données génériques.

Note

Pour configurer vos groupes d'utilisateurs afin d'utiliser ces déclencheurs Lambda, vous pouvez utiliser l'interface AWS CLI ou le kit SDK. Ces configurations ne sont pas disponibles à partir de la console Amazon Cognito.

CustomEmailSender

Amazon Cognito appelle ce déclencheur pour envoyer des notifications par e-mail à des utilisateurs.

CustomSMSSender

Amazon Cognito appelle ce déclencheur pour envoyer des notifications par SMS à des utilisateurs.

Ressources

Les ressources suivantes peuvent vous aider à utiliser les déclencheurs CustomEmailSender et CustomSMSSender.

AWS KMS

AWS KMS est un service géré permettant de créer et de contrôler des clés AWS KMS. Ces clés chiffrent vos données. Pour plus d'informations, consultez [Qu'est-ce qu'AWS Key Management Service ?](#).

Clé KMS

Une clé KMS est une représentation logique d'une clé cryptographique. La clé KMS inclut des métadonnées, telles que l'ID de clé, la date de création, la description et l'état de la clé. La clé KMS contient également les éléments de clé utilisés pour chiffrer et déchiffrer les données. Pour plus d'informations, consultez [Clés AWS KMS](#).

Clé KMS symétrique

Une clé KMS symétrique est une clé de chiffrement 256 bits qui ne sort pas de AWS KMS sous forme non chiffrée. Pour utiliser une clé KMS symétrique, vous devez appeler AWS KMS. Amazon Cognito utilise des clés symétriques. La même clé sert à chiffrer et déchiffrer. Pour plus d'informations, consultez [Clés KMS symétriques](#).

Déclencheur Lambda expéditeur d'e-mail personnalisé

Lorsque vous attribuez un déclencheur d'expéditeur d'e-mails personnalisé à votre groupe d'utilisateurs, Amazon Cognito invoque une fonction Lambda à la place de son comportement par défaut lorsqu'un événement utilisateur nécessite l'envoi d'un e-mail. Avec un déclencheur d'expéditeur personnalisé, votre fonction AWS Lambda peut envoyer des notifications par e-mail à vos utilisateurs via une méthode et un fournisseur que vous choisirez. Le code personnalisé de votre fonction doit traiter et livrer tous les e-mails de votre groupe d'utilisateurs.

Note

Actuellement, vous ne pouvez pas attribuer de déclencheurs d'expéditeur personnalisé dans la console Amazon Cognito. Vous pouvez attribuer un déclencheur à l'aide du paramètre `LambdaConfig` dans une demande d'API `CreateUserPool` ou `UpdateUserPool`.

Pour configurer ce déclencheur, effectuez les opérations suivantes :

1. Créez une [clé de chiffrement symétrique](#) dans AWS Key Management Service (AWS KMS). Amazon Cognito génère des secrets (mots de passe, codes de vérification et codes de confirmation temporaires), puis utilise cette clé KMS pour chiffrer les secrets. Vous pouvez ensuite utiliser l'opération API [Decrypt](#) dans votre fonction Lambda pour déchiffrer les secrets et les envoyer à l'utilisateur en texte brut. Le kit [AWS Encryption SDK](#) est un outil utile pour les opérations AWS KMS dans votre fonction.
2. Créez une fonction Lambda à attribuer en tant que déclencheur d'expéditeur personnalisé. Accordez des autorisations `kms:Decrypt` au rôle de la fonction Lambda pour votre clé KMS.
3. Accordez l'accès au principal de service Amazon Cognito `cognito-idp.amazonaws.com` pour appeler la fonction Lambda.
4. Écrivez un code de fonction Lambda qui dirige vos messages vers des méthodes de remise personnalisées ou des fournisseurs tiers. Pour fournir le code de vérification ou de confirmation de

votre utilisateur, décidez et déchiffrez en Base64 la valeur du paramètre code dans la demande. Cette opération produit un code ou un mot de passe en texte brut que vous devez inclure dans votre message.

5. Mettez à jour le groupe d'utilisateurs pour qu'il utilise un déclencheur Lambda Expéditeur personnalisé. Le principal IAM qui met à jour ou crée un groupe d'utilisateurs avec un déclencheur d'expéditeur personnalisé doit être autorisé à créer un octroi pour votre clé KMS. L'extrait de code LambdaConfig suivant attribue des fonctions personnalisées d'expéditeur de SMS et d'e-mails.

```
"LambdaConfig": {
  "KMSKeyID": "arn:aws:kms:us-
east-1:123456789012:key/a6c4f8e2-0c45-47db-925f-87854bc9e357",
  "CustomEmailSender": {
    "LambdaArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
    "LambdaVersion": "V1_0"
  },
  "CustomSMSSender": {
    "LambdaArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
    "LambdaVersion": "V1_0"
  }
}
```

Paramètres de déclencheur Lambda d'expéditeur d'e-mail personnalisé

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
    "type": "customEmailSenderRequestV1",
    "code": "string",
    "clientMetadata": {
      "string": "string",
      . . .
    },
    "userAttributes": {
      "string": "string",
      . . .
    }
  }
}
```

```
}
```

Paramètres de demande d'expéditeur d'e-mail personnalisé

type

Version de la demande. Pour un événement d'expéditeur d'e-mail personnalisé, la valeur de cette chaîne est toujours `customEmailSenderRequestV1`.

code

Code chiffré que votre fonction peut déchiffrer et envoyer à votre utilisateur.

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée au déclencheur de la fonction Lambda d'expéditeur d'e-mail personnalisé. Pour transmettre ces données à votre fonction Lambda, vous pouvez utiliser le paramètre `ClientMetadata` dans les actions d'API [AdminRespondToAuthChallenge](#) et [RespondToAuthChallenge](#). Amazon Cognito n'inclut pas les données provenant du paramètre `ClientMetadata` dans les opérations d'API [AdminInitiateAuth](#) et [InitiateAuth](#) dans la demande transmise à la fonction après authentification.

userAttributes

Une ou plusieurs paires clé-valeur qui représentent les attributs utilisateur.

Paramètres de réponse d'expéditeur d'e-mail personnalisé

Amazon Cognito n'attend aucune information en retour supplémentaire dans la réponse d'expéditeur d'e-mail personnalisé. Votre fonction peut utiliser les opérations d'API pour interroger et modifier vos ressources, ou enregistrer des métadonnées d'événements dans un système externe.

Activation du déclencheur Lambda d'expéditeur d'e-mail personnalisé

Pour configurer un déclencheur d'expéditeur d'e-mail personnalisé qui utilise une logique personnalisée pour envoyer des e-mails pour votre groupe d'utilisateurs, activez le déclencheur comme suit. La procédure qui suit attribue un déclencheur d'e-mail personnalisé, un déclencheur de SMS personnalisé, ou les deux, à votre groupe d'utilisateurs. Une fois que vous avez ajouté votre déclencheur d'expéditeur d'e-mail personnalisé, Amazon Cognito envoie toujours les attributs utilisateur, dont l'adresse e-mail et le code à usage unique à votre fonction Lambda, alors qu'il aurait autrement envoyé un e-mail avec Amazon Simple Email Service.

⚠ Important

Amazon Cognito échappe en HTML les caractères réservés tels que `<` (`<`) et `>` (`>`) dans le mot de passe temporaire de votre utilisateur. Ces caractères peuvent apparaître dans les mots de passe temporaires qu'Amazon Cognito envoie à votre fonction d'expéditeur d'e-mail personnalisé, mais ils n'apparaissent pas dans les codes de vérification temporaires. Pour envoyer des mots de passe temporaires, votre fonction Lambda doit annuler l'échappement de ces caractères après avoir déchiffré le mot de passe et avant d'envoyer le message à votre utilisateur.

1. Créez une clé de chiffrement dans AWS KMS. Cette clé chiffre les mots de passe temporaires et les codes d'autorisation générés par Amazon Cognito. Vous pouvez ensuite déchiffrer ces secrets dans la fonction Lambda d'expéditeur personnalisé et les envoyer à l'utilisateur en texte brut.
2. Accordez au principal du service Amazon Cognito un accès `cognito-idp.amazonaws.com` pour chiffrer les codes à l'aide de la clé KMS.

Appliquez à votre clé KMS la stratégie basée sur les ressources suivante.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "cognito-idp.amazonaws.com"
    },
    "Action": "kms:CreateGrant",
    "Resource": "arn:aws:kms:us-
west-2:111222333444:key/1example-2222-3333-4444-999example",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111222333444"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:cognito-idp:us-
west-2:111222333444:userpool/us-east-1_EXAMPLE"
      }
    }
  }
}]
```



```
}
```

3. Créez une fonction Lambda pour le déclencheur d'expéditeur personnalisé. Amazon Cognito utilise le [kit SDK de chiffrement AWS](#) pour chiffrer les secrets, les mots de passe et les codes temporaires qui autorisent les demandes d'API de vos utilisateurs.
 - Attribuez un rôle IAM à votre fonction Lambda qui, au minimum, dispose d'autorisations `kms:Decrypt` pour votre clé KMS.
4. Accordez l'accès au principal de service Amazon Cognito `cognito-idp.amazonaws.com` pour appeler la fonction Lambda.

La commande AWS CLI suivante accorde à Amazon Cognito l'autorisation d'appeler votre fonction Lambda :

```
aws lambda add-permission --function-name lambda_arn --statement-id  
"CognitoLambdaInvokeAccess" --action lambda:InvokeFunction --principal cognito-  
idp.amazonaws.com
```

5. Composez le code de votre fonction Lambda pour qu'elle envoie vos messages. Amazon Cognito utilise AWS Encryption SDK pour chiffrer les secrets avant de les envoyer à la fonction Lambda d'expéditeur personnalisé. Dans votre fonction, déchiffrez le secret et traitez les métadonnées pertinentes. Envoyez ensuite le code, votre propre message personnalisé ainsi que le numéro de téléphone de destination à l'API personnalisée qui remet votre message.
6. Ajoutez AWS Encryption SDK à votre fonction Lambda. Pour en savoir plus, consultez [Langages de programmation du kit SDK de chiffrement AWS](#). Pour mettre à jour le package Lambda, effectuez les étapes suivantes.
 - a. Exportez votre fonction Lambda sous forme de fichier `.zip` dans la AWS Management Console.
 - b. Ouvrez votre fonction et ajoutez AWS Encryption SDK. Pour obtenir des informations supplémentaires et les liens de téléchargement, consultez [Langages de programmation du AWS Encryption SDK](#) dans le Guide du développeur AWS Encryption SDK.
 - c. Zippez votre fonction avec les dépendances du kit SDK et chargez-la sur Lambda. Pour en savoir plus, consultez [Déploiement de fonctions Lambda sous forme d'archives de fichiers .zip](#) dans le Guide du développeur AWS Lambda.

7. Mettez à jour votre groupe d'utilisateurs pour ajouter des déclencheurs Lambda d'expéditeur personnalisé. Incluez un paramètre `CustomSMSSender` ou `CustomEmailSender` dans une demande d'API `UpdateUserPool`. L'opération d'API `UpdateUserPool` a besoin de tous les paramètres de votre groupe d'utilisateurs et des paramètres que vous voulez modifier. Si vous ne fournissez pas tous les paramètres pertinents, Amazon Cognito définit les valeurs de tous les paramètres manquants sur leurs valeurs par défaut. Comme dans l'exemple suivant, incluez des entrées pour toutes les fonctions Lambda que vous souhaitez ajouter ou conserver dans votre groupe d'utilisateurs. Pour de plus amples informations, veuillez consulter [Mise à jour de la configuration d'un groupe d'utilisateurs](#).

```
#Send this parameter in an 'aws cognito-idp update-user-pool' CLI command,
including any existing
#user pool configurations.

--lambda-config "PreSignUp=lambda-arn, \
                 CustomSMSSender={LambdaVersion=V1_0,LambdaArn=lambda-arn}, \
                 CustomEmailSender={LambdaVersion=V1_0,LambdaArn=lambda-arn},
\
                 KMSKeyID=key-id"
```

Pour supprimer un déclencheur Lambda d'expéditeur personnalisé avec une commande `update-user-pool` AWS CLI, omettez le paramètre `CustomSMSSender` ou `CustomEmailSender` dans `--lambda-config`, puis incluez tous les autres déclencheurs que vous souhaitez utiliser avec votre groupe d'utilisateurs.

Pour supprimer un déclencheur Lambda d'expéditeur personnalisé avec une demande d'API `UpdateUserPool`, omettez le paramètre `CustomSMSSender` ou `CustomEmailSender` dans corps de la demande qui contient le reste de la configuration de votre groupe d'utilisateurs.

Exemple de code

L'exemple Node.js suivant traite un événement d'e-mail dans votre fonction Lambda d'expéditeur d'e-mail personnalisé. Cet exemple suppose que votre fonction possède deux variables d'environnement définies.

KEY_ALIAS

L'[alias](#) de la clé KMS que vous souhaitez utiliser pour chiffrer et déchiffrer les codes de vos utilisateurs.

KEY_ARN

L'Amazon Resource Name (ARN) de la clé KMS que vous souhaitez utiliser pour chiffrer et déchiffrer les codes de vos utilisateurs.

```
const AWS = require('aws-sdk');
const b64 = require('base64-js');
const encryptionSdk = require('@aws-crypto/client-node');
//Configure the encryption SDK client with the KMS key from the environment variables.
const { encrypt, decrypt } =
  encryptionSdk.buildClient(encryptionSdk.CommitmentPolicy.REQUIRE_ENCRYPT_ALLOW_DECRYPT);
const generatorKeyId = process.env.KEY_ALIAS;
const keyIds = [ process.env.KEY_ARN ];
const keyring = new encryptionSdk.KmsKeyringNode({ generatorKeyId, keyIds })
exports.handler = async (event) => {
  //Decrypt the secret code using encryption SDK.
  let plainTextCode;
  if(event.request.code){
    const { plaintext, messageHeader } = await decrypt(keyring,
      b64.toByteArray(event.request.code));
    plainTextCode = plaintext
  }
  //PlainTextCode now contains the decrypted secret.
  if(event.triggerSource == 'CustomEmailSender_SignUp'){
    //Send an email message to your user via a custom provider.
    //Include the temporary password in the message.
  }
  else if(event.triggerSource == 'CustomEmailSender_ResendCode'){
  }
  else if(event.triggerSource == 'CustomEmailSender_ForgotPassword'){
  }
  else if(event.triggerSource == 'CustomEmailSender_UpdateUserAttribute'){
  }
  else if(event.triggerSource == 'CustomEmailSender_VerifyUserAttribute'){
  }
  else if(event.triggerSource == 'CustomEmailSender_AdminCreateUser'){
  }
  else if(event.triggerSource == 'CustomEmailSender_AccountTakeOverNotification'){
```

```

}
return;
};

```

Sources du déclencheur Lambda expéditeur d'e-mail personnalisé

Le tableau suivant montre les événements de déclenchement pour les sources du déclencheur d'e-mail personnalisé dans votre code Lambda.

TriggerSource value	Événement
CustomEmailSender_SignUp	Un utilisateur s'inscrit et Amazon Cognito envoie un message de bienvenue.
CustomEmailSender_ForgotPassword	Un utilisateur demande un code pour réinitialiser son mot de passe.
CustomEmailSender_ResendCode	Un utilisateur demande un code de remplacement pour réinitialiser son mot de passe.
CustomEmailSender_UpdateUserAttribute	Un utilisateur met à jour un attribut d'adresse e-mail ou de numéro de téléphone et Amazon Cognito envoie un code pour vérifier cet attribut.
CustomEmailSender_VerifyUserAttribute	Un utilisateur crée un nouvel attribut d'adresse e-mail ou de numéro de téléphone et Amazon Cognito envoie un code pour vérifier cet attribut.
CustomEmailSender_AdminCreateUser	Vous créez un nouvel utilisateur dans votre groupe d'utilisateurs et Amazon Cognito lui envoie un mot de passe temporaire.
CustomEmailSender_AccountTakeOverNotification	Amazon Cognito détecte une tentative de prise de contrôle d'un compte d'utilisateur et envoie une notification à l'utilisateur.

Déclencheur Lambda de l'expéditeur de SMS personnalisé

Lorsque vous attribuez un déclencheur d'expéditeur de SMS personnalisé à votre groupe d'utilisateurs, Amazon Cognito invoque une fonction Lambda à la place de son comportement par défaut lorsqu'un événement utilisateur nécessite l'envoi d'un SMS. Grâce à un déclencheur d'expéditeur personnalisé, votre AWS Lambda fonction peut envoyer des notifications par SMS à vos utilisateurs par le biais d'une méthode et d'un fournisseur de votre choix. Le code personnalisé de votre fonction doit traiter et livrer tous les SMS de votre groupe d'utilisateurs.

Note

Actuellement, vous ne pouvez pas attribuer de déclencheurs d'expéditeur personnalisé dans la console Amazon Cognito. Vous pouvez attribuer un déclencheur à l'aide du paramètre `LambdaConfig` dans une demande d'API `CreateUserPool` ou `UpdateUserPool`.

Pour configurer ce déclencheur, effectuez les opérations suivantes :

1. Créez une [clé de chiffrement symétrique](#) dans AWS Key Management Service (AWS KMS). Amazon Cognito génère des secrets (mots de passe, codes de vérification et codes de confirmation temporaires), puis utilise cette clé KMS pour chiffrer les secrets. Vous pouvez ensuite utiliser l'opération API [Decrypt](#) dans votre fonction Lambda pour déchiffrer les secrets et les envoyer à l'utilisateur en texte brut. [AWS Encryption SDK](#) est un outil utile pour les AWS KMS opérations de votre fonction.
2. Créez une fonction Lambda à attribuer en tant que déclencheur d'expéditeur personnalisé. Accordez des autorisations `kms:Decrypt` au rôle de la fonction Lambda pour votre clé KMS.
3. Accordez l'accès au principal de service Amazon Cognito `cognito-idp.amazonaws.com` pour appeler la fonction Lambda.
4. Écrivez un code de fonction Lambda qui dirige vos messages vers des méthodes de remise personnalisées ou des fournisseurs tiers. Pour fournir le code de vérification ou de confirmation de votre utilisateur, décidez et déchiffrez en Base64 la valeur du paramètre `code` dans la demande. Cette opération produit un code ou un mot de passe en texte brut que vous devez inclure dans votre message.
5. Mettez à jour le groupe d'utilisateurs pour qu'il utilise un déclencheur Lambda Expéditeur personnalisé. Le principal IAM qui met à jour ou crée un groupe d'utilisateurs avec un déclencheur d'expéditeur personnalisé doit être autorisé à créer un octroi pour votre clé KMS. L'extrait de code `LambdaConfig` suivant attribue des fonctions personnalisées d'expéditeur de SMS et d'e-mails.

```
"LambdaConfig": {
  "KMSKeyID": "arn:aws:kms:us-east-1:123456789012:key/a6c4f8e2-0c45-47db-925f-87854bc9e357",
  "CustomEmailSender": {
    "LambdaArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
    "LambdaVersion": "V1_0"
  },
  "CustomSMSSender": {
    "LambdaArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
    "LambdaVersion": "V1_0"
  }
}
```

Paramètres de déclencheur Lambda d'expéditeur de SMS personnalisé

La demande qu'Amazon Cognito transmet à cette fonction Lambda est une combinaison des paramètres ci-dessous et des [paramètres courants](#) qu'Amazon Cognito ajoute à toutes les demandes.

JSON

```
{
  "request": {
    "type": "customSMSSenderRequestV1",
    "code": "string",
    "clientMetadata": {
      "string": "string",
      . . .
    },
    "userAttributes": {
      "string": "string",
      . . .
    }
  }
}
```

Paramètres de demande d'expéditeur de SMS personnalisé

type

Version de la demande. Pour un événement d'expéditeur de SMS personnalisé, la valeur de cette chaîne est toujours `customSMSSenderRequestV1`.

code

Code chiffré que votre fonction peut déchiffrer et envoyer à votre utilisateur.

clientMetadata

Une ou plusieurs paires clé-valeur que vous pouvez fournir en tant qu'entrée personnalisée au déclencheur de la fonction Lambda d'expéditeur de SMS personnalisé. Pour transmettre ces données à votre fonction Lambda, vous pouvez utiliser le ClientMetadata paramètre dans les actions [AdminRespondToAuthChallenge](#) et [RespondToAuthChallenge](#) API. Amazon Cognito n'inclut pas les données issues du ClientMetadata paramètre [AdminInitiateAuth](#) et des opérations d'[InitiateAuth](#) API dans la demande transmise à la fonction de post-authentification.

userAttributes

Une ou plusieurs paires clé-valeur qui représentent les attributs utilisateur.

Paramètres de réponse d'expéditeur de SMS personnalisé

Amazon Cognito n'attend aucune information de retour supplémentaire dans la réponse. Votre fonction peut utiliser les opérations d'API pour interroger et modifier vos ressources, ou enregistrer des métadonnées d'événements dans un système externe.

Activation du déclencheur Lambda d'expéditeur de SMS personnalisé

Vous pouvez configurer un déclencheur d'expéditeur de SMS personnalisé qui utilise une logique personnalisée pour l'envoi des SMS de votre groupe d'utilisateurs. La procédure suivante attribue un déclencheur de SMS personnalisé, un déclencheur d'e-mail personnalisé, ou les deux, à votre groupe d'utilisateurs. Une fois que vous avez ajouté votre déclencheur d'expéditeur de SMS personnalisé, Amazon Cognito envoie toujours les attributs utilisateur, dont le numéro de téléphone et le code à usage unique à votre fonction Lambda, au lieu d'envoyer un message SMS avec Amazon Simple Notification Service (comportement par défaut).

Important

Amazon Cognito échappe en HTML les caractères réservés tels que `<` (<) et `>` (>) dans le mot de passe temporaire de votre utilisateur. Ces caractères peuvent apparaître dans les mots de passe temporaires qu'Amazon Cognito envoie à votre fonction d'expéditeur d'e-mail personnalisé, mais ils n'apparaissent pas dans les codes de vérification temporaires. Pour envoyer des mots de passe temporaires, votre fonction Lambda doit

annuler l'échappement de ces caractères après avoir déchiffré le mot de passe et avant d'envoyer le message à votre utilisateur.

1. Créez une clé de chiffrement dans AWS KMS. Cette clé chiffre les mots de passe temporaires et les codes d'autorisation générés par Amazon Cognito. Vous pouvez ensuite déchiffrer ces secrets dans la fonction Lambda d'expéditeur personnalisé et les envoyer à l'utilisateur en texte brut.
2. Accordez au principal du service Amazon Cognito un accès `cognito-idp.amazonaws.com` pour chiffrer les codes à l'aide de la clé KMS.

Appliquez à votre clé KMS la stratégie basée sur les ressources suivante.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "cognito-idp.amazonaws.com"
    },
    "Action": "kms:CreateGrant",
    "Resource": "arn:aws:kms:us-
west-2:111222333444:key/1example-2222-3333-4444-999example",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111222333444"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:cognito-idp:us-
west-2:111222333444:userpool/us-east-1_EXAMPLE"
      }
    }
  }]
}
```

3. Créez une fonction Lambda pour le déclencheur d'expéditeur personnalisé. Amazon Cognito utilise le [kit SDK de chiffrement AWS](#) pour chiffrer les secrets, les mots de passe et les codes temporaires qui autorisent les demandes d'API de vos utilisateurs.
 - Attribuez un rôle IAM à votre fonction Lambda qui, au minimum, dispose d'autorisations `kms:Decrypt` pour votre clé KMS.

4. Accordez l'accès au principal de service Amazon Cognito `cognito-idp.amazonaws.com` pour appeler la fonction Lambda.

La AWS CLI commande suivante autorise Amazon Cognito à appeler votre fonction Lambda :

```
aws lambda add-permission --function-name lambda_arn --statement-id
"CognitoLambdaInvokeAccess" --action lambda:InvokeFunction --principal cognito-
idp.amazonaws.com
```

5. Composez le code de votre fonction Lambda pour qu'elle envoie vos messages. Amazon Cognito crypte les secrets AWS Encryption SDK avant qu'Amazon Cognito ne les envoie à la fonction Lambda d'expéditeur personnalisée. Dans votre fonction, déchiffrez le secret et traitez les métadonnées pertinentes. Envoyez ensuite le code, votre propre message personnalisé ainsi que le numéro de téléphone de destination à l'API personnalisée qui remet votre message.
6. Ajoutez le AWS Encryption SDK à votre fonction Lambda. Pour en savoir plus, consultez [Langages de programmation du kit SDK de chiffrement AWS](#). Pour mettre à jour le package Lambda, effectuez les étapes suivantes.
 - a. Exportez votre fonction Lambda sous forme de fichier `.zip` dans la AWS Management Console.
 - b. Ouvrez votre fonction et ajoutez le AWS Encryption SDK. Pour obtenir des informations supplémentaires et les liens de téléchargement, consultez [Langages de programmation du AWS Encryption SDK](#) dans le Guide du développeur AWS Encryption SDK .
 - c. Zippez votre fonction avec les dépendances du kit SDK et chargez-la sur Lambda. Pour en savoir plus, consultez [Déploiement de fonctions Lambda sous forme d'archives de fichiers .zip](#) dans le Guide du développeur AWS Lambda .
7. Mettez à jour votre groupe d'utilisateurs pour ajouter des déclencheurs Lambda d'expéditeur personnalisé. Incluez un paramètre `CustomSMSSender` ou `CustomEmailSender` dans une demande d'API `UpdateUserPool`. L'opération d'API `UpdateUserPool` a besoin de tous les paramètres de votre groupe d'utilisateurs et des paramètres que vous voulez modifier. Si vous ne fournissez pas tous les paramètres pertinents, Amazon Cognito définit les valeurs de tous les paramètres manquants sur leurs valeurs par défaut. Comme dans l'exemple suivant, incluez des entrées pour toutes les fonctions Lambda que vous souhaitez ajouter ou conserver dans votre groupe d'utilisateurs. Pour de plus amples informations, veuillez consulter [Mise à jour de la configuration d'un groupe d'utilisateurs](#).

```
#Send this parameter in an 'aws cognito-idp update-user-pool' CLI command,
including any existing
#user pool configurations.

--lambda-config "PreSignUp=lambda-arn, \
                 CustomSMSSender={LambdaVersion=V1_0,LambdaArn=lambda-arn}, \
                 CustomEmailSender={LambdaVersion=V1_0,LambdaArn=lambda-arn},
\
                 KMSKeyID=key-id"
```

Pour supprimer un déclencheur Lambda d'expéditeur personnalisé avec un `update-user-pool` AWS CLI, omettez le `CustomEmailSender` paramètre `CustomSMSSender` or et incluez tous les autres déclencheurs que vous souhaitez utiliser avec votre groupe d'utilisateurs. `--lambda-config`

Pour supprimer un déclencheur Lambda d'expéditeur personnalisé avec une demande d'API `UpdateUserPool`, omettez le paramètre `CustomSMSSender` ou `CustomEmailSender` dans corps de la demande qui contient le reste de la configuration de votre groupe d'utilisateurs.

Exemple de code

L'exemple Node.js suivant traite un événement de SMS dans votre fonction Lambda d'expéditeur de SMS personnalisé. Cet exemple suppose que votre fonction possède deux variables d'environnement définies.

KEY_ALIAS

L'[alias](#) de la clé KMS que vous souhaitez utiliser pour chiffrer et déchiffrer les codes de vos utilisateurs.

KEY_ARN

L'Amazon Resource Name (ARN) de la clé KMS que vous souhaitez utiliser pour chiffrer et déchiffrer les codes de vos utilisateurs.

```
const AWS = require('aws-sdk');
const b64 = require('base64-js');
const encryptionSdk = require('@aws-crypto/client-node');
```

```
//Configure the encryption SDK client with the KMS key from the environment variables.

const { encrypt, decrypt } =
  encryptionSdk.buildClient(encryptionSdk.CommitmentPolicy.REQUIRE_ENCRYPT_ALLOW_DECRYPT);
const generatorKeyId = process.env.KEY_ALIAS;
const keyIds = [ process.env.KEY_ARN ];
const keyring = new encryptionSdk.KmsKeyringNode({ generatorKeyId, keyIds })
exports.handler = async (event) => {
  //Decrypt the secret code using encryption SDK.
  let plainTextCode;
  if(event.request.code){
    const { plaintext, messageHeader } = await decrypt(keyring,
      b64.toByteArray(event.request.code));
    plainTextCode = plaintext
  }
  //PlainTextCode now contains the decrypted secret.
  if(event.triggerSource == 'CustomSMSSender_SignUp'){
    //Send an SMS message to your user via a custom provider.
    //Include the temporary password in the message.
  }
  else if(event.triggerSource == 'CustomSMSSender_ResendCode'){
  }
  else if(event.triggerSource == 'CustomSMSSender_ForgotPassword'){
  }
  else if(event.triggerSource == 'CustomSMSSender_UpdateUserAttribute'){
  }
  else if(event.triggerSource == 'CustomSMSSender_VerifyUserAttribute'){
  }
  else if(event.triggerSource == 'CustomSMSSender_AdminCreateUser'){
  }
  else if(event.triggerSource == 'CustomSMSSender_AccountTakeOverNotification'){
  }
  return;
};
```

Rubriques

- [Évaluer les fonctionnalités liées aux SMS avec une fonction d'expéditeur de SMS personnalisé](#)
- [Sources du déclencheur Lambda Expéditeur de SMS personnalisé](#)

Évaluer les fonctionnalités liées aux SMS avec une fonction d'expéditeur de SMS personnalisé

Une fonction Lambda d'expéditeur de SMS personnalisé accepte les SMS que votre groupe d'utilisateurs envoie, et la fonction effectue la remise de ce contenu en fonction de votre logique personnalisée. Amazon Cognito envoie les [Paramètres de déclencheur Lambda d'expéditeur de SMS personnalisé](#) à votre fonction. Votre fonction peut traiter ces informations comme vous le souhaitez. Par exemple, vous pouvez envoyer le code à une rubrique Amazon Simple Notification Service (Amazon SNS). L'abonné d'une rubrique Amazon SNS peut être un SMS, un point de terminaison HTTPS ou une adresse e-mail.

[Pour créer un environnement de test pour la messagerie SMS Amazon Cognito avec une fonction Lambda d'expéditeur de SMS personnalisée, consultez `amazon-cognito-user-pool-development-and-testing-with - sms-redirected-to-email` dans la bibliothèque `aws-samples` sur \[GitHub\]\(#\)](#) Le référentiel contient des AWS CloudFormation modèles qui peuvent créer un nouveau groupe d'utilisateurs ou fonctionner avec un groupe d'utilisateurs que vous possédez déjà. Ces modèles créent des fonctions Lambda et une rubrique Amazon SNS. La fonction Lambda que le modèle attribue en tant que déclencheur d'expéditeur de SMS personnalisé redirige vos SMS aux abonnés vers la rubrique Amazon SNS.

Quand vous déployez cette solution dans un groupe d'utilisateurs, tous les messages qu'Amazon Cognito envoie généralement par SMS sont envoyés à la place par la fonction Lambda à une adresse e-mail centrale. Utilisez cette solution pour personnaliser et prévisualiser les SMS, et pour tester les événements de groupe d'utilisateurs qui provoquent l'envoi d'un SMS par Amazon Cognito. Une fois vos tests terminés, annulez la CloudFormation pile ou supprimez l'attribution personnalisée de la fonction d'expéditeur de SMS de votre groupe d'utilisateurs.

Important

N'utilisez pas les modèles contenus dans [amazon-cognito-user-pool-development-and-testing-with - sms-redirected-to-email](#) pour créer un environnement de production. La fonction Lambda d'expéditeur de SMS personnalisé figurant dans la solution simule des SMS, mais les envoie tous à une adresse e-mail centrale unique. Avant de pouvoir envoyer des SMS dans un groupe d'utilisateurs Amazon Cognito de production, vous devez remplir les exigences spécifiées dans [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#).

Sources du déclencheur Lambda Expéditeur de SMS personnalisé

Le tableau suivant montre l'événement de déclenchement pour les sources du déclencheur de SMS personnalisé dans votre code Lambda.

TriggerSource value	Événement
CustomSMSSender_SignUp	Un utilisateur s'inscrit et Amazon Cognito envoie un message de bienvenue.
CustomSMSSender_ForgotPassword	Un utilisateur demande un code pour réinitialiser son mot de passe.
CustomSMSSender_ResendCode	Un utilisateur demande un nouveau code pour confirmer son inscription.
CustomSMSSender_VerifyUserAttribute	Un utilisateur crée un nouvel attribut d'adresse e-mail ou de numéro de téléphone et Amazon Cognito envoie un code pour vérifier cet attribut.
CustomSMSSender_UpdateUserAttribute	Un utilisateur met à jour un attribut d'adresse e-mail ou de numéro de téléphone et Amazon Cognito envoie un code pour vérifier cet attribut.
CustomSMSSender_Authentication	Un utilisateur configuré avec l'authentification multifacteur (MFA) par SMS se connecte.
CustomSMSSender_AdminCreateUser	Vous créez un nouvel utilisateur dans votre groupe d'utilisateurs et Amazon Cognito lui envoie un mot de passe temporaire.

Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito.

Les groupes d'utilisateurs Amazon Cognito sont intégrés à Amazon Pinpoint afin de fournir une analytique pour les groupes d'utilisateurs Amazon Cognito et d'enrichir les données utilisateur pour

les campagnes Amazon Pinpoint. Amazon Pinpoint fournit une analytique et des campagnes ciblées pour susciter l'implication des utilisateurs dans les applications mobiles à l'aide de notifications push. Avec la prise en charge de l'analytique Amazon Pinpoint dans les groupes d'utilisateurs Amazon Cognito, vous pouvez suivre les inscriptions de groupes d'utilisateurs, les connexions, les échecs d'authentification, les utilisateurs actifs quotidiens et les utilisateurs actifs mensuels dans la console Amazon Pinpoint. Vous pouvez explorer les données pour différentes plages de dates ou divers attributs, comme la plateforme, les paramètres régionaux des appareils et la version des applications.

Vous pouvez également configurer des attributs personnalisés pour votre application. Ceux-ci peuvent ensuite être utilisés pour segmenter vos utilisateurs sur Amazon Pinpoint et leur envoyer des notifications push ciblées. Si vous choisissez `Share user attribute data with Amazon Pinpoint` (Partager les données d'attribut utilisateur avec Amazon Pinpoint) dans l'onglet Analytics (Analytique) de la console Amazon Cognito, Amazon Pinpoint crée des points de terminaison supplémentaires pour les adresses e-mail et les numéros de téléphone des utilisateurs.

Lorsque vous activez l'analytique Amazon Pinpoint dans votre groupe d'utilisateurs à l'aide de la console Amazon Cognito, vous créez également un [rôle lié à un service](#) qu'Amazon Cognito endosse lorsqu'il envoie une demande d'API à Amazon Pinpoint pour votre groupe d'utilisateurs. Le principal IAM qui ajoute votre configuration d'analyse doit disposer des autorisations [CreateServiceLinkedRole](#). Le rôle lié à un service est [AWSServiceRoleForAmazonCognitoIdp](#). Pour de plus amples informations, veuillez consulter [Utilisation de rôles liés à un service pour Amazon Cognito](#).

Lorsque vous appliquez un `AnalyticsConfiguration` à votre client d'application dans l'API Amazon Cognito, vous pouvez attribuer un rôle IAM personnalisé pour Amazon Pinpoint et un ID externe pour endosser le rôle. Le rôle doit faire confiance au principal de service `cognito-idp`, et si la politique d'approbation de rôle nécessite un ID externe, celui-ci doit correspondre à votre `AnalyticsConfiguration`. Vous devez accorder les autorisations `cognito-idp:Describe*` de rôle, ainsi que les autorisations suivantes pour votre projet Amazon Pinpoint.

- `mobiletargeting:UpdateEndpoint`
- `mobiletargeting:PutEvents`

Disponibilité des régions Amazon Cognito et Amazon Pinpoint

Le tableau suivant présente les mappages Région AWS entre Amazon Cognito et Amazon Pinpoint qui répondent à l'une des conditions suivantes.

- Vous ne pouvez utiliser un projet Amazon Pinpoint que dans la région USA Est (Virginie du Nord) (us-east-1).
- Vous pouvez utiliser un projet Amazon Pinpoint dans la même région ou dans la région USA Est (Virginie du Nord) (us-east-1).

Par défaut, Amazon Cognito ne peut envoyer des données analytiques qu'à un projet Amazon Pinpoint dans la même Région AWS. Les exceptions à cette règle sont les régions figurant dans le tableau suivant et celles où Amazon Pinpoint n'est pas disponible.

Amazon Pinpoint n'est pas disponible dans les régions suivantes : Les groupes d'utilisateurs Amazon Cognito de ces régions ne prennent pas en charge l'analytique.

- Europe (Milan)
- Moyen-Orient (Bahreïn)
- Asie-Pacifique (Osaka)
- Israël (Tel Aviv)
- Afrique (Le Cap)
- Asie-Pacifique (Jakarta)

Le tableau montre la relation entre la région dans laquelle vous avez créé votre groupe d'utilisateurs Amazon Cognito et la région correspondante dans Amazon Pinpoint. Vous devez configurer votre projet Amazon Pinpoint dans une région disponible pour l'intégrer à Amazon Cognito.

Région du groupe d'utilisateurs Amazon Cognito	Régions du projet Amazon Pinpoint
ap-northeast-1	us-east-1
ap-northeast-2	us-east-1
ap-south-1	us-east-1, ap-south-1
ap-southeast-1	us-east-1
ap-southeast-2	us-east-1, ap-southeast-2
ca-central-1	us-east-1

Région du groupe d'utilisateurs Amazon Cognito	Régions du projet Amazon Pinpoint
eu-central-1	us-east-1, eu-central-1
eu-west-1	us-east-1, eu-west-1
eu-west-2	us-east-1
us-east-1	us-east-1
us-east-2	us-east-1
us-west-2	us-east-1, us-west-2

Exemples de correspondances entre régions

- Si vous créez un groupe d'utilisateurs dans la région ap-northeast-1, vous pouvez créer votre projet Amazon Pinpoint dans la région us-east-1.
- Si vous créez un groupe d'utilisateurs dans la région ap-south-1, vous pouvez créer votre projet Amazon Pinpoint dans la région us-east-1 ou ap-south-1.

Note

Pour toutes les régions Régions AWS à l'exception de celles figurant dans le tableau précédent, Amazon Cognito peut uniquement utiliser un projet Amazon Pinpoint dans la même région que votre groupe d'utilisateurs. Si Amazon Pinpoint n'est pas disponible dans la région où vous avez créé votre groupe d'utilisateurs et qu'elle ne figure pas dans le tableau, Amazon Cognito ne prend pas en charge l'analytique Amazon Pinpoint dans cette région. Pour obtenir des informations détaillées sur la Région AWS, consultez [Points de terminaison et quotas Amazon Pinpoint](#).

Spécification des paramètres d'analytique Amazon Pinpoint (AWS Management Console)

Vous pouvez configurer votre groupe d'utilisateurs Amazon Cognito pour envoyer des données analytiques à Amazon Pinpoint. Amazon Cognito envoie des données analytiques à Amazon Pinpoint uniquement pour les utilisateurs locaux. Après avoir configuré votre groupe d'utilisateurs pour l'associer à un projet Amazon Pinpoint, vous devez inclure `AnalyticsMetadata` dans vos demandes d'API. Pour de plus amples informations, veuillez consulter [Intégration de votre application à Amazon Pinpoint](#).

Pour spécifier les paramètres d'analyse

1. Accédez à la [console Amazon Cognito](#). Il se peut que vous soyez invité à saisir vos informations d'identification AWS.
2. Sélectionnez User Pools (Groupes d'utilisateurs) et choisissez un groupe d'utilisateurs existant dans la liste.
3. Choisissez l'onglet App integration (Intégration d'applications).
4. Sous App clients and analytics (Clients d'application et analytique), choisissez un nom de client d'application existant dans la liste.
5. Sous Pinpoint analytics (Analytique Pinpoint), choisissez Enable (Activer).
6. Choisissez un élément dans Pinpoint Region (Région Pinpoint).
7. Choisissez un élément dans Amazon Pinpoint project (Projet Amazon Pinpoint) ou sélectionnez Create Amazon Pinpoint project (Créer un projet Amazon Pinpoint).


Note

L'ID de projet Amazon Pinpoint est une chaîne de 32 caractères propre à votre projet Amazon Pinpoint. Il s'affiche dans la console Amazon Pinpoint.

Vous pouvez mapper plusieurs applications Amazon Cognito à un seul projet Amazon Pinpoint. Cependant, chaque application Amazon Cognito ne peut être mappée qu'à un seul projet Amazon Pinpoint.

Dans Amazon Pinpoint, chaque projet doit être une seule application. Par exemple, si un développeur de jeux possède deux jeux, chaque d'eux doit être un projet Amazon Pinpoint distinct, même si les deux utilisent le même groupe d'utilisateurs Amazon Cognito. Pour plus d'informations sur les projets Pinpoint, consultez [Créer un projet dans Amazon Pinpoint](#).

8. Sous User data sharing (Partage de données utilisateur), choisissez Share user data with Amazon Pinpoint (Partager des données utilisateur avec Amazon Pinpoint) si vous souhaitez qu'Amazon Cognito envoie les adresses e-mail et les numéros de téléphone à Amazon Pinpoint et crée des points de terminaison supplémentaires pour les utilisateurs. Une fois le numéro de téléphone et l'adresse e-mail de vos utilisateurs vérifiés, Amazon Cognito les partage uniquement avec Amazon Pinpoint s'ils sont accessibles au compte utilisateur.

 Note

Un point de terminaison identifie de façon unique un appareil d'utilisateur auquel vous pouvez envoyer des notifications push avec Amazon Pinpoint. Pour plus d'informations sur les points de terminaison, consultez [Ajout de points de terminaison](#) dans le Guide du développeur Amazon Pinpoint.

9. Choisissez Enregistrer les modifications.

Spécification des paramètres d'analytique Amazon Pinpoint (AWS CLI et API AWS)

Utilisez les commandes suivantes pour spécifier les paramètres d'analytique Amazon Pinpoint pour votre groupe d'utilisateurs.

Pour spécifier les paramètres d'analyse de l'application cliente existante de votre groupe d'utilisateurs au moment de la création de l'application

- AWS CLI: `aws cognito-idp create-user-pool-client`
- API AWS : [CreateUserPoolClient](#)

Pour mettre à jour les paramètres d'analyse de l'application cliente existante de votre groupe d'utilisateurs

- AWS CLI: `aws cognito-idp update-user-pool-client`
- API AWS : [UpdateUserPoolClient](#)

Note

Amazon Cognito prend en charge les intégrations dans la région lorsque vous utilisez `ApplicationArn`

Intégration de votre application à Amazon Pinpoint

Vous pouvez publier des métadonnées analytiques dans Amazon Pinpoint pour des utilisateurs locaux Amazon Cognito dans l'API des groupes d'utilisateurs.

Utilisateurs locaux

Utilisateurs qui se sont inscrits pour obtenir un compte ou qui ont été créés dans votre groupe d'utilisateurs au lieu de se connecter via un fournisseur d'identité (IdP) tiers.

API des groupes d'utilisateurs

Opérations que vous pouvez intégrer avec un kit SDK AWS, à l'aide d'une application dotée d'une interface utilisateur personnalisée. Vous ne pouvez pas transmettre de métadonnées analytiques pour des utilisateurs fédérés ou locaux qui se connectent via l'interface utilisateur hébergée. Pour obtenir la liste des opérations d'API des groupes d'utilisateurs, consultez [Référence d'API Amazon Cognito](#).

Une fois que vous avez configuré votre groupe d'utilisateurs en vue d'une publication dans une campagne, Amazon Cognito transmet des métadonnées à Amazon Pinpoint pour les opérations d'API suivantes.

- `AdminInitiateAuth`
- `AdminRespondToAuthChallenge`
- `ConfirmForgotPassword`
- `ConfirmSignUp`
- `ForgotPassword`
- `InitiateAuth`
- `ResendConfirmationCode`
- `RespondToAuthChallenge`
- `SignUp`

Pour transmettre les métadonnées relatives à votre session utilisateur à votre campagne Amazon Pinpoint, incluez une valeur `AnalyticsEndpointId` dans le paramètre `AnalyticsMetadata` de votre demande d'API. Pour accéder à un exemple de JavaScript, veuillez consulter [Pourquoi les analyses de mon groupe d'utilisateurs Amazon Cognito n'apparaissent-elles pas sur mon tableau de bord Amazon Pinpoint ?](#) dans le Centre de connaissances AWS.

Configuration de l'analytique des groupe d'utilisateurs

Avec l'analytique Amazon Pinpoint, vous pouvez suivre les inscriptions de groupes d'utilisateurs Amazon Cognito, les connexions, les échecs d'authentification, les utilisateurs actifs quotidiennement et les utilisateurs actifs mensuellement. Vous pouvez également configurer des attributs d'utilisateurs spécifiques à votre application avec AWS Mobile SDK for Android ou AWS Mobile SDK for iOS. Ceux-ci peuvent ensuite être utilisés pour segmenter vos utilisateurs dans Amazon Pinpoint, et leur envoyer des notifications push ciblées.

Dans l'onglet Intégration d'application, sous Clients d'application et analytique, vous pouvez accéder à un client d'application existant ou en créer un nouveau. Sous la configuration du client d'application, vous pouvez spécifier un projet Amazon Pinpoint que vous souhaitez utiliser avec votre application. Pour plus d'informations, consultez [Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito](#).

Note

Amazon Pinpoint est disponible dans plusieurs régions AWS en Amérique du Nord, en Europe, en Asie et en Océanie. Les régions Amazon Pinpoint incluent l'API Amazon Pinpoint. Si une région Amazon Pinpoint est prise en charge par Amazon Cognito, Amazon Cognito envoie des événements à des projets Amazon Pinpoint dans la même région Amazon Pinpoint. Si une région n'est pas prise en charge par Amazon Pinpoint, Amazon Cognito prend en charge uniquement l'envoi d'événements dans la région us-east-1. Pour obtenir des informations détaillées sur une région Amazon Pinpoint, consultez [Points de terminaison et quotas Amazon Pinpoint](#) et [Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito](#).

Pour ajouter l'analytique et des campagnes

1. Choisissez Ajouter l'analytique et des campagnes).
2. Choisissez un client d'application Cognito dans la liste.

3. Pour mapper votre application Amazon Cognito à un Projet Amazon Pinpoint, choisissez le projet Amazon Pinpoint dans la liste.


 Note

L'ID de projet Amazon Pinpoint est une chaîne de 32 caractères propre à votre projet Amazon Pinpoint. Il s'affiche dans la console Amazon Pinpoint.

Vous pouvez mapper plusieurs applications Amazon Cognito à un seul projet Amazon Pinpoint. Cependant, chaque application Amazon Cognito ne peut être mappée qu'à un seul projet Amazon Pinpoint.

Dans Amazon Pinpoint, chaque projet doit être une seule application. Par exemple, si un développeur de jeux possède deux jeux, chaque d'eux doit être un projet Amazon Pinpoint distinct, même si les deux utilisent les mêmes groupes d'utilisateurs Amazon Cognito.

4. Choisissez Share user attribute data with Amazon Pinpoint (Partager les données d'attribut utilisateur avec Amazon Pinpoint) si vous souhaitez qu'Amazon Cognito envoie des adresses e-mail et numéros de téléphone à Amazon Pinpoint afin de créer des points de terminaison supplémentaires pour des utilisateurs.

 Note

Un point de terminaison identifie de façon unique un appareil d'utilisateur auquel vous pouvez envoyer des notifications push avec Amazon Pinpoint. Pour plus d'informations sur les points de terminaison, consultez [Ajout de points de terminaison à Amazon Pinpoint](#) dans le Guide du développeur Amazon Pinpoint.

5. Saisissez un rôle IAM que vous avez déjà créé ou choisissez Create new role (Créer un nouveau rôle) pour créer un rôle dans la console IAM.
6. Choisissez Enregistrer les modifications.
7. Pour spécifier d'autres mappages d'application, choisissez Ajouter un mappage d'application.
8. Choisissez Enregistrer les modifications.

Gestion des utilisateurs dans votre groupe d'utilisateurs

Une fois que vous avez créé un groupe d'utilisateurs, vous pouvez créer, confirmer et gérer des comptes utilisateur. Avec les groupes d'utilisateurs Amazon Cognito, vous pouvez gérer vos utilisateurs et leur accès aux ressources en mappant des rôles IAM à des groupes.

Vous pouvez importer vos utilisateurs dans un groupe d'utilisateurs avec un déclencheur Lambda Migration d'utilisateur. Cette approche permet de migrer en toute transparence des utilisateurs de votre répertoire utilisateur existant aux groupes d'utilisateurs lorsqu'ils se connectent à votre groupe d'utilisateurs pour la première fois.

Rubriques

- [Configuration de politiques pour la création d'utilisateurs](#)
- [Inscription et confirmation des comptes d'utilisateur](#)
- [Création de comptes d'utilisateur en tant qu'administrateur](#)
- [Ajout de groupes à un groupe d'utilisateurs](#)
- [Gestion et recherche de comptes d'utilisateur](#)
- [Récupération de comptes d'utilisateurs](#)
- [Importation d'utilisateurs dans un groupe d'utilisateurs](#)
- [Attributs de groupe d'utilisateurs](#)
- [Ajout d'exigences de mot de passe pour un groupe d'utilisateurs](#)

Configuration de politiques pour la création d'utilisateurs

Votre groupe d'utilisateurs peut autoriser les utilisateurs à s'inscrire, ou vous pouvez les créer en tant qu'administrateur. Vous pouvez également contrôler la part du processus de vérification et de confirmation qui incombe à vos utilisateurs après l'inscription. Vous pouvez par exemple vérifier les inscriptions et les accepter sur selon un processus de validation externe. Cette configuration, ou la politique de création d'utilisateurs pour les administrateurs, définit également le délai avant qu'un utilisateur ne puisse plus confirmer son compte utilisateur.

Amazon Cognito peut répondre aux besoins de vos clients publics en tant que plateforme de gestion de l'identité et de l'accès des clients (CIAM) pour votre logiciel. Un groupe d'utilisateurs qui accepte l'inscription et possède un client d'application, avec ou sans interface utilisateur hébergée, crée un profil utilisateur pour toute personne sur Internet qui connaît votre identifiant client d'application accessible au public et qui demande à s'inscrire. Un profil utilisateur enregistré peut recevoir des

jetons d'accès et d'identité et peut accéder aux ressources que vous avez autorisées pour votre application. Avant d'activer l'inscription dans votre groupe d'utilisateurs, passez en revue vos options et assurez-vous que votre configuration est conforme à vos normes de sécurité. Définissez avec soin Activer l'auto-inscription et `AllowAdminCreateUserOnly`, comme décrit dans les procédures suivantes.

AWS Management Console

L'onglet Expérience d'inscription de votre groupe d'utilisateurs et l'étape Configurer l'expérience d'inscription de l'assistant de création de groupe d'utilisateurs contiennent certains des paramètres d'inscription et de création administrative des utilisateurs de votre groupe d'utilisateurs.

Pour configurer l'expérience d'inscription

1. Dans Vérification et confirmation assistées par Cognito, indiquez si vous souhaitez Autoriser Cognito à envoyer automatiquement des messages pour vérifier et confirmer. Lorsque ce paramètre est activé, Amazon Cognito envoie un e-mail ou un SMS aux nouveaux utilisateurs avec un code qu'ils doivent présenter à votre groupe d'utilisateurs. Cela confirme qu'ils sont propriétaires de l'adresse e-mail ou du numéro de téléphone, en définissant l'attribut équivalent comme vérifié et en confirmant le compte utilisateur pour la connexion. Les Attributs à vérifier que vous choisissez déterminent les méthodes de livraison et les destinations des messages de vérification.
2. La vérification des modifications d'attributs n'est pas importante lorsque vous créez des utilisateurs, mais concerne la vérification des attributs. Vous pouvez autoriser les utilisateurs qui ont modifié leurs [attributs de connexion](#), mais qui ne les ont pas encore vérifiés, à poursuivre la connexion avec leur nouvelle valeur d'attribut ou avec leur valeur d'origine. Pour de plus amples informations, veuillez consulter [Vérification en cas de modification de l'adresse e-mail ou du numéro de téléphone par l'utilisateur](#).
3. Les attributs obligatoires affichent les attributs pour lesquels une valeur doit être fournie avant qu'un utilisateur ne puisse s'inscrire ou avant que vous ne puissiez créer un utilisateur. Vous ne pouvez définir les attributs requis que dans l'assistant de création d'un groupe d'utilisateurs.
4. Les attributs personnalisés sont importants pour le processus de création et d'inscription des utilisateurs, car vous ne pouvez définir une valeur pour les attributs personnalisés immuables que lorsque vous créez un utilisateur pour la première fois. Pour plus d'informations sur les attributs personnalisés, consultez [Attributs personnalisés](#).

5. Dans l'onglet Inscription en libre-service, sélectionnez Activer l'auto-inscription si vous souhaitez que les utilisateurs puissent générer un nouveau compte à l'aide de l'API [SignUp non authentifiée](#). Si vous désactivez l'auto-inscription, vous ne pouvez créer de nouveaux utilisateurs qu'en tant qu'administrateur, sur la console Amazon Cognito ou avec des demandes d'API [AdminCreateUser](#). Dans un groupe d'utilisateurs où l'auto-inscription est inactive, les demandes de l'API [SignUp](#) renvoient `NotAuthorizedException` et l'interface utilisateur hébergée n'affiche aucun lien S'inscrire.

Pour les groupes d'utilisateurs dans lesquels vous prévoyez de créer des utilisateurs en tant qu'administrateur, vous pouvez configurer la durée de leurs mots de passe temporaires dans l'onglet Expérience de connexion, sous Les mots de passe temporaires définis par les administrateurs expirent dans.

Le message d'invitation est un autre élément important de la création d'utilisateurs en tant qu'administrateur. Lorsque vous créez un utilisateur, Amazon Cognito lui envoie un message contenant un lien vers votre application afin qu'il puisse se connecter pour la première fois. Personnalisez ce modèle de message dans l'onglet Messagerie sous Modèles de message.

Vous pouvez configurer des [clients d'application confidentiels](#), généralement des applications Web, avec un secret client qui empêche l'inscription sans le secret du client d'application. Selon une bonne pratique de sécurité, ne diffusez pas les secrets des clients d'applications dans des clients d'applications publics, généralement des applications mobiles. Vous pouvez créer des clients d'applications avec des secrets clients dans l'onglet Intégration des applications de la console Amazon Cognito.

Amazon Cognito user pools API

Vous pouvez définir par programmation les paramètres de création d'utilisateurs dans un groupe d'utilisateurs dans une demande d'API [CreateUserPool](#) ou [UpdateUserPool](#).

L'élément [AdminCreateUserConfig](#) définit les valeurs des propriétés suivantes d'un groupe d'utilisateurs.

1. Activer l'inscription en libre-service
2. Le message d'invitation que vous envoyez aux nouveaux utilisateurs créés par l'administrateur

L'exemple suivant, lorsqu'il est ajouté au corps complet d'une demande d'API, définit un groupe d'utilisateurs avec une inscription en libre-service inactive et un e-mail d'invitation de base.


```
"AdminCreateUserConfig": {
  "AllowAdminCreateUserOnly": true,
  "InviteMessageTemplate": {
    "EmailMessage": "Your username is {username} and temporary password is
{#####}.",
    "EmailSubject": "Welcome to ExampleApp",
    "SMSMessage": "Your username is {username} and temporary password is
{#####}."
  }
}
```

Les paramètres supplémentaires suivants d'une demande d'API [CreateUserPool](#) ou [UpdateUserPool](#) régissent la création de nouveaux utilisateurs.

[AutoVerifiedAttributes](#)

Les attributs, adresses e-mail ou numéros de téléphone auxquels vous souhaitez [envoyer automatiquement un message](#) lorsque vous enregistrez un nouvel utilisateur.

[Stratégies](#)

La [politique de mot de passe](#) du groupe d'utilisateurs.

[Schema](#) (Schéma)

Les [attributs personnalisés](#) du groupe d'utilisateurs. Ils sont importants pour le processus de création et d'inscription des utilisateurs, car vous ne pouvez définir une valeur pour les attributs personnalisés immuables que lorsque vous créez un utilisateur pour la première fois.

Ce paramètre définit également les attributs requis pour votre groupe d'utilisateurs. Le texte suivant, lorsqu'il est inséré dans l'élément Schema du corps complet d'une demande d'API, définit l'attribut email de la manière requise.

```
{
  "Name": "email",
  "Required": true
}
```

Inscription et confirmation des comptes d'utilisateur

Des comptes d'utilisateur sont ajoutés à votre groupe d'utilisateurs suite aux opérations suivantes :

- L'utilisateur s'inscrit dans l'application client de votre groupe d'utilisateurs. Il peut s'agir d'une application web ou mobile.
- Vous pouvez importer le compte d'utilisateur dans votre groupe d'utilisateurs. Pour plus d'informations, consultez [Importation d'utilisateurs dans des groupes d'utilisateurs depuis un fichier CSV](#).
- Vous pouvez créer un compte d'utilisateur dans votre groupe d'utilisateurs et inviter cet utilisateur à se connecter. Pour plus d'informations, consultez [Création de comptes d'utilisateur en tant qu'administrateur](#).

Les utilisateurs qui s'inscrivent eux-mêmes doivent être confirmés afin de pouvoir se connecter. Les utilisateurs importés et créés sont déjà confirmés, mais ils doivent créer leur mot de passe la première fois qu'ils se connectent. Les sections suivantes expliquent la procédure de confirmation, ainsi que le mode de vérification par téléphone et par e-mail.

Présentation de la procédure de confirmation d'un compte d'utilisateur

Le schéma suivant illustre la procédure de confirmation :

Si le compte d'utilisateur a été confirmé par l'administrateur ou un déclencheur Lambda avant l'inscription, il se peut qu'aucune adresse e-mail ni aucun numéro de téléphone vérifiés ne soient associés au compte.

Réinitialisation du mot de passe requise

Le compte d'utilisateur est confirmé, mais l'utilisateur doit demander un code et réinitialiser son mot de passe afin de pouvoir se connecter.

Au départ, les comptes d'utilisateur qui sont importés par un administrateur ou un développeur affichent cet état.

Modification forcée du mot de passe

Le compte d'utilisateur est confirmé et l'utilisateur peut se connecter à l'aide d'un mot de passe temporaire. Toutefois, à la première connexion et avant toute autre opération, l'utilisateur doit changer son mot de passe en indiquant une nouvelle valeur.

Au départ, les comptes d'utilisateur créés par un administrateur ou un développeur affichent cet état.

Désactivées

Avant de pouvoir supprimer un compte d'utilisateur, vous devez désactiver l'accès à la connexion pour cet utilisateur.

Vérification des coordonnées à l'inscription

Lorsque de nouveaux utilisateurs s'inscrivent à votre application, il est probable que vous voudrez qu'ils fournissent au moins une méthode de contact. Par exemple, avec les coordonnées de vos utilisateurs, vous pouvez :

- Envoyer un mot de passe temporaire lorsqu'un utilisateur choisit de réinitialiser son mot de passe.
- Informer les utilisateurs lorsque leurs informations personnelles ou financières sont mises à jour.
- Envoyer des messages promotionnels, tels que des offres spéciales ou des remises.
- Envoyer des récapitulatifs de compte ou de rappels de facturation.

Pour de tels cas d'utilisation, il est important que vous adressiez vos messages à une destination vérifiée. Sinon, vous risquez d'envoyer vos messages à une adresse e-mail ou un numéro de téléphone non valide qui a été saisi de façon incorrecte. Ou pire, vous risquez d'envoyer des informations sensibles à des personnes malveillantes qui se font passer pour vos utilisateurs.

Pour vous assurer d'envoyer les messages uniquement aux bonnes personnes, configurez votre groupe d'utilisateurs Amazon Cognito afin que les utilisateurs fournissent les informations suivantes lors de l'inscription :

- a. Adresse e-mail ou numéro de téléphone.
- b. Code de vérification envoyé par Amazon Cognito à l'adresse e-mail ou au numéro de téléphone. Si 24 heures se sont écoulées et que le code ou le lien de votre utilisateur n'est plus valide, appelez l'opération [ResendConfirmationCode](#) API pour générer et envoyer un nouveau code ou un nouveau lien.

En fournissant le code de vérification, un utilisateur prouve qu'il a accès à la boîte aux lettres ou au téléphone qui a reçu le code. Une fois que l'utilisateur a fourni le code, Amazon Cognito met à jour les informations relatives à l'utilisateur dans votre groupe d'utilisateurs en :

- Définissant le statut de l'utilisateur sur CONFIRMED.
- Mettant à jour les attributs de l'utilisateur pour indiquer que l'adresse e-mail ou le numéro de téléphone est vérifié(e).

Pour afficher ces informations, vous pouvez utiliser la console Amazon Cognito. Vous pouvez également utiliser l'opération `AdminGetUser` API, la `admin-get-user` commande associée au AWS CLI ou une action correspondante dans l'un des AWS SDK.

Si un utilisateur dispose d'une méthode de contact vérifiée, Amazon Cognito envoie automatiquement un message à l'utilisateur lorsque l'utilisateur demande la réinitialisation du mot de passe.

Pour configurer votre groupe d'utilisateurs pour exiger une vérification par téléphone ou par e-mail

Pensez à vérifier les adresses e-mail et les numéros de téléphone de vos utilisateurs afin de pouvoir les contacter. Procédez comme suit AWS Management Console pour configurer votre groupe d'utilisateurs afin de demander à vos utilisateurs de confirmer leur adresse e-mail ou leur numéro de téléphone.

Note

Si vous n'avez pas encore de groupe d'utilisateurs dans votre compte, consultez [Démarrage avec les groupes d'utilisateurs](#).

Pour configurer votre groupe d'utilisateurs

1. Allez dans la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Dans le volet de navigation, choisissez Groupes d'utilisateurs. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
3. Choisissez l'onglet Sign-up experience (Expérience d'inscription) et localisez Attribute verification and user account confirmation (Vérification des attributs et confirmation du compte d'utilisateur). Choisissez Modifier.
4. Sous Vérification et confirmation assistées par Cognito, indiquez si vous allez Autoriser Cognito à envoyer automatiquement des messages pour vérifier et confirmer. Lorsque ce paramètre est activé, Amazon Cognito envoie des messages aux attributs de contact utilisateur que vous choisissez lorsqu'un utilisateur s'inscrit ou que vous créez un profil utilisateur. Pour vérifier les attributs et confirmer les profils utilisateur pour la connexion, Amazon Cognito envoie un code ou un lien dans des messages aux utilisateurs. Les utilisateurs doivent ensuite saisir le code dans votre interface utilisateur afin que votre application puisse le confirmer dans une demande d'API `ConfirmSignUp` ou `AdminConfirmSignUp`.

Note

Vous pouvez aussi désactiver Vérification et confirmation assistées par Cognito et utilisez des actions API authentifiées ou des déclencheurs Lambda pour vérifier les attributs et confirmer les utilisateurs.

Si vous choisissez cette option, Amazon Cognito n'envoie pas de code de vérification quand l'utilisateur s'inscrit. Choisissez cette option si vous utilisez un flux d'authentification personnalisé qui vérifie au moins une méthode de contact sans utiliser les codes de vérification en provenance d'Amazon Cognito. Par exemple, vous pouvez utiliser un déclencheur Lambda avant inscription qui vérifie automatiquement les adresses électroniques qui appartiennent à un domaine spécifique.

Si vous ne vérifiez pas les coordonnées de vos utilisateurs, ils risquent de ne pas pouvoir utiliser votre application. N'oubliez pas que les utilisateurs nécessitent des informations de contact vérifiées pour :

- Réinitialiser leurs mots de passe – Lorsqu'un utilisateur choisit une option dans votre application qui appelle l'action API `ForgotPassword`, Amazon Cognito envoie un mot de passe temporaire à l'adresse e-mail ou au numéro de téléphone de l'utilisateur.

Amazon Cognito envoie ce mot de passe uniquement si l'utilisateur a vérifié au moins une méthode de contact.

- Se connecter en utilisant une adresse e-mail ou un numéro de téléphone comme alias – Si vous configurez votre groupe d'utilisateurs pour autoriser ces alias, un utilisateur peut se connecter avec un alias uniquement si celui-ci est vérifié. Pour plus d'informations, consultez [Personnalisation des attributs de connexion](#).

5. Choisissez votre Attributs à vérifier :

Envoyer un message SMS, vérifier le numéro de téléphone

Amazon Cognito envoie par SMS un code de vérification lorsque l'utilisateur se connecte. Choisissez cette option si vous communiquez généralement avec vos utilisateurs par SMS. Par exemple, vous voudrez utiliser des numéros de téléphone vérifiés si vous envoyez des notifications de livraison, des confirmations de rendez-vous ou des alertes. Les numéros de téléphone des utilisateurs seront l'attribut vérifié lorsque les comptes sont confirmés. Vous devez prendre des mesures supplémentaires pour vérifier et communiquer avec les adresses e-mail des utilisateurs.

Envoyer un message électronique, vérifier l'adresse e-mail

Amazon Cognito envoie par e-mail un code de vérification lorsque l'utilisateur se connecte. Choisissez cette option si vous communiquez généralement avec vos utilisateurs par e-mail. Par exemple, vous souhaitez utiliser des adresses e-mail vérifiées si vous envoyez des relevés de facturation, des récapitulatifs de commande ou des offres spéciales. Les adresses e-mail des utilisateurs seront l'attribut vérifié lorsque les comptes sont confirmés. Vous devez prendre des mesures supplémentaires pour vérifier et communiquer avec les numéros de téléphone des utilisateurs.

Envoyer un message SMS si le numéro de téléphone est disponible, sinon envoyer un message électronique

Choisissez cette option si vous n'exigez pas que tous les utilisateurs aient la même méthode de contact vérifié. Dans ce cas, la page d'inscription de votre application peut demander aux utilisateurs de vérifier seulement leur mode de contact préféré. Quand Amazon Cognito envoie un code de vérification, il envoie le code à la méthode de contact fournie dans la demande SignUp de votre application. Si un utilisateur fournit une adresse e-mail et un numéro de téléphone, et que votre application fournit les deux méthodes de contact dans la

demande `SignUp`, Amazon Cognito envoie un code de vérification uniquement au numéro de téléphone.

Si vous exigez que les utilisateurs soient vérifiés tant à l'aide de leur adresse e-mail qu'à l'aide de leur numéro de téléphone, choisissez cette option. Amazon Cognito vérifie une méthode de contact lorsque l'utilisateur s'inscrit, et votre application doit vérifier l'autre méthode de contact une fois l'utilisateur connecté. Pour plus d'informations, consultez [Si vous exigez que les utilisateurs confirment leurs adresses e-mail et numéros de téléphone](#).

6. Choisissez Enregistrer les modifications.

Flux d'authentification avec vérification par e-mail ou par téléphone

Si votre groupe d'utilisateurs requiert que les utilisateurs vérifient leurs informations de contact, votre application doit faciliter le flux suivant quand un utilisateur se connecte :

1. Un utilisateur s'inscrit dans votre application en entrant un nom d'utilisateur, un numéro de téléphone et/ou une adresse e-mail, et éventuellement d'autres attributs.
2. Le service Amazon Cognito reçoit la demande d'inscription de l'application. Après avoir vérifié que la demande contient tous les attributs nécessaires pour l'inscription, le service termine le processus d'inscription et envoie un code de confirmation à l'utilisateur par téléphone (via SMS) ou par e-mail. Le code est valide pendant 24 heures.
3. Le service indique à l'application que l'inscription est terminée et que le compte d'utilisateur est en attente de confirmation. La réponse précise où le code de confirmation a été envoyé. A ce stade, le compte d'utilisateur a l'état Non confirmé, et l'adresse e-mail de l'utilisateur ainsi que son numéro de téléphone affichent l'état Non vérifié.
4. L'application peut maintenant inviter l'utilisateur à entrer le code de confirmation. Il n'est pas nécessaire que l'utilisateur entre ce code immédiatement. Toutefois, il ne pourra pas se connecter tant qu'il n'aura pas saisi le code de confirmation.
5. L'utilisateur entre le code de confirmation dans l'application.
6. L'application appelle l'API [ConfirmSignUp](#) pour envoyer le code au service Amazon Cognito qui le vérifie et, si le code est correct, affecte au compte d'utilisateur l'état Confirmed (Confirmé). Une fois le compte d'utilisateur effectivement confirmé, le service Amazon Cognito marque automatiquement l'attribut utilisé pour la confirmation (adresse e-mail ou numéro de téléphone) comme vérifié. Sauf modification de la valeur de cet attribut, l'utilisateur n'aura pas à le vérifier à nouveau.
7. A ce stade, le compte d'utilisateur a l'état Confirmé et l'utilisateur peut se connecter.

Si vous exigez que les utilisateurs confirment leurs adresses e-mail et numéros de téléphone

Amazon Cognito ne vérifie qu'une seule méthode de contact lorsqu'un utilisateur se connecte. Dans les cas où Amazon Cognito doit choisir entre une vérification par adresse e-mail et par numéro de téléphone, le service choisit de vérifier le numéro de téléphone en envoyant un code de vérification par SMS. Par exemple, si vous configurez votre groupe d'utilisateurs pour permettre aux utilisateurs de vérifier l'adresse e-mail ou le numéro de téléphone, et si votre application fournit ces deux attributs au moment de la connexion, Amazon Cognito vérifie uniquement le numéro de téléphone. Une fois qu'un utilisateur a vérifié son numéro de téléphone, Amazon Cognito définit le statut de l'utilisateur sur CONFIRMED et l'utilisateur est autorisé à se connecter à votre application.

Une fois que l'utilisateur s'est connecté, votre application peut fournir la possibilité de vérifier la méthode de contact qui n'a pas été vérifiée au cours de la connexion. Pour vérifier cette deuxième méthode, votre application appelle l'action d'API `VerifyUserAttribute`. Notez que cette action requiert un paramètre `AccessToken` et qu'Amazon Cognito fournit uniquement les jetons d'accès pour les utilisateurs authentifiés. Par conséquent, vous pouvez vérifier la deuxième méthode de contact uniquement une fois que l'utilisateur s'est connecté.

Si vous avez besoin que vos utilisateurs vérifient leur adresse électronique et leur numéro de téléphone, procédez comme suit :

1. Configurez votre groupe d'utilisateurs afin d'autoriser les utilisateurs à vérifier l'adresse e-mail ou le numéro de téléphone.
2. Dans le flux de connexion de l'application, exigez que les utilisateurs fournissent à la fois une adresse e-mail et un numéro de téléphone. Appelez l'action d'API [SignUp](#) et fournissez l'adresse e-mail et le numéro de téléphone pour le paramètre `UserAttributes`. À ce stade, Amazon Cognito envoie un code de vérification au téléphone de l'utilisateur.
3. Dans l'interface de votre application, présentez une page de confirmation sur laquelle l'utilisateur entre le code de vérification. Confirmez l'utilisateur en appelant l'action d'API [ConfirmSignUp](#). À ce stade, le statut de l'utilisateur est CONFIRMED et le numéro de téléphone de l'utilisateur est vérifié, mais l'adresse e-mail, elle, ne l'est pas.
4. Affichez la page de connexion et authentifiez l'utilisateur en appelant l'action d'API [InitiateAuth](#). Une fois l'utilisateur authentifié, Amazon Cognito renvoie un jeton d'accès à votre application.
5. Appelez l'action d'API [GetUserAttributeVerificationCode](#). Spécifiez les paramètres suivants dans la demande :

- `AccessToken` – Jeton d'accès renvoyé par Amazon Cognito lorsque l'utilisateur s'est connecté.
- `AttributeName` – Spécifiez "email" comme valeur d'attribut.

Amazon Cognito envoie un code de vérification à l'adresse e-mail de l'utilisateur.

6. Affichez une page de confirmation sur laquelle l'utilisateur entre le code de vérification. Lorsque l'utilisateur envoie le code, appelez l'action d'API [VerifyUserAttribute](#). Spécifiez les paramètres suivants dans la demande :

- `AccessToken` – Jeton d'accès renvoyé par Amazon Cognito lorsque l'utilisateur s'est connecté.
- `AttributeName` – Spécifiez "email" comme valeur d'attribut.
- `Code` – Code de vérification que l'utilisateur a fourni.

À ce stade, l'adresse e-mail est vérifiée.

Autorisation des utilisateurs à s'inscrire dans votre application, mais en les confirmant en tant qu'administrateur du groupe d'utilisateurs

Vous ne voulez peut-être pas que votre groupe d'utilisateurs envoie automatiquement des messages de vérification dans votre groupe d'utilisateurs, mais vous souhaitez tout de même autoriser tout le monde à s'inscrire à un compte. Ce modèle permet, par exemple, une vérification humaine des nouvelles demandes d'inscription, ainsi que la validation et le traitement par lots des inscriptions. Vous pouvez confirmer les nouveaux comptes utilisateurs dans la console Amazon Cognito ou à l'aide de l'API authentifiée par IAM. [AdminConfirmSignUp](#) Vous pouvez confirmer les comptes d'utilisateur en tant qu'administrateur, que votre groupe d'utilisateurs envoie ou non des messages de vérification.

Vous ne pouvez confirmer l'inscription en libre-service d'un utilisateur qu'à l'aide de cette technique. Pour confirmer un utilisateur que vous créez en tant qu'administrateur, créez une demande d'[AdminSetUserPassword](#)API avec `Permanent set to True`.

1. Un utilisateur s'inscrit dans votre application en entrant un nom d'utilisateur, un numéro de téléphone et/ou une adresse e-mail, et éventuellement d'autres attributs.

2. Le service Amazon Cognito reçoit la demande d'inscription de l'application. Après avoir vérifié que la demande contient tous les attributs nécessaires pour l'inscription, le service termine la procédure d'inscription et indique à l'application que l'inscription est faite, en attente de confirmation. A ce stade, le compte d'utilisateur affiche l'état Non confirmé. L'utilisateur ne peut pas se connecter tant que son compte n'a pas été confirmé.
3. Confirmez le compte de l'utilisateur. Vous devez vous connecter AWS Management Console ou signer votre demande d'API avec des AWS informations d'identification pour confirmer le compte.
 - a. Pour confirmer un utilisateur dans la console Amazon Cognito, accédez à l'onglet Utilisateurs, choisissez l'utilisateur que vous souhaitez confirmer, puis, dans le menu Actions, sélectionnez Confirmer.
 - b. Pour confirmer un utilisateur dans l' AWS API ou la CLI, créez une demande d'[AdminConfirmSignUp](#)API ou [admin-confirm-sign-up](#) dans le AWS CLI.
4. A ce stade, le compte d'utilisateur a l'état Confirmé et l'utilisateur peut se connecter.

Calcul des valeurs de hachage secret

Attribuez un secret client à votre client d'application confidentiel en guise de bonne pratique.

Lorsque vous attribuez un secret client à votre client d'application, vos demandes d'API de groupes d'utilisateurs Amazon Cognito doivent comporter un hachage qui inclut le secret client dans le corps de la demande. Pour valider votre connaissance du secret client pour les opérations d'API figurant dans les listes suivantes, concaténez le secret client avec l'ID de votre client d'application et le nom d'utilisateur de votre utilisateur, puis codez cette chaîne en base64.

Lorsque votre application connecte des utilisateurs à un client doté d'un hachage secret, vous pouvez utiliser la valeur de n'importe quel attribut de connexion au groupe d'utilisateurs comme élément de nom d'utilisateur du hachage secret. Lorsque votre application demande de nouveaux jetons lors d'une opération d'authentification avec `REFRESH_TOKEN_AUTH`, la valeur de l'élément `username` dépend de vos attributs de connexion. Lorsque votre groupe d'utilisateurs n'a pas d'attribut de connexion `username`, définissez la valeur du nom d'utilisateur de hachage secret de la demande sub de l'utilisateur à partir de son jeton d'accès ou d'identification. Lorsque `username` est un attribut de connexion, définissez la valeur du nom d'utilisateur de hachage secret indiquée dans la demande `username`.

Les API de groupes d'utilisateurs Amazon Cognito suivantes acceptent une valeur de hachage de secret client dans un paramètre `SecretHash`.

- [ConfirmForgotPassword](#)
- [ConfirmSignUp](#)
- [ForgotPassword](#)
- [ResendConfirmationCode](#)
- [SignUp](#)

De plus, les API suivantes acceptent une valeur de hachage de secret client dans un paramètre SECRET_HASH, que ce soit dans les paramètres d'authentification ou dans la réponse à une question de sécurité.

Opération API	Paramètre parent pour SECRET_HASH
InitiateAuth	AuthParameters
AdminInitiateAuth	AuthParameters
RespondToAuthChallenge	ChallengeResponses
AdminRespondToAuthChallenge	ChallengeResponses

La valeur de hachage secret est un code d'authentification de message de hachage (HMAC) à clé codé en Base 64 qui est calculé à partir de la clé secrète du client et du nom d'utilisateur d'un groupe d'utilisateurs et de l'ID client contenu dans le message. L'exemple de pseudo-code suivant montre comment cette valeur est calculée. Dans ce pseudo-code, + indique une concaténation, HMAC_SHA256 représente une fonction qui génère une valeur HMAC avec HmacSHA256, et Base64 représente une fonction qui génère une version codée en Base 64 de la sortie de hachage.

```
Base64 ( HMAC_SHA256 ( "Client Secret Key", "Username" + "Client Id" ) )
```

Pour une présentation détaillée du calcul et de l'utilisation du SecretHash paramètre, consultez [Comment résoudre les erreurs « Impossible de vérifier le hachage secret pour le client » dans l'API de mon groupe d'utilisateurs Amazon Cognito <client-id>?](#) dans le AWS Knowledge Center.

Vous pouvez utiliser les exemples de code suivants dans votre code d'application côté serveur.

Shell

```
echo -n "[username][app client ID]" | openssl dgst -sha256 -hmac [app client secret]
-binary | openssl enc -base64
```

Java

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

public static String calculateSecretHash(String userPoolClientId, String
userPoolClientSecret, String userName) {
    final String HMAC_SHA256_ALGORITHM = "HmacSHA256";

    SecretKeySpec signingKey = new SecretKeySpec(
        userPoolClientSecret.getBytes(StandardCharsets.UTF_8),
        HMAC_SHA256_ALGORITHM);

    try {
        Mac mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
        mac.init(signingKey);
        mac.update(userName.getBytes(StandardCharsets.UTF_8));
        byte[] rawHmac =
mac.doFinal(userPoolClientId.getBytes(StandardCharsets.UTF_8));
        return Base64.getEncoder().encodeToString(rawHmac);
    } catch (Exception e) {
        throw new RuntimeException("Error while calculating ");
    }
}
```

Python

```
import sys
import hmac, hashlib, base64
username = sys.argv[1]
app_client_id = sys.argv[2]
key = sys.argv[3]
message = bytes(sys.argv[1]+sys.argv[2], 'utf-8')
key = bytes(sys.argv[3], 'utf-8')
secret_hash = base64.b64encode(hmac.new(key, message,
digestmod=hashlib.sha256).digest()).decode()
print("SECRET HASH:", secret_hash)
```

Confirmation des comptes d'utilisateur sans vérification de l'e-mail ou du numéro de téléphone

Le déclencheur Lambda avant l'inscription peut servir à valider automatiquement les comptes d'utilisateur au moment de l'inscription, sans exiger de code de confirmation ni vérifier l'adresse e-mail ou le numéro de téléphone. Les utilisateurs qui sont confirmés de cette façon peuvent immédiatement se connecter, sans avoir besoin d'un code.

Vous pouvez également marquer un e-mail ou le numéro de téléphone de l'utilisateur comme étant vérifié grâce à ce déclencheur.

Note

Bien que cette méthode soit pratique pour les utilisateurs qui débutent, nous recommandons de vérifier au moins l'e-mail ou le numéro de téléphone. Dans le cas contraire, l'utilisateur peut se trouver dans l'incapacité de récupérer son mot de passe en cas d'oubli.

Si vous n'exigez pas que l'utilisateur reçoive et saisisse un code de confirmation au moment de l'inscription et que vous ne vérifiez pas automatiquement l'adresse e-mail ni le numéro de téléphone dans le déclencheur Lambda avant l'inscription, vous risquez de ne pas avoir de numéro de téléphone ni d'adresse e-mail vérifiés pour ce compte d'utilisateur. L'utilisateur peut vérifier son numéro de téléphone ou son adresse e-mail ultérieurement. Toutefois, si l'utilisateur oublie son mot de passe et n'a pas de numéro de téléphone ni d'adresse e-mail vérifiés, il n'a plus aucun moyen d'accéder à son compte, puisque le flux d'oubli du mot de passe exige une adresse e-mail ou un numéro de téléphone vérifié pour envoyer un code de vérification à l'utilisateur.

Vérification en cas de modification de l'adresse e-mail ou du numéro de téléphone par l'utilisateur

Lorsqu'un utilisateur met à jour son adresse e-mail ou son numéro de téléphone dans votre application, Amazon Cognito envoie immédiatement un message contenant un code de vérification à un utilisateur si vous avez configuré votre groupe d'utilisateurs pour vérifier automatiquement cet attribut. L'utilisateur doit ensuite fournir le code du message de vérification à votre application. Votre application envoie ensuite le code dans une demande d'[VerifyUserAttribute](#) API pour terminer la vérification de la nouvelle valeur d'attribut.

Si votre groupe d'utilisateurs n'exige pas que les utilisateurs vérifient une adresse e-mail ou un numéro de téléphone mis à jour, Amazon Cognito modifie immédiatement la valeur d'un attribut

`email` ou `phone_number` mis à jour, et marque l'attribut comme non vérifié. Votre utilisateur ne peut pas se connecter avec une adresse e-mail ou un numéro de téléphone non vérifiés. Il doit terminer la vérification de la valeur mise à jour avant de pouvoir utiliser cet attribut comme alias de connexion.

Si votre groupe d'utilisateurs exige que les utilisateurs vérifient une adresse e-mail ou un numéro de téléphone mis à jour, Amazon Cognito laisse l'attribut vérifié et défini sur sa valeur d'origine jusqu'à ce que votre utilisateur vérifie la nouvelle valeur d'attribut. Si l'attribut est un alias de connexion, votre utilisateur peut se connecter avec la valeur d'attribut d'origine jusqu'à ce que la vérification modifie l'attribut avec la nouvelle valeur. Pour plus d'informations sur la façon de configurer votre groupe d'utilisateurs pour demander aux utilisateurs de vérifier les attributs mis à jour, consultez [Configuration de la vérification par e-mail ou par téléphone](#).

Pour personnaliser ce message, vous pouvez utiliser un déclencheur Lambda de message personnalisé. Pour plus d'informations, consultez [Déclencheur Lambda message personnalisé](#). Quand l'adresse e-mail et le numéro de téléphone d'un utilisateur ne sont pas vérifiés, votre application doit informer l'utilisateur qu'il doit vérifier l'attribut et elle doit fournir un bouton ou un lien permettant à l'utilisateur de vérifier sa nouvelle adresse e-mail ou son nouveau numéro de téléphone.

Procédures de confirmation et de vérification pour les comptes d'utilisateur créés par des administrateurs ou des développeurs

Les comptes d'utilisateur créés par un administrateur ou un développeur affichent déjà l'état Confirmé, et les utilisateurs n'ont pas à saisir de code de confirmation. Le message d'invitation que le service Amazon Cognito envoie à ces utilisateurs inclut leur nom d'utilisateur et un mot de passe temporaire. L'utilisateur doit modifier le mot de passe avant de se connecter. Pour plus d'informations, consultez la section [Personnalisation des e-mails et SMS](#) dans [Création de comptes d'utilisateur en tant qu'administrateur](#) et celle relative au déclencheur de message personnalisé dans [Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda](#).

Procédures de vérification et de confirmation pour les comptes d'utilisateur importés

Les comptes utilisateur créés à l'aide de la fonctionnalité d'importation d'utilisateurs de la AWS Management Console CLI ou de l'API (voir [Importation d'utilisateurs dans des groupes d'utilisateurs depuis un fichier CSV](#)) sont déjà confirmés. Les utilisateurs ne sont donc pas tenus de saisir un code de confirmation. Aucun message d'invitation n'est envoyé. Cependant, les comptes d'utilisateur importés nécessitent que les utilisateurs demandent d'abord un code en appelant

l'API `ForgotPassword`, puis en créant un mot de passe avec le code reçu après l'appel de l'API `ConfirmForgotPassword` avant de se connecter. Pour plus d'informations, consultez [Obligation pour les utilisateurs importés de réinitialiser leur mot de passe](#).

L'e-mail de l'utilisateur ou son numéro de téléphone est marqué comme vérifié lorsque le compte d'utilisateur est importé ; aucune vérification n'est donc requise lorsque l'utilisateur se connecte.

Envoi d'e-mails pendant le test de votre application

Amazon Cognito envoie un e-mail à vos utilisateurs lorsqu'ils créent et gèrent leurs comptes dans l'application cliente de votre groupe d'utilisateurs. Si vous configurez votre groupe d'utilisateurs afin d'exiger la vérification des e-mails, Amazon Cognito envoie un e-mail dans les cas suivants :

- Un utilisateur se connecte.
- Un utilisateur met à jour son adresse e-mail.
- Un utilisateur exécute une action qui appelle l'action d'API `ForgotPassword`.
- Vous créez un compte d'utilisateur en tant qu'administrateur.

En fonction de l'action qui déclenche l'e-mail, celui-ci contient un code de vérification ou un mot de passe temporaire. Vos utilisateurs doivent recevoir ces e-mails et comprendre le message. Sinon, ils peuvent ne pas être en mesure de se connecter et d'utiliser votre application.

Pour garantir que les e-mails sont envoyés avec succès et que le message paraît correct, testez les actions de votre application qui initient les remises d'e-mail à partir d'Amazon Cognito. Par exemple, en utilisant la page de connexion de votre application, ou en utilisant l'action d'API `SignUp`, vous pouvez lancer un e-mail en vous connectant avec une adresse e-mail de test. Lorsque vous testez ainsi, n'oubliez pas les points suivants :

Important

Lorsque vous utilisez une adresse e-mail pour tester les actions qui initient des e-mails à partir d'Amazon Cognito, n'utilisez pas d'adresse e-mail fictive (une adresse sans boîte aux lettres). Utilisez une adresse e-mail réelle qui recevra l'e-mail à partir d'Amazon Cognito sans déclencher de retour à l'expéditeur.

Un message d'erreur définitif se produit quand Amazon Cognito ne parvient pas à remettre l'e-mail à la boîte aux lettres du destinataire, ce qui se produit toujours si la boîte aux lettres n'existe pas.

Amazon Cognito limite le nombre d'e-mails pouvant être envoyés par des AWS comptes régulièrement soumis à des hard bounces.

Lorsque vous testez des actions qui initient des e-mails, utilisez l'une des adresses suivantes pour empêcher les retours à l'expéditeur définitifs :

- Une adresse pour un compte de messagerie que vous possédez et utilisez à des fins de test. Lorsque vous utilisez votre propre adresse e-mail, vous recevez l'e-mail envoyé par Amazon Cognito. Avec cet e-mail, vous pouvez utiliser le code de vérification pour tester la connexion de votre application. Si vous avez personnalisé le message électronique pour votre groupe d'utilisateurs, vous pouvez vérifier que vos personnalisations est correcte.
- Adresse du simulateur de boîte aux lettres, `success@simulator.amazonses.com`. Si vous utilisez l'adresse du simulateur, Amazon Cognito envoie l'e-mail avec succès, mais vous n'êtes pas en mesure de l'afficher. Cette option est utile lorsque vous n'avez pas besoin d'utiliser le code de vérification ni de vérifier le message électronique.
- Adresse du simulateur de boîte aux lettres avec ajout d'une étiquette arbitraire, comme `success+user1@simulator.amazonses.com` ou `success+user2@simulator.amazonses.com`. Amazon Cognito envoie des e-mails à ces adresses avec succès, mais vous n'êtes pas en mesure de les afficher. Cette option est utile lorsque vous souhaitez tester la procédure de connexion en ajoutant plusieurs utilisateurs de test à votre groupe d'utilisateurs et que chaque utilisateur de test possède une adresse e-mail unique.

Configuration de la vérification par e-mail ou par téléphone

Vous pouvez choisir des paramètres pour la vérification par e-mail ou par téléphone dans l'onglet Messagerie. Pour plus d'informations sur l'authentification multifacteur (MFA), consultez [MFA par SMS](#).

Amazon Cognito utilise Amazon SNS pour envoyer des SMS. Si vous n'avez jamais envoyé de SMS depuis Amazon Cognito ou un autre Service AWS service, Amazon SNS peut placer votre compte dans le sandbox SMS. Nous vous recommandons d'envoyer un message de test à un numéro de téléphone vérifié avant de migrer votre compte de l'environnement de test (sandbox) en production. En outre, si vous prévoyez d'envoyer des SMS à des numéros de téléphone de destination aux États-Unis, vous devez obtenir un identifiant d'origine ou d'expéditeur à partir d'Amazon Pinpoint. Pour configurer votre groupe d'utilisateurs Amazon Cognito pour les SMS, consultez [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#).

Amazon Cognito peut vérifier automatiquement les adresses e-mail et les numéros de téléphone. Pour effectuer cette vérification, Amazon Cognito envoie un code de vérification ou un lien de vérification. Pour les adresses e-mail, Amazon Cognito peut envoyer un code ou un lien dans un e-mail. Vous pouvez choisir un Type de vérification par Code ou Lien lorsque vous modifiez votre modèle Message de vérification dans l'onglet Messagerie de la console Amazon Cognito. Pour plus d'informations, consultez [Personnalisation des messages de vérification d'adresse e-mail](#).

Pour les numéros de téléphone, Amazon Cognito envoie un code par SMS.

Amazon Cognito doit vérifier un numéro de téléphone ou une adresse e-mail afin de confirmer les utilisateurs et les aider à récupérer les mots de passe oubliés. Vous pouvez également confirmer automatiquement les utilisateurs à l'aide du déclencheur Lambda préalable à l'inscription ou utiliser [AdminConfirmSignUp](#) l'opération API. Pour plus d'informations, consultez [Inscription et confirmation des comptes d'utilisateur](#).

Le code ou lien de vérification est valide pendant 24 heures.

Si vous décidez d'exiger une vérification pour une adresse e-mail ou un numéro de téléphone, Amazon Cognito envoie automatiquement le code ou le lien de vérification quand un utilisateur s'inscrit. Si le groupe d'utilisateurs dispose d'un [Déclencheur Lambda de l'expéditeur de SMS personnalisé](#) ou [Déclencheur Lambda expéditeur d'e-mail personnalisé](#) configuré, cette fonction est appelée à la place.

Remarques

- Amazon SNS facture séparément les SMS utilisés pour la vérification des numéros de téléphone. Aucun frais ne s'applique à l'envoi d'e-mails. Pour plus d'informations sur la tarification Amazon SNS, consultez [Tarification SMS internationaux](#). Pour obtenir la liste des pays où la messagerie SMS est disponible, consultez la page [Régions et pays pris en charge](#).
- Lorsque vous testez des actions dans votre application qui génèrent des e-mails à partir d'Amazon Cognito, utilisez une adresse e-mail réelle joignable par Amazon Cognito sans messages d'erreur définitifs. Pour plus d'informations, consultez [the section called “Envoi d'e-mails pendant le test de votre application”](#).
- Le flux de mot de passe oublié requiert l'adresse e-mail ou le numéro de téléphone de l'utilisateur pour vérifier l'utilisateur.

⚠ Important

Si un utilisateur s'inscrit en indiquant à la fois un numéro de téléphone et une adresse e-mail, et que les paramètres du groupe d'utilisateurs nécessitent la vérification des deux attributs, Amazon Cognito envoie un code de vérification par SMS au numéro de téléphone. Amazon Cognito n'a pas encore vérifié l'adresse e-mail. Votre application doit donc appeler [GetUser](#) pour savoir si une adresse e-mail attend d'être vérifiée. Si une vérification est nécessaire, l'application doit appeler [GetUserAttributeVerificationCode](#) pour lancer le flux de vérification par e-mail. Il doit ensuite soumettre le code de vérification en appelant [VerifyUserAttribute](#).

Vous pouvez ajuster votre quota de dépenses par SMS pour un Compte AWS et pour des messages individuels. Les limites s'appliquent uniquement au coût d'envoi de messages SMS. Pour plus d'informations, consultez [Que sont les quotas de dépenses par compte et par messages, et comment fonctionnent-ils ?](#) dans les [Questions fréquentes sur Amazon SNS](#).

Amazon Cognito envoie des SMS à l'aide des ressources Amazon SNS dans la région où vous avez créé Région AWS le groupe d'utilisateurs ou dans une ancienne région Amazon SNS, comme indiqué dans le tableau suivant. L'exception se rapporte aux groupes d'utilisateurs Amazon Cognito dans la région Asie-Pacifique (Séoul). Ces groupes d'utilisateurs utilisent votre configuration Amazon SNS dans la région Asie-Pacifique (Tokyo). Pour plus d'informations, consultez [Choisissez le Région AWS pour les messages SMS Amazon SNS](#).

Région Amazon Cognito	Autre région Amazon SNS héritée
USA Est (Ohio)	USA Est (Virginie du Nord)
Asie-Pacifique (Mumbai)	Asie-Pacifique (Singapour)
Asie-Pacifique (Séoul)	Asie-Pacifique (Tokyo)
Canada (Centre)	USA Est (Virginie du Nord)
Europe (Francfort)	Europe (Irlande)
Europe (Londres)	Europe (Irlande)

Exemple : Si votre groupe d'utilisateurs Amazon Cognito se trouve en Asie-Pacifique (Mumbai) et que vous avez augmenté votre limite de dépenses dans ap-southeast-1, vous ne voudrez peut-être pas demander une augmentation distincte dans ap-south-1. Au lieu de cela, vous pouvez utiliser vos ressources Amazon SNS en Asie-Pacifique (Singapour).

Vérification des mises à jour des adresses e-mail et des numéros de téléphone

Un attribut d'adresse e-mail ou de numéro de téléphone peut devenir actif et non vérifié immédiatement après modification de sa valeur par l'utilisateur. Amazon Cognito peut également exiger que votre utilisateur vérifie la nouvelle valeur avant qu'Amazon Cognito mette à jour l'attribut. Quand vous exigez que vos utilisateurs vérifient d'abord la nouvelle valeur, ils peuvent utiliser la valeur d'origine pour se connecter et recevoir des messages jusqu'à ce qu'ils vérifient la nouvelle valeur.

Quand vos utilisateurs peuvent utiliser leur adresse e-mail ou leur numéro de téléphone comme alias de connexion dans votre groupe d'utilisateurs, leur nom de connexion pour un attribut mis à jour dépend des exigences de vérification éventuellement définies pour les attributs mis à jour. Quand vous exigez que les utilisateurs vérifient un attribut mis à jour, un utilisateur peut se connecter avec la valeur d'attribut d'origine jusqu'à ce qu'il vérifie la nouvelle valeur. Quand vous n'exigez pas que les utilisateurs vérifient un attribut mis à jour, un utilisateur ne peut pas se connecter ni recevoir de messages à la nouvelle valeur d'attribut ou à la valeur d'origine tant qu'il n'a pas vérifié la nouvelle valeur.

Par exemple, votre groupe d'utilisateurs autorise la connexion avec un alias d'adresse e-mail et exige que les utilisateurs vérifient leur adresse e-mail lors de la mise à jour. Sue, qui se connecte avec `sue@example.com`, souhaite remplacer son adresse e-mail par `sue2@example.com`, mais saisit accidentellement `ssue2@example.com`. Sue ne reçoit pas l'e-mail de vérification et ne peut donc pas vérifier `ssue2@example.com`. Sue se connecte avec `sue@example.com` et soumet à nouveau le formulaire dans votre application pour mettre à jour son adresse e-mail en spécifiant `sue2@example.com`. Elle reçoit cet e-mail, fournit le code de vérification à votre application et commence à se connecter avec `sue2@example.com`.

Quand un utilisateur met à jour un attribut et que votre groupe d'utilisateurs vérifie les nouvelles valeurs d'attribut

- Ils peuvent se connecter avec la valeur d'attribut d'origine avant d'avoir confirmé le code pour vérifier la nouvelle valeur.
- Ils peuvent se connecter uniquement avec la nouvelle valeur d'attribut après avoir confirmé le code pour vérifier cette nouvelle valeur.

- Si vous avez défini `email_verified` ou `phone_number_verified` envoyé `true` une demande d'[AdminUpdateUserAttributes](#) API, ils peuvent se connecter avant d'avoir confirmé le code qu'Amazon Cognito leur a envoyé.

Quand un utilisateur met à jour un attribut et que votre groupe d'utilisateurs ne vérifie pas les nouvelles valeurs d'attribut

- Ils ne peuvent pas se connecter, ni recevoir de messages, avec la valeur d'attribut d'origine.
- Ils ne peuvent pas se connecter, ni recevoir de messages autres qu'un code de confirmation, avec la nouvelle valeur d'attribut avant d'avoir confirmé le code pour vérifier cette nouvelle valeur.
- Si vous avez défini `email_verified` ou `phone_number_verified` envoyé `true` une demande d'[AdminUpdateUserAttributes](#) API, ils peuvent se connecter avant d'avoir confirmé le code qu'Amazon Cognito leur a envoyé.

Pour exiger une vérification des attributs lorsque les utilisateurs mettent à jour leur adresse e-mail ou leur numéro de téléphone

1. Connectez-vous à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Dans le volet de navigation, choisissez Groupes d'utilisateurs, puis choisissez le groupe d'utilisateurs que vous souhaitez modifier.
3. Dans l'onglet Sign-up experience (Expérience d'inscription), choisissez Edit (Modifier) sous Attribute verification and user account confirmation (Vérification des attributs et confirmation du compte d'utilisateur).
4. Choisissez Keep original attribute value active when an update is pending (Conserver la valeur d'attribut d'origine active lorsqu'une mise à jour est en attente).
5. Sous Active attribute values when an update is pending (Valeurs d'attribut actives lorsqu'une mise à jour est en attente), choisissez les attributs que vos utilisateurs devront vérifier avant qu'Amazon Cognito mette à jour la valeur.
6. Sélectionnez Enregistrer les modifications.

Pour demander la vérification de la mise à jour des attributs avec l'API Amazon Cognito, vous pouvez définir le `AttributesRequireVerificationBeforeUpdate` paramètre dans une [UpdateUserPool](#) demande.

Autoriser Amazon Cognito à envoyer des SMS en votre nom.

Pour envoyer des SMS à vos utilisateurs en votre nom, Amazon Cognito a besoin de votre autorisation. Pour accorder cette autorisation, vous pouvez créer un rôle AWS Identity and Access Management (IAM). Dans l'onglet Messagerie de la console Amazon Cognito, sous SMS, choisissez Modifier pour définir un rôle.

Configuration des messages SMS, e-mails de vérification et messages d'invitation des utilisateurs

Amazon Cognito vous permet de personnaliser les messages de vérification par SMS et e-mail, ainsi que les messages d'invitation des utilisateurs, afin d'améliorer la sécurité et l'expérience utilisateur de votre application. Avec Amazon Cognito, vous pouvez choisir entre des vérifications basées sur le code ou des vérifications de liens en un clic pour répondre aux besoins de votre application. Cette rubrique explique comment personnaliser l'authentification multifactorielle (MFA) et les communications de vérification dans la console Amazon Cognito.

Dans l'onglet Messagerie, sous Modèles de messages, vous pouvez personnaliser :

- Votre message d'authentification multifacteur (MFA) par SMS
- Vos messages de vérification par SMS et e-mail
- Le type de vérification pour l'e-mail : code ou lien
- Vos messages d'invitation d'utilisateur
- Adresses DE et de RÉPONDRE À pour les e-mails transitant par votre groupe d'utilisateurs

Note

Les modèles de messages de vérification par SMS et par e-mail s'affichent uniquement si vous avez indiqué vouloir vérifier le numéro de téléphone et l'e-mail dans l'onglet Vérifications. De la même façon, le modèle de message MFA SMS s'affiche uniquement si le paramètre MFA est obligatoire ou facultatif.

Rubriques

- [Modèles de messages](#)
- [Personnalisation du message SMS](#)

- [Personnalisation des messages de vérification d'adresse e-mail](#)
- [Personnalisation des messages d'invitation des utilisateurs](#)
- [Personnalisation de votre adresse e-mail](#)
- [Autoriser Amazon Cognito à envoyer des e-mails Amazon SES en votre nom \(à partir d'une adresse DE personnalisée\)](#)

Modèles de messages

Vous pouvez utiliser des modèles de messages pour insérer des champs dans vos messages à l'aide d'espaces réservés qui seront remplacés par les valeurs correspondantes.

Espaces réservés de modèle

Description	Jeton
Code de vérification	{#####}
Mot de passe temporaire	{#####}
Nom utilisateur	{username}

Note

Vous ne pouvez pas utiliser l'espace réservé {username} dans les messages électroniques de vérification. Vous pouvez utiliser l' {username} espace réservé dans les e-mails d'invitation que vous générez avec l'[AdminCreateUser](#) opération. Ces messages électroniques d'invitation utilisent deux espaces réservés : le nom d'utilisateur, sous la forme {username}, et le mot de passe temporaire, sous la forme {#####}.

Vous pouvez utiliser des espaces réservés de modèle avec sécurité avancée pour effectuer les opérations suivantes :

- Inclure des détails spécifiques sur un événement, comme l'adresse IP, la ville, le pays, l'heure de connexion et le nom de l'appareil. Les fonctions de sécurité avancées d'Amazon Cognito peuvent analyser ces détails.
- Vérifier si un lien en 1 clic est valide.

- Utiliser un ID d'événement, un jeton de commentaire et un nom d'utilisateur pour créer votre propre lien en 1 clic.

Note

Pour générer des liens en un clic et utiliser les espaces réservés `{one-click-link-valid}` et `{one-click-link-invalid}` dans les modèles d'e-mails avec sécurité avancée, vous devez déjà disposer d'un domaine configuré pour votre groupe d'utilisateurs.

Espaces réservés de modèle avec sécurité avancée

Description	Jeton
Adresse IP	<code>{ip-address}</code>
Ville	<code>{city}</code>
Pays	<code>{country}</code>
Heure de connexion	<code>{login-time}</code>
Nom d'appareil	<code>{device-name}</code>
Le lien en un clic est valide	<code>{one-click-link-valid}</code>
Le lien en un clic est non valide	<code>{one-click-link-invalid}</code>
ID de l'événement	<code>{event-id}</code>
Jeton de commentaire	<code>{feedback-token}</code>

Personnalisation du message SMS

Note

Dans la nouvelle expérience de console Amazon Cognito, vous pouvez personnaliser les messages SMS

Vous pouvez personnaliser le message SMS pour l'authentification multifactorielle (MFA) dans l'onglet Messagerie situé sous le titre Modèles de messages.

⚠ Important

Votre message personnalisé doit contenir l'espace réservé {####}. Cet espace réservé est remplacé par le code d'authentification avant l'envoi.

Amazon Cognito impose une longueur maximale pour les SMS, code d'authentification compris, avec 140 caractères UTF-8.

Personnalisation des messages de vérification par SMS

Vous pouvez personnaliser le message SMS envoyé pour vérifier le numéro de téléphone. Pour ce faire, modifiez le modèle sous la question Souhaitez-vous personnaliser vos messages de vérification par SMS ?.

⚠ Important

Votre message personnalisé doit contenir l'espace réservé {####}. Cet espace réservé est remplacé par le code de vérification avant l'envoi.

La longueur maximale du message est de 140 caractères UTF-8, en comptant le code de vérification.

Personnalisation des messages de vérification d'adresse e-mail

Pour vérifier l'adresse e-mail d'un utilisateur de votre groupe d'utilisateurs avec Amazon Cognito, vous pouvez envoyer à l'utilisateur un message électronique contenant un lien qu'il peut sélectionner ou lui envoyer un code à saisir.

Pour personnaliser l'objet et le contenu du message pour les messages de vérification d'adresse e-mail, modifiez le modèle Message de vérification dans l'onglet Messagerie de votre groupe d'utilisateurs. Vous pouvez choisir un Type de vérification par Code ou Lien lorsque vous modifiez votre modèle Message de vérification.

Lorsque vous choisissez Code comme type de vérification, votre message personnalisé doit contenir l'espace réservé {####}. Quand vous envoyez ce message, le code de vérification remplace cet espace réservé.

Si vous choisissez Lien comme type de vérification, votre message personnalisé doit inclure un espace réservé au format {##Verify Your Email##}. Vous pouvez modifier la chaîne de texte entre les caractères d'espace réservé, par exemple {##Click here##}. Un lien de vérification intitulé Verify Your Email (Vérifiez votre adresse e-mail) remplace cet espace réservé.

Le lien vers un message de vérification par e-mail redirige votre utilisateur vers une URL comme dans l'exemple suivant.

```
https://<your user pool domain>/confirmUser/?  
client_id=abcdefg12345678&user_name=emailtest&confirmation_code=123456
```

La longueur maximale du message est de 20 000 caractères UTF-8, en comptant le code de vérification (le cas échéant). Vous pouvez utiliser des balises HTML dans ce message pour formater le contenu.

Personnalisation des messages d'invitation des utilisateurs

Vous pouvez personnaliser le message d'invitation des utilisateurs qu'Amazon Cognito envoie par SMS ou e-mail aux nouveaux utilisateurs en modifiant le modèle Messages d'invitation sous l'onglet Messagerie.

Important

Votre message personnalisé doit contenir les espaces réservés {username} et {####}. Lors de l'envoi du message d'invitation, Amazon Cognito remplace ces espaces réservés par le nom d'utilisateur et le mot de passe de votre utilisateur.

La longueur maximale d'un message SMS est de 140 caractères UTF-8, code de vérification inclus. La longueur maximale d'un message e-mail est de 20 000 caractères UTF-8, code de vérification inclus. Vous pouvez utiliser des balises HTML dans vos e-mails pour formater le contenu.

Personnalisation de votre adresse e-mail

Par défaut, les messages e-mail qu'Amazon Cognito envoie aux utilisateurs dans vos groupes d'utilisateurs arrivent par l'adresse no-reply@verificationemail.com. Vous pouvez choisir de spécifier des adresses e-mail DE et RÉPONDRE À personnalisées qui remplaceront no-reply@verificationemail.com.

Pour personnaliser les adresses e-mail DE et de RÉPONDRE À

1. Accédez à la [console Amazon Cognito](#), puis choisissez Groupes d'utilisateurs.
2. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
3. Choisissez l'onglet Messaging (Messagerie). Sous E-mail, choisissez Modifier.
4. Choisissez une région SES.
5. Choisissez une adresse e-mail d'expéditeur dans la liste des adresses e-mail que vous avez vérifiées avec Amazon SES dans la région SES que vous avez sélectionnée. Pour utiliser une adresse e-mail provenant d'un domaine vérifié, configurez les paramètres de messagerie dans l'AWS Command Line Interface ou l'API AWS. Pour plus d'informations, consultez la section [Vérification des adresses e-mail et des domaines dans Amazon SES](#) dans le Guide du développeur Amazon Simple Email Service.
6. Choisissez un jeu de configurations dans la liste des jeux de configurations de la région SES que vous avez choisie.
7. Saisissez un nom d'expéditeur DE convivial pour vos messages électroniques, au format `John Stiles <johnstiles@example.com>`.
8. Pour personnaliser l'adresse e-mail pour RÉPONDRE À, saisissez une adresse e-mail valide dans le champ Adresse e-mail pour RÉPONDRE À.

Autoriser Amazon Cognito à envoyer des e-mails Amazon SES en votre nom (à partir d'une adresse DE personnalisée)

Vous pouvez configurer Amazon Cognito pour qu'il envoie des e-mails à partir d'une adresse e-mail DE personnalisée au lieu de son adresse par défaut. Pour utiliser une adresse personnalisée, vous devez autoriser Amazon Cognito à envoyer un e-mail à partir d'une identité vérifiée Amazon SES. Dans la plupart des cas, vous pouvez accorder une autorisation en créant une politique d'autorisation d'envoi. Pour plus d'informations, consultez la section [Utilisation de l'autorisation d'envoi avec Amazon SES](#) dans le Guide du développeur Amazon Simple Email Service.

Lorsque vous configurez un groupe d'utilisateurs pour qu'il utilise Amazon SES pour les e-mails, Amazon Cognito crée le rôle `AWSServiceRoleForAmazonCognitoIdpEmailService` dans votre compte pour accorder l'accès à Amazon SES. Aucune politique d'autorisation d'envoi n'est nécessaire lorsque le rôle lié à un service `AWSServiceRoleForAmazonCognitoIdpEmailService` est utilisé. Vous n'avez besoin d'ajouter une politique d'autorisation d'envoi que lorsque vous utilisez la fonctionnalité de messagerie par défaut de votre groupe d'utilisateurs et une identité Amazon SES vérifiée en tant qu'adresse DE.

Pour plus d'informations sur le rôle lié à un service créé par Amazon Cognito, consultez [Utilisation de rôles liés à un service pour Amazon Cognito](#).

L'exemple suivant de politique d'autorisation d'envoi accorde à Amazon Cognito la possibilité limitée d'utiliser une identité vérifiée Amazon SES. Amazon Cognito ne peut envoyer des e-mails que lorsqu'il le fait pour le compte du groupe d'utilisateurs dans la condition `aws:SourceArn` et le compte dans la condition `aws:SourceAccount`. Pour d'autres exemples, consultez [Exemples de politiques d'autorisation d'envoi Amazon SES](#) dans le Guide du développeur Amazon Simple Email Service.

Note

Dans cet exemple, la valeur « Sid » est une chaîne arbitraire qui identifie l'instruction de manière unique. Pour plus d'informations sur la syntaxe de la politique, consultez [Politiques d'autorisation d'envoi Amazon SES](#) dans le Guide du développeur Amazon Simple Email Service.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "stmt1234567891234",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "email.cognito-idp.amazonaws.com"
        ]
      },
      "Action": [
        "SES:SendEmail",
        "SES:SendRawEmail"
      ],
      "Resource": "<your SES identity ARN>",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<your account number>"
        },
        "ArnLike": {
          "aws:SourceArn": "<your user pool ARN>"
        }
      }
    }
  ]
}
```

```
}  
  }  
] }  
}
```

La console Amazon Cognito vous ajoute une politique semblable lorsque vous sélectionnez une identité Amazon SES dans le menu déroulant. Si vous utilisez la CLI ou l'API pour configurer le groupe d'utilisateurs, vous devez attacher une politique structurée comme celle de l'exemple précédent à votre identité Amazon SES.

Création de comptes d'utilisateur en tant qu'administrateur

Une fois que vous avez défini un groupe d'utilisateurs, vous pouvez créer des utilisateurs via la AWS Management Console, l'AWS Command Line Interface ou l'API Amazon Cognito. Vous pouvez créer un profil pour un nouvel utilisateur dans un groupe d'utilisateurs et envoyer un message de bienvenue avec des instructions d'inscription à l'utilisateur par SMS ou e-mail.

Les développeurs et les administrateurs peuvent effectuer les tâches suivantes :

- Créez un nouveau profil utilisateur en utilisant AWS Management Console ou en appelant l'API `AdminCreateUser`.
- Définissez les valeurs des attributs utilisateur.
- Créez des attributs personnalisés.
- Définissez la valeur des attributs personnalisés immuables dans les demandes d'API `AdminCreateUser`. Cette fonctionnalité n'est pas disponible dans la console Amazon Cognito.
- Spécifiez le mot de passe temporaire ou laissez Amazon Cognito en générer un automatiquement.
- Indiquer si les adresses e-mail et numéros de téléphone fournis sont marqués comme étant vérifiés pour les nouveaux utilisateurs.
- Spécifiez des messages d'invitation personnalisés par SMS et e-mail pour les nouveaux utilisateurs en utilisant la AWS Management Console ou un déclencheur Lambda Message personnalisé. Pour de plus amples informations, veuillez consulter [Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda](#).
- Spécifier si les messages d'invitation sont envoyés par SMS, e-mail ou les deux.
- Renvoyer le message de bienvenue à un utilisateur existant en appelant l'API `AdminCreateUser`, en spécifiant `RESEND` pour le paramètre `MessageAction`.

Note

Cette action ne peut pas être effectuée actuellement en utilisant AWS Management Console.

- Supprimer l'envoi du message d'invitation lorsque l'utilisateur est créé.
- Spécifier un délai d'expiration pour le compte utilisateur (jusqu'à 90 jours).
- Permettre aux utilisateurs de s'inscrire ou exiger que les nouveaux utilisateurs soient uniquement ajoutés par l'administrateur.

Flux d'authentification pour les utilisateurs créés par des développeurs ou des administrateurs

Le flux d'authentification pour ces utilisateurs inclut une étape supplémentaire pour envoyer le nouveau mot de passe et fournir les valeurs manquantes pour les attributs obligatoires. Les différentes étapes sont décrites ci-après, et les étapes 5, 6 et 7 sont spécifiques à ces utilisateurs.

1. L'utilisateur commence par se connecter pour la première fois en soumettant son nom d'utilisateur et son mot de passe.
2. Le kit SDK appelle `InitiateAuth(Username, USER_SRP_AUTH)`.
3. Amazon Cognito renvoie la stimulation `PASSWORD_VERIFIER` avec le bloc Salt & Secret.
4. Le kit SDK effectue les calculs SRP et appelle `RespondToAuthChallenge(Username, <SRP variables>, PASSWORD_VERIFIER)`.
5. Amazon Cognito renvoie la stimulation `NEW_PASSWORD_REQUIRED`. Le corps de ce défi inclut les attributs actuels de l'utilisateur et tous les attributs requis dans votre groupe d'utilisateurs qui n'ont actuellement aucune valeur dans le profil de l'utilisateur. Pour de plus amples informations, veuillez consulter [RespondToAuthChallenge](#).
6. L'utilisateur est invité à entrer un nouveau mot de passe et toutes les valeurs manquantes pour les attributs obligatoires.
7. Le kit SDK appelle `RespondToAuthChallenge(Username, <New password>, <User attributes>)`.
8. Si l'utilisateur a besoin d'un deuxième facteur pour l'authentification multi-facteurs (MFA), Amazon Cognito renvoie la stimulation `SMS_MFA` et le code est soumis.

9. Une fois que l'utilisateur est parvenu à changer son mot de passe et qu'il a fourni, le cas échéant, les valeurs des attributs requis ou effectué l'authentification MFA, il est connecté, et des jetons sont émis.

Lorsque l'utilisateur a répondu à toutes les stimulations, le service Amazon Cognito marque l'utilisateur comme confirmé, émet l'ID et l'accès, et actualise les jetons pour l'utilisateur. Pour de plus amples informations, veuillez consulter [Utilisation des jetons avec des groupes d'utilisateurs](#).

Création d'un nouvel utilisateur dans AWS Management Console

Vous pouvez définir les exigences relatives au mot de passe utilisateur, configurer les messages d'invitation et de vérification envoyés aux utilisateurs et ajouter de nouveaux utilisateurs avec la console Amazon Cognito.

Définir une politique de mot de passe et activer l'auto-enregistrement

Vous pouvez configurer des paramètres pour une complexité minimale des mots de passe et indiquer si les utilisateurs peuvent s'inscrire à l'aide d'API publiques de votre groupe d'utilisateurs.

Configurer une politique de mot de passe

1. Accédez à la [console Amazon Cognito](#), puis choisissez Groupes d'utilisateurs.
2. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
3. Choisissez l'onglet Sign-in experience (Expérience de connexion) et localisez Password policy (Politique de mots de passe). Choisissez Modifier.
4. Choisissez un Mode politique de mot de passe sur Personnalisé.
5. Choisissez une Longueur minimum du mot de passe. Pour connaître les limites de la longueur de mot de passe requise, consultez [Quotas de ressources pour les groupes](#).
6. Choisissez une exigence de Complexité pour le mot de passe.
7. Choisissez la durée pendant laquelle le mot de passe défini par les administrateurs doit être valide.
8. Choisissez Enregistrer les modifications.

Autoriser l'inscription en libre-service

1. Accédez à la [console Amazon Cognito](#), puis choisissez Groupes d'utilisateurs.
2. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).

3. Choisissez l'onglet Sign-up experience (Expérience d'inscription) et localisez Self-service sign-up (Inscription en libre-service). Tâche de sélection Modifier.
4. Choisissez si vous souhaitez Activer l'auto-enregistrement. L'auto-enregistrement est généralement utilisé avec les clients d'applications publiques qui doivent enregistrer de nouveaux utilisateurs dans votre groupe d'utilisateurs sans distribuer un secret client ni des informations d'identification d'API AWS Identity and Access Management (IAM).

Désactivation de l'auto-enregistrement

Si vous n'activez pas l'auto-enregistrement, les nouveaux utilisateurs doivent être créés par des actions d'API administratives utilisant les informations d'identification d'API IAM ou en se connectant avec des fournisseurs fédérés.

5. Choisissez Enregistrer les modifications.

Personnalisation des e-mails et SMS

Personnalisation des messages utilisateurs

Vous pouvez personnaliser les messages qu'Amazon Cognito envoie à vos utilisateurs lorsque vous les invitez à se connecter, qu'ils s'inscrivent à un compte d'utilisateur ou qu'ils se connectent et sont invités à effectuer une authentification multifacteur (MFA).

Note


Un Message d'invitation est envoyé lorsque vous créez un utilisateur dans votre groupe d'utilisateurs et invitez cet utilisateur à se connecter. Amazon Cognito envoie les informations de connexion initiales à l'adresse e-mail ou au numéro de téléphone de l'utilisateur.

Un Message de vérification est envoyé lorsqu'un utilisateur s'inscrit à un compte utilisateur dans votre groupe d'utilisateurs. Amazon Cognito envoie un code à l'utilisateur. Lorsque l'utilisateur fournit le code à Amazon Cognito, il vérifie ses coordonnées et confirme la connexion de son compte. Les codes de vérification restent valables pendant 24 heures.

Un Message MFA est envoyé lorsque vous activez SMS MFA dans votre groupe d'utilisateurs, et qu'un utilisateur qui a configuré SMS MFA se connecte et est invité à entrer MFA.

1. Accédez à la [console Amazon Cognito](#), puis choisissez Groupes d'utilisateurs.

2. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
3. Choisissez l'onglet Messaging (Messagerie) et localisez les Message templates (Modèles de messages). Sélectionnez Verification messages (Messages de vérification), Invitation messages (Messages d'invitation), ou les MFA messages (Messages MFA) et choisissez Edit (Modifier).
4. Personnalisez les messages pour le type de message choisi.

 Note

Toutes les variables des modèles de messages doivent être incluses lorsque vous personnalisez le message. Si la variable, par exemple {#####}, n'est pas inclus, votre utilisateur ne disposera pas d'informations suffisantes pour terminer l'action de message. Pour plus d'informations, consultez [Modèles de messages](#).

5. a. Messages de vérification
 - i. Choisissez un Type de vérification pour les messages e-mail. Un Code de vérification transfère un code numérique que l'utilisateur doit entrer. Un Lien de vérification transfère un lien sur lequel l'utilisateur peut cliquer pour vérifier ses coordonnées. Le texte de la variable d'un Lien du message s'affiche sous forme de texte de lien hypertexte. Par exemple, un modèle de message utilisant la variable {# #Click ici##} s'affiche sous la forme [Cliquez ici](#) dans l'e-mail.
 - ii. Saisissez un Objet de l'e-mail pour les messages e-mail.
 - iii. Saisissez un modèle de message e-mail personnalisé pour des messages e-mail. Vous pouvez personnaliser ce modèle avec du code HTML.
 - iv. Saisissez une personnalisation pour modèle de Message SMS pour SMS.
 - v. Choisissez Enregistrer les modifications.
- b. Messages d'invitation
 - i. Saisissez un Objet de l'e-mail pour les messages e-mail.
 - ii. Saisissez un modèle de message e-mail personnalisé pour des messages e-mail. Vous pouvez personnaliser ce modèle avec du code HTML.
 - iii. Saisissez une personnalisation pour modèle de Message SMS pour SMS.
 - iv. Choisissez Enregistrer les modifications.
- c. Messages MFA
 - i. Saisissez une personnalisation pour modèle de Message SMS pour SMS.

ii. Choisissez Enregistrer les modifications.

Créez un utilisateur

Créez un utilisateur

Vous pouvez créer de nouveaux utilisateurs pour votre groupe d'utilisateurs à partir de la console Amazon Cognito. En règle générale, les utilisateurs peuvent se connecter après avoir défini un mot de passe. Pour se connecter avec une adresse e-mail ou un numéro de téléphone, un utilisateur doit vérifier l'attribut `email`. Pour se connecter avec un numéro de téléphone, l'utilisateur doit vérifier l'attribut `phone_number`. Pour confirmer les comptes en tant qu'administrateur, vous pouvez également utiliser l'interface AWS CLI ou l'API, ou créer des profils utilisateur avec un fournisseur d'identité fédéré. Pour plus d'informations, consultez la rubrique [Référence d'API Amazon Cognito](#).

1. Accédez à la [console Amazon Cognito](#), puis choisissez Groupes d'utilisateurs.
2. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
3. Choisissez l'onglet Users (Utilisateurs), puis Create a user (Créer un utilisateur).
4. Vérifiez l'option User pool sign-in and security requirements (Exigences de sécurité et de connexion au groupe d'utilisateurs) pour obtenir des conseils sur les exigences de mot de passe, les méthodes de récupération de compte disponibles et les attributs d'alias pour votre groupe d'utilisateurs.
5. Choisissez comment envoyer un message d'invitation. Choisissez par SMS, par e-mail ou les deux.

Note

Avant de pouvoir envoyer des messages d'invitation, configurez un expéditeur et une Région AWS avec Amazon Simple Notification Service et Amazon Simple Email Service dans l'onglet Messaging (Messagerie) de votre groupe d'utilisateurs. Les tarifs des messages et des données du destinataire s'appliquent. Amazon SES facture séparément les e-mails et Amazon SNS facture séparément les SMS.

6. Choisissez un Nom d'utilisateur pour le nouvel utilisateur.
7. Choisissez si vous souhaitez créer un mot de passe ou autoriser Amazon Cognito à générer un mot de passe pour l'utilisateur. Tout mot de passe temporaire doit respecter la politique de mot de passe du groupe d'utilisateurs.

8. Choisissez Créer.
9. Choisissez l'onglet Users (Utilisateurs) et choisissez l'entrée User name (Nom d'utilisateur) pour l'utilisateur. Ajoutez et modifiez des User attributes (Attributs utilisateur) et des Group memberships (Appartenances de groupe). Passez en revue l'historique des événements d'utilisateur.

Ajout de groupes à un groupe d'utilisateurs

La prise en charge des groupes dans les groupes d'utilisateurs Amazon Cognito vous permet de créer et gérer des groupes, d'ajouter des utilisateurs à des groupes et de supprimer des utilisateurs de groupes. Utilisez des groupes pour créer des ensembles d'utilisateurs afin de gérer leurs autorisations ou de représenter différents types d'utilisateurs. Vous pouvez attribuer un rôle AWS Identity and Access Management (IAM) à un groupe afin de définir les autorisations accordées aux membres d'un groupe.

Vous pouvez utiliser des groupes pour créer un ensemble d'utilisateurs dans un pool d'utilisateurs, ce qui est souvent effectué pour définir les autorisations pour ces utilisateurs. Par exemple, vous pouvez créer des groupes distincts pour les utilisateurs qui sont des lecteurs, des contributeurs et des éditeurs de votre site Web et de votre application. À l'aide du rôle IAM associé à un groupe, vous pouvez également définir des autorisations différentes pour ces différents groupes afin que seuls des contributeurs puissent placer des contenus dans Amazon S3, et que seuls des éditeurs puissent publier des contenus via une API dans Amazon API Gateway.

Vous pouvez créer et gérer des groupes dans un groupe d'utilisateurs à partir des AWS Management Console API et de la CLI. En tant que développeur (à l'aide des AWS informations d'identification), vous pouvez créer, lire, mettre à jour, supprimer et répertorier les groupes d'un groupe d'utilisateurs. Vous pouvez également ajouter et supprimer des utilisateurs dans des groupes.

L'utilisation de groupes au sein d'un pool d'utilisateurs n'entraîne aucun coût supplémentaire. Pour plus d'informations, consultez [Tarification Amazon Cognito](#).

Attribution de rôles IAM à des groupes

Vous pouvez utiliser des groupes pour contrôler les autorisations sur vos ressources à l'aide d'un rôle IAM. Les rôles IAM incluent des politiques d'approbation et des politiques d'autorisation. La politique [d'approbation](#) de rôle spécifie qui peut utiliser le rôle. Les politiques [d'autorisations](#) spécifient les actions et les ressources auxquelles les membres de votre groupe peuvent accéder. Lorsque vous créez un rôle IAM, configurez la politique de confiance du rôle pour permettre aux utilisateurs de votre

groupe d'assumer le rôle. Dans les politiques d'autorisations de rôle, spécifiez les autorisations que vous souhaitez attribuer à votre groupe.

Lorsque vous créez un groupe dans Amazon Cognito, vous spécifiez un rôle IAM en fournissant l'[ARN](#) du rôle. Lorsque les membres du groupe se connectent à l'aide d'Amazon Cognito, ils peuvent recevoir des informations d'identification temporaires à partir des groupes d'identités. Leurs autorisations sont déterminées par le rôle IAM associé.

Des utilisateurs individuels peuvent appartenir à plusieurs groupes. En tant que développeur, vous disposez des options suivantes pour choisir automatiquement le rôle IAM lorsqu'un utilisateur appartient à plusieurs groupes :

- Vous pouvez affecter des valeurs de priorité à chaque groupe. Le groupe prioritaire (valeur la plus faible) sera choisi et son rôle IAM associé sera appliqué.
- Votre application peut également choisir parmi les rôles disponibles lorsqu'elle demande les AWS informations d'identification d'un utilisateur via un pool d'identités, en spécifiant un ARN de rôle dans le [GetCredentialsForIdentityCustomRoleARN](#) paramètre. Le rôle IAM spécifié doit correspondre à un rôle qui est disponible pour l'utilisateur.

Affectation de valeurs de priorité à des groupes

Un utilisateur peut appartenir à plusieurs groupes. Dans les jetons d'accès et d'ID de l'utilisateur, la demande `cognito:groups` contient la liste de tous les groupes auxquels l'utilisateur appartient. La demande `cognito:roles` contient la liste des rôles correspondant aux groupes.

Comme un utilisateur peut appartenir à plusieurs groupes, une priorité peut être affectée à chaque groupe. Il s'agit d'un nombre non négatif qui spécifie la priorité de ce groupe par rapport à d'autres groupes auxquels un utilisateur appartient au sein du groupe d'utilisateurs. Zéro est la valeur de priorité la plus haute. Les groupes avec des valeurs de priorité plus faibles sont prioritaires par rapport aux groupes avec des valeurs de priorité plus élevées ou null. Si un utilisateur appartient à deux groupes ou plus, le groupe ayant la valeur de priorité la plus basse verra son rôle IAM appliqué à la `cognito:preferred_role` demande dans le jeton d'identification de l'utilisateur.

Deux groupes peuvent avoir la même valeur de priorité. Si cela se produit, aucun des deux groupes n'est prioritaire sur l'autre. Si deux groupes avec la même valeur de priorité ont le même ARN de rôle, ce rôle est utilisé dans la demande `cognito:preferred_role` dans les jetons d'ID pour les utilisateurs dans chaque groupe. Si les deux groupes ont des ARN de rôle différents, la demande `cognito:preferred_role` n'est pas définie dans les jetons d'ID des utilisateurs.

Utilisation de groupes pour contrôler une autorisation avec Amazon API Gateway

Vous pouvez utiliser des groupes dans un groupe d'utilisateurs pour contrôler une autorisation avec Amazon API Gateway. Les groupes dont un utilisateur est un membre sont inclus dans le jeton d'ID et dans le jeton d'accès d'un groupe d'utilisateurs dans la demande `cognito:groups`. Vous pouvez soumettre un ID ou des jetons d'accès avec des demandes à Amazon API Gateway et utiliser un mécanisme d'autorisation de groupe d'utilisateurs Amazon Cognito pour une API REST. Pour plus d'informations, consultez [Contrôle de l'accès à une API REST à l'aide de groupes d'utilisateurs Amazon Cognito en tant que mécanisme d'autorisation](#) dans le [Guide du développeur API Gateway](#).

Vous pouvez également autoriser l'accès à une API HTTP Amazon API Gateway avec un mécanisme d'autorisation JWT personnalisé. Pour plus d'informations, consultez [Contrôle de l'accès aux API HTTP avec les mécanismes d'autorisation JWT](#) dans le [Guide du développeur API Gateway](#).

Limitations sur les groupes

Les groupes d'utilisateurs sont soumis aux limitations suivantes :

- Le nombre de groupes que vous pouvez créer est limité par les quotas du [service Amazon Cognito](#).
- Les groupes ne peuvent pas être imbriqués.
- Vous ne pouvez pas rechercher des utilisateurs dans un groupe.
- Vous ne pouvez pas rechercher des groupes par nom, mais vous pouvez afficher les groupes.

Création d'un nouveau groupe dans la AWS Management Console

Utilisez la procédure suivante pour créer un nouveau groupe.

Pour créer un nouveau groupe.

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste.
4. Choisissez l'onglet Groupes puis choisissez Créer un groupe.
5. Dans la page Créez un groupe, dans Nom du groupe, saisissez un nom convivial pour votre nouveau groupe.

6. Vous pouvez éventuellement fournir des informations supplémentaires sur ce groupe à l'aide de l'un des champs suivants :
 - Description – Saisissez des informations sur la raison pour laquelle ce nouveau groupe sera utilisé.
 - Priorité – Amazon Cognito évalue et applique toutes les autorisations de groupe pour un utilisateur donné en fonction des groupes auxquels il appartient à une valeur de priorité inférieure. Le groupe prioritaire sera choisi et son rôle IAM associé sera appliqué. Pour de plus amples informations, veuillez consulter [Affectation de valeurs de priorité à des groupes](#).
 - Rôle IAM – Vous pouvez attribuer un rôle IAM à votre groupe lorsque vous devez contrôler les autorisations sur vos ressources. Si vous intégrez un groupe d'utilisateurs à un groupe d'identités, le paramètre IAM role (Rôle IAM) détermine le rôle qui est attribué au jeton d'ID de l'utilisateur si le groupe d'identités est configuré pour choisir le rôle à partir du jeton. Pour de plus amples informations, veuillez consulter [Attribution de rôles IAM à des groupes](#).
 - Ajoutez des utilisateurs au groupe – Ajoutez des utilisateurs existants en tant que membres de ce groupe après sa création.
7. Choisissez Créer pour confirmer.

Gestion et recherche de comptes d'utilisateur

Une fois que vous avez créé un groupe d'utilisateurs, vous pouvez afficher et gérer les utilisateurs via la AWS Management Console, l'AWS Command Line Interface ou l'API Amazon Cognito. Cette rubrique décrit comment afficher et rechercher des utilisateurs via AWS Management Console.

Affichage des attributs utilisateur

Utilisez la procédure suivante pour consulter les attributs d'un utilisateur dans la console Amazon Cognito.

Pour consulter les attributs d'un utilisateur

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos informations d'identification AWS.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste.
4. Choisissez l'onglet Users (Utilisateurs), puis sélectionnez un utilisateur dans la liste.

5. Sur la page des détails de l'utilisateur, sous Attributs utilisateur, vous pouvez voir quels attributs sont associés à l'utilisateur.

Réinitialisation du mot de passe d'un utilisateur

Utilisez la procédure suivante pour réinitialiser le mot de passe d'un utilisateur dans la console Amazon Cognito.

Pour réinitialiser le mot de passe d'un utilisateur

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos informations d'identification AWS.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste.
4. Choisissez l'onglet Users (Utilisateurs), puis sélectionnez un utilisateur dans la liste.
5. Sur la page des détails de l'utilisateur, choisissez Actions, Réinitialiser le mot de passe.
6. Dans Réinitialiser le mot de passe, vérifiez les informations et, lorsque vous êtes prêt, choisissez Réinitialiser.

Cette action entraîne immédiatement l'envoi d'un code de confirmation à l'utilisateur et désactive le mot de passe actuel de l'utilisateur en changeant l'état de l'utilisateur en RESET_REQUIRED. Le code de (réinitialisation du mot de passe) est valide pendant 1 heure.

Recherche d'attributs utilisateur

Si vous avez déjà créé un groupe d'utilisateurs, vous pouvez effectuer des recherches dans le volet Users (Utilisateurs) d'AWS Management Console. Vous pouvez également utiliser l'[API ListUsers](#) d'Amazon Cognito qui accepte un paramètre Filter (Filtre).

Vous pouvez rechercher les attributs standard suivants, mais pas les attributs personnalisés.

- username (sensible à la casse)
- e-mail
- phone_number
- name
- given_name

- family_name
- preferred_username
- cognito:user_status (appelé Status dans la console) (non sensible à la casse)
- statut (appelé Enabled dans la console) (sensible à la casse)
- sub

Note

Vous pouvez également répertorier les utilisateurs avec un filtre côté client. Le filtre côté serveur ne correspond pas à plus d'un attribut. Pour la recherche avancée, utilisez un filtre côté client avec le `--queryparamètre` de `list-users` dans le AWS Command Line Interface. Lorsque vous utilisez un filtre côté client, ListUsers renvoie une liste paginée de zéro utilisateur ou plus. Vous pouvez recevoir plusieurs pages consécutives ne contenant aucun résultat. Répétez la requête avec chaque jeton de pagination renvoyé jusqu'à ce que vous receviez une valeur de jeton de pagination NULL, puis passez en revue le résultat combiné.

Pour de plus amples informations sur le filtrage côté serveur et côté client, veuillez consulter [FiltrageAWS CLI sortie](#) dans le AWS Command Line Interface Guide de l'utilisateur .

Recherche d'utilisateurs via AWS Management Console

Si vous avez déjà créé un groupe d'utilisateurs, vous pouvez effectuer des recherches dans le volet Users (Utilisateurs) d'AWS Management Console.

Les recherches AWS Management Console sont toujours des recherches à préfixe (« commence par »).

Pour rechercher un utilisateur dans la console Amazon Cognito

1. Accédez à la [console Amazon Cognito](#). Il se peut que vous soyez invité à saisir vos informations d'identification AWS.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste.

4. Choisissez l'onglet Users (Utilisateurs), puis saisissez le nom d'un utilisateur dans le champ de recherche. Notez que certaines valeurs d'attribut sont sensibles à la casse (Nom d'utilisateur, par exemple).

Vous pouvez également trouver des utilisateurs en ajustant le filtre de recherche pour réduire le périmètre aux autres propriétés de l'utilisateur, comme Adresse de messagerie, Numéro de téléphone, ou Nom.

Recherche d'utilisateurs via l'API **ListUsers**

Pour rechercher les utilisateurs de votre application, utilisez l'[API ListUsers](#) d'Amazon Cognito. Cette API utilise les paramètres suivants :

- **AttributesToGet** : tableau de chaînes, où chaque chaîne correspond au nom d'un attribut à renvoyer pour chaque utilisateur dans les résultats de recherche. Pour récupérer tous les attributs, n'incluez pas de paramètre **AttributesToGet** ni de demande **AttributesToGet** avec une valeur de la chaîne littérale `null`.
- **Filter** : chaîne de filtrage représentée sous la forme "AttributeName Filter-Type AttributeValue". Les guillemets utilisés dans la chaîne de filtrage doivent être précédés d'une barre oblique inverse (`\`). Par exemple, "family_name = \"Reddy\"". Si la chaîne de filtrage est vide, **ListUsers** renvoie tous les utilisateurs du groupe.
- **AttributeName** : nom de l'attribut à rechercher. Vous pouvez uniquement rechercher un attribut à la fois.

Note

Vous pouvez ne rechercher que des attributs standard, mais pas les attributs personnalisés. La raison en est que seuls les attributs indexés peuvent être explorés et que les attributs personnalisés ne peuvent pas être indexés.

- **Filter-Type** : pour une correspondance exacte, utilisez `=` (par exemple, `given_name = "Jon"`). Pour une correspondance sur le préfixe (« commence par »), utilisez `^=` (par exemple, `given_name ^= "Jon"`).
- **AttributeValue** : valeur d'attribut qui doit correspondre pour chaque utilisateur.
- **Limit** : nombre maximum d'utilisateurs à renvoyer.

- `PaginationToken` : jeton permettant d'obtenir plus de résultats à partir d'une recherche précédente. Amazon Cognito fait expirer le jeton de pagination au bout d'une heure.
- `UserPoolId` : ID du groupe d'utilisateurs sur lequel la recherche doit porter.

Toutes les recherches sont sensible à la casse. Les résultats de recherche sont triés en fonction de l'attribut dénommé par la chaîne `AttributeName`, dans l'ordre croissant.

Exemples d'utilisation de l'API `ListUsers`

L'exemple suivant renvoie tous les utilisateurs et inclut tous les attributs.

```
{
  "AttributesToGet": null,
  "Filter": "",
  "Limit": 10,
  "UserPoolId": "us-east-1_samplepool"
}
```

L'exemple suivant renvoie tous les utilisateurs dont le numéro de téléphone commence par « +1312 », et inclut tous les attributs.

```
{
  "AttributesToGet": null,
  "Filter": "phone_number ^= \"+1312\"",
  "Limit": 10,
  "UserPoolId": "us-east-1_samplepool"
}
```

L'exemple suivant retourne les 10 premiers utilisateurs dont le nom de famille est « Reddy ». Pour chaque utilisateur, les résultats de recherche incluent son nom, son numéro de téléphone et son adresse e-mail. Si le groupe d'utilisateurs compte plus de 10 utilisateurs correspondants, la réponse inclut un jeton de pagination.

```
{
  "AttributesToGet": [
    "given_name",
```

```
    "phone_number",
    "email"
  ],
  "Filter": "family_name = \"Reddy\"",
  "Limit": 10,
  "UserPoolId": "us-east-1_samplepool"
}
```

Si l'exemple précédent renvoie un jeton de pagination, l'exemple suivant renvoie les 10 utilisateurs suivants qui correspondent à la même chaîne de filtrage.

```
{
  "AttributesToGet": [
    "given_name",
    "phone_number",
    "email"
  ],
  "Filter": "family_name = \"Reddy\"",
  "Limit": 10,
  "PaginationToken": "pagination_token_from_previous_search",
  "UserPoolId": "us-east-1_samplepool"
}
```

Récupération de comptes d'utilisateurs

Le paramètre `AccountRecoverySetting` vous permet de personnaliser la méthode qu'un utilisateur peut employer pour récupérer son mot de passe lorsqu'il appelle l'API [ForgotPassword](#). `ForgotPassword` envoie un code de récupération à une adresse e-mail ou un numéro de téléphone vérifié. Le code de récupération reste valide pendant une heure. Lorsque vous spécifiez un [AccountRecoverySetting](#) pour votre groupe d'utilisateurs, Amazon Cognito choisit la destination de livraison de code en fonction de la priorité que vous avez définie.

Lorsque vous définissez `AccountRecoverySetting` et qu'un utilisateur a configuré SMS MFA, SMS ne peut pas être utilisé comme mécanisme de récupération de compte. La priorité de ce paramètre est déterminée et 1 est la priorité la plus élevée. Cognito envoie une vérification à une seule des méthodes spécifiées.

Par exemple, `admin_only` est une valeur utilisée lorsque l'administrateur ne souhaite pas que l'utilisateur récupère lui-même son compte et lui demande de contacter l'administrateur pour le réinitialiser. Vous ne pouvez pas utiliser `admin_only` avec aucun autre mécanisme de récupération de compte.

Si vous ne spécifiez pas `AccountRecoverySetting`, Amazon Cognito utilise le mécanisme hérité pour déterminer la méthode de récupération du mot de passe. Dans ce cas, Cognito utilise d'abord un téléphone vérifié. Si le téléphone vérifié n'est pas trouvé pour l'utilisateur, Cognito revient en arrière et utilise l'e-mail vérifié.

Pour plus d'informations sur `AccountRecoverySetting`, consultez [CreateUserPool](#) et [UpdateUserPool](#) dans le manuel Amazon Cognito Identity Provider API Reference.

Comportement en cas d'oubli de mot de passe

Dans une heure donnée, nous accordons aux utilisateurs entre 5 et 20 tentatives pour demander ou entrer un code de réinitialisation de mot de passe dans le cadre des actions `forgot-password` et `confirm-forgot-password`. La valeur exacte dépend des paramètres de risque associés aux demandes. Veuillez noter que ce comportement est sujet à changement.

Importation d'utilisateurs dans un groupe d'utilisateurs

Il existe deux façons d'importer ou de migrer des utilisateurs à partir de votre répertoire d'utilisateurs ou de votre base de données d'utilisateurs existants vers des groupes d'utilisateurs Amazon Cognito. Vous pouvez migrer les utilisateurs lorsqu'ils se connectent à l'aide d'Amazon Cognito pour la première fois. Pour ce faire, utilisez un déclencheur Lambda Migration d'utilisateur. Avec cette approche, les utilisateurs peuvent continuer à utiliser leurs mots de passe. Ils ne devront pas les réinitialiser à l'issue de la migration vers votre groupe d'utilisateurs. Vous pouvez également migrer des utilisateurs en masse en chargeant un fichier CSV contenant les attributs de profil de tous les utilisateurs. Les sections suivantes décrivent ces deux approches.

Rubriques

- [Importation d'utilisateurs dans des groupes d'utilisateurs avec un déclencheur Lambda de migration d'utilisateur](#)
- [Importation d'utilisateurs dans des groupes d'utilisateurs depuis un fichier CSV](#)

Importation d'utilisateurs dans des groupes d'utilisateurs avec un déclencheur Lambda de migration d'utilisateur

Cette approche vous permet de migrer en toute transparence des utilisateurs de votre répertoire utilisateur existant vers des groupes d'utilisateurs quand ces utilisateurs se connectent pour la première fois avec votre application ou demandent la réinitialisation de leur mot de passe. Ajoutez

une fonction [Déclencheur Lambda de migration d'utilisateur](#) dans votre groupe d'utilisateurs pour qu'elle reçoive des métadonnées sur les utilisateurs qui tentent de se connecter et renvoie les informations de profil utilisateur à partir d'une source d'identité externe. Pour obtenir plus d'informations et un exemple de code pour ce déclencheur Lambda, y compris les paramètres de demande et de réponse, consultez [Paramètres du déclencheur Lambda Migration d'utilisateur](#).

Avant de commencer à migrer des utilisateurs, créez une fonction Lambda de migration d'utilisateur dans votre Compte AWS et définissez cette fonction Lambda comme déclencheur de migration d'utilisateur dans votre groupe d'utilisateurs . Ajoutez une politique d'autorisation à votre fonction Lambda pour autoriser uniquement le principal du compte de service Amazon Cognito, `cognito-idp.amazonaws.com`, à appeler la fonction Lambda, et cela uniquement dans le contexte de votre propre groupe d'utilisateurs. Pour plus d'informations, consultez [Utilisation de stratégies basées sur les ressources pour AWS Lambda \(stratégies de fonction Lambda\)](#).

Processus de connexion

1. L'utilisateur ouvre votre application et se connecte avec l'API des groupes d'utilisateurs Amazon Cognito ou via l'interface utilisateur hébergée d'Amazon Cognito. Pour plus d'informations sur la façon de faciliter la connexion avec les API Amazon Cognito, consultez [Intégration de l'authentification et de l'autorisation Amazon Cognito avec des applications Web et mobiles](#).
2. Votre application envoie le nom d'utilisateur et le mot de passe à Amazon Cognito. Si votre application dispose d'une interface utilisateur de connexion personnalisée que vous avez créée avec un kit AWS SDK, l'application doit utiliser [InitiateAuth](#) ou [AdminInitiateAuth](#) avec le flux `USER_PASSWORD_AUTH` ou `ADMIN_USER_PASSWORD_AUTH`. Lorsque votre application utilise l'un de ces flux, le kit SDK envoie le mot de passe au serveur.


Note

Avant d'ajouter un déclencheur de migration d'utilisateur, activez le flux `USER_PASSWORD_AUTH` ou `ADMIN_USER_PASSWORD_AUTH` dans les paramètres de votre client d'application. Vous devez utiliser ces flux à la place du flux `USER_SRP_AUTH` par défaut. Amazon Cognito doit envoyer un mot de passe à votre fonction Lambda afin qu'elle puisse vérifier l'authentification de votre utilisateur dans l'autre répertoire. Le protocole SRP occulte le mot de passe de votre utilisateur de votre fonction Lambda.

3. Amazon Cognito vérifie si le nom d'utilisateur soumis correspond à un nom d'utilisateur ou à un alias dans le groupe d'utilisateurs. Vous pouvez définir l'adresse e-mail, le numéro de téléphone ou le nom d'utilisateur préféré de l'utilisateur comme alias dans votre groupe d'utilisateurs. Si

l'utilisateur n'existe pas, Amazon Cognito envoie des paramètres, y compris le nom d'utilisateur et le mot de passe, à votre fonction [Déclencheur Lambda de migration d'utilisateur](#).

4. Votre fonction [Déclencheur Lambda de migration d'utilisateur](#) vérifie ou authentifie l'utilisateur avec votre répertoire utilisateur existant ou votre base de données utilisateur existante. La fonction renvoie les attributs utilisateur qu'Amazon Cognito stocke dans le profil de l'utilisateur dans le groupe d'utilisateurs. Vous pouvez renvoyer un paramètre `username` seulement si le nom d'utilisateur soumis correspond à un attribut d'alias. Si vous souhaitez que les utilisateurs continuent d'utiliser leurs mots de passe existants, votre fonction définit l'attribut `finalUserStatus` sur `CONFIRMED` dans la réponse Lambda. Votre application doit renvoyer tous les paramètres "response" présentés dans [Paramètres du déclencheur Lambda Migration d'utilisateur](#).

 Important

Ne conservez pas l'objet d'événement de demande entier dans votre code Lambda de migration d'utilisateur. Cet objet d'événement de demande inclut le mot de passe de l'utilisateur. Si vous ne nettoyez pas les journaux, les mots de passe apparaissent dans CloudWatch Logs.

5. Amazon Cognito crée le profil utilisateur dans votre groupe d'utilisateurs et renvoie les jetons à votre client d'application.
6. Votre application récupère les jetons, accepte l'authentification utilisateur et passe au contenu demandé.

Après avoir migré vos utilisateurs, utilisez `USER_SRP_AUTH` pour la connexion. Le protocole SRP (Secure Remote Password) n'envoie pas le mot de passe sur le réseau et présente des avantages de sécurité par rapport au flux `USER_PASSWORD_AUTH` que vous utilisez durant la migration.

En cas d'erreurs pendant la migration, y compris des problèmes liés à l'appareil client ou au réseau, votre application reçoit des réponses d'erreur de l'API des groupes d'utilisateurs Amazon Cognito. Dans ce cas, Amazon Cognito peut créer ou non le compte d'utilisateur dans votre groupe d'utilisateurs. L'utilisateur doit ensuite tenter de se connecter à nouveau. En cas d'échec répété de la connexion, essayez de réinitialiser le mot de passe de l'utilisateur avec le flux de mot de passe oublié dans votre application.

Le flux de mot de passe oublié appelle également votre fonction [Déclencheur Lambda de migration d'utilisateur](#) avec une source d'événement `UserMigration_ForgotPassword`. Comme l'utilisateur

ne soumet pas de mot de passe lorsqu'il demande une réinitialisation de mot de passe, Amazon Cognito n'inclut pas de mot de passe dans l'événement envoyé à votre fonction Lambda. Votre fonction peut uniquement rechercher l'utilisateur dans votre répertoire utilisateur existant et renvoyer les attributs à ajouter au profil utilisateur dans votre groupe d'utilisateurs. Une fois que votre fonction a terminé son invocation et a renvoyé sa réponse à Amazon Cognito, votre groupe d'utilisateurs envoie un code de réinitialisation du mot de passe par e-mail ou par SMS. Dans votre application, demandez à votre utilisateur de saisir son code de confirmation et un nouveau mot de passe, puis envoyez ces informations à Amazon Cognito dans la demande d'API [ConfirmForgotPassword](#). Vous pouvez également utiliser les pages intégrées pour le flux de mot de passe oublié dans l'interface utilisateur hébergée sur Amazon Cognito.

Importation d'utilisateurs dans des groupes d'utilisateurs depuis un fichier CSV

Vous pouvez importer des utilisateurs dans un pool d'utilisateurs Amazon Cognito. Les informations utilisateur sont importées à partir d'un fichier .csv spécialement mis en forme. Le processus d'importation définit les valeurs de tous les attributs utilisateur, sauf password (mot de passe). L'importation du mot de passe n'est pas prise en charge, car les bonnes pratiques de sécurité nécessitent que les mots de passe ne soient pas disponibles sous forme de texte brut, et nous ne prenons pas en charge l'importation des hachages. Cela signifie que vos utilisateurs doivent changer leur mot de passe la première fois qu'ils se connectent. Par conséquent, si vous importez vos utilisateurs avec cette méthode, ils auront l'état `RESET_REQUIRED`.

Vous pouvez définir le mot de passe de vos utilisateurs à l'aide d'une demande d'API [AdminSetUserPassword](#) qui définit le paramètre `Permanent` sur `true`.

Note

La date de création pour chaque utilisateur est celle où celui-ci a été importé dans le pool d'utilisateurs. La date de création n'est pas l'un des attributs importés.

Les étapes de base sont les suivantes :

1. Créez un rôle Amazon CloudWatch Logs dans la console AWS Identity and Access Management (IAM).
2. Créez le fichier .csv d'importation d'utilisateurs.
3. Créez et exécutez la tâche d'importation d'utilisateurs.
4. Téléchargez le fichier .csv d'importation d'utilisateurs.

5. Démarrez et exécutez la tâche d'importation d'utilisateurs.
6. Utilisez CloudWatch pour vérifier le journal des événements.
7. Exigez des utilisateurs importés qu'ils réinitialisent leur mot de passe.

Rubriques

- [Création du rôle IAM CloudWatch Logs](#)
- [Création du fichier CSV d'importation d'utilisateurs](#)
- [Création et exécution de la tâche d'importation de groupe d'utilisateurs Amazon Cognito](#)
- [Affichage des résultats de l'importation du groupe d'utilisateurs dans la console CloudWatch](#)
- [Obligation pour les utilisateurs importés de réinitialiser leur mot de passe](#)

Création du rôle IAM CloudWatch Logs

Si vous utilisez la CLI ou l'API Amazon Cognito, vous devez créer un rôle IAM CloudWatch. La procédure suivante montre comment créer un rôle IAM dont Amazon Cognito peut se servir pour écrire les résultats de votre tâche d'importation dans CloudWatch Logs.

Note

Lorsque vous créez une tâche d'importation dans la console Amazon Cognito, vous pouvez créer le rôle IAM en même temps. Lorsque vous choisissez Create a new IAM role (Créer un nouveau rôle IAM), Amazon Cognito applique automatiquement la politique d'approbation et la politique IAM adaptées au rôle.

Pour créer le rôle IAM CloudWatch Logs pour l'importation de groupes d'utilisateurs (AWS CLI, API)

1. Connectez-vous à l'outil AWS Management Console, puis ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Créez un rôle IAM pour un Service AWS. Pour obtenir des instructions détaillées, consultez [Création d'un rôle pour un Service AWS](#) dans le Guide de l'utilisateur AWS Identity and Access Management.
 - a. Lorsque vous sélectionnez un cas d'utilisation pour votre type d'entité approuvée, choisissez n'importe quel service. Pour l'heure, Amazon Cognito ne figure pas dans les cas d'utilisation de service.

- b. Dans l'écran Add permissions (Ajouter des autorisations), choisissez Create policy (Créer une politique) et insérez la déclaration de politique suivante. Remplacez **REGION** par la Région AWS de votre groupe d'utilisateurs, par exemple us-east-1. Remplacez **ACCOUNT** par l'ID de votre Compte AWS, par exemple 111122223333.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:REGION:ACCOUNT:log-group:/aws/cognito/*"
      ]
    }
  ]
}
```

3. Comme vous n'avez pas choisi Amazon Cognito comme entité approuvée au moment de créer le rôle, vous devez maintenant modifier manuellement la relation d'approbation du rôle. Choisissez Roles (Rôles) dans le volet de navigation de la console IAM, puis choisissez le rôle que vous avez créé.
4. Choisissez l'onglet Trust relationships.
5. Choisissez Edit trust policy (Modifier la politique).
6. Collez la déclaration de politique suivante dans Edit trust policy (Modifier la politique d'approbation), en remplaçant le texte existant éventuel :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cognito-idp.amazonaws.com"
      },

```

```
        "Action": "sts:AssumeRole"
      }
    ]
  }
```

7. Choisissez Update policy (Mettre à jour une politique).
8. Notez l'ARN du rôle. Vous devrez fournir l'ARN au moment de créer votre tâche d'importation.

Création du fichier CSV d'importation d'utilisateurs

Avant de pouvoir importer des utilisateurs existants dans votre groupe d'utilisateurs, vous devez d'abord créer un fichier CSV (valeurs séparées par des virgules) contenant les utilisateurs que vous souhaitez importer ainsi que leurs attributs. À partir de votre groupe d'utilisateurs, vous pouvez extraire un fichier d'importation d'utilisateurs dont les en-têtes reflètent le schéma d'attributs de votre groupe d'utilisateurs. Vous pouvez ensuite insérer les informations utilisateur qui répondent aux exigences de mise en forme décrites dans [Mise en forme du fichier CSV](#).

Téléchargement de l'en-tête du fichier CSV (console)

Utilisez la procédure suivante pour télécharger le fichier d'en-tête CSV.

Pour télécharger l'en-tête du fichier CSV

1. Accédez à la [console Amazon Cognito](#). Il se peut que vous soyez invité à saisir vos informations d'identification AWS.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste.
4. Sélectionnez l'onglet Utilisateurs.
5. Dans la section Import users (Importer des utilisateurs), choisissez Create an import job (Créer une tâche d'importation).
6. Sous Upload CSV (Charger un fichier CSV), sélectionnez le lien template.csv et téléchargez le fichier CSV.

Téléchargement de l'en-tête du fichier CSV (AWS CLI)

Pour obtenir la liste des en-têtes corrects, exécutez la commande de CLI, où *USER_POOL_ID* est l'identifiant du groupe d'utilisateurs vers lequel vous allez importer les utilisateurs :

```
aws cognito-idp get-csv-header --user-pool-id "USER_POOL_ID"
```

Exemple de réponse :

```
{
  "CSVHeader": [
    "name",
    "given_name",
    "family_name",
    "middle_name",
    "nickname",
    "preferred_username",
    "profile",
    "picture",
    "website",
    "email",
    "email_verified",
    "gender",
    "birthdate",
    "zoneinfo",
    "locale",
    "phone_number",
    "phone_number_verified",
    "address",
    "updated_at",
    "cognito:mfa_enabled",
    "cognito:username"
  ],
  "UserPoolId": "USER_POOL_ID"
}
```

Mise en forme du fichier CSV

Une fois téléchargé, le fichier d'en-tête CSV d'importation d'utilisateurs ressemble à la chaîne suivante. Il comporte également les attributs personnalisés que vous avez éventuellement ajoutés à votre groupe d'utilisateurs.


```
cognito:username,name,given_name,family_name,middle_name,nickname,preferred_username,profile,pi
```

Modifiez votre fichier CSV de telle sorte qu'il contienne cet en-tête et les valeurs d'attribut pour vos utilisateurs et qu'il soit mis en forme selon les règles suivantes :

 Note

Pour plus d'informations sur les valeurs d'attributs, par exemple, un format approprié pour les numéros de téléphone, consultez [Attributs de groupe d'utilisateurs](#).

- La première ligne du fichier est la ligne d'en-tête téléchargée qui contient les noms des attributs utilisateur.
- L'ordre des colonnes dans le fichier CSV n'a pas d'importance.
- Chaque ligne après la première ligne contient les valeurs d'attribut d'un utilisateur.
- Toutes les colonnes de l'en-tête doivent être présentes, mais vous n'avez pas besoin de fournir des valeurs dans chaque colonne.
- Les attributs suivants sont obligatoires :
 - `cognito:username`
 - `cognito:mfa_enabled`
 - `email_verified` ou `phone_number_verified`
 - Au moins l'un des attributs à vérification automatique doit avoir la valeur `true` pour chaque utilisateur. Un attribut à vérification automatique est une adresse e-mail ou un numéro de téléphone auxquels Amazon Cognito envoie automatiquement un code lorsqu'un nouvel utilisateur rejoint votre groupe d'utilisateurs.
 - Le groupe d'utilisateurs doit avoir au moins un attribut à vérification automatique, que ce soit `email_verified` ou `phone_number_verified`. Si le pool d'utilisateurs n'a pas d'attributs à vérification automatique, la tâche d'importation ne démarre pas.
 - Si le pool d'utilisateurs n'a qu'un seul attribut à vérification automatique, cet attribut doit être vérifié pour chaque utilisateur. Par exemple, si le groupe d'utilisateurs n'a que `phone_number` comme attribut à vérification automatique, la valeur de `phone_number_verified` doit être `true` pour chaque utilisateur.

 Note

Pour pouvoir réinitialiser leur mot de passe, les utilisateurs doivent disposer d'une adresse e-mail ou d'un numéro de téléphone vérifiés. Amazon Cognito envoie un message contenant un code de réinitialisation de mot de passe à l'adresse e-mail ou au numéro de téléphone spécifiés dans le fichier CSV. Si le message est envoyé au

numéro de téléphone, il est envoyé par SMS. Pour de plus amples informations, veuillez consulter [Vérification des coordonnées à l'inscription](#).

- email (si `email_verified` a la valeur `true`)
- phone_number (si `phone_number_verified` a la valeur `true`)
- Tous les attributs que vous avez marqués comme obligatoires lorsque vous avez créé le pool d'utilisateurs
- Les valeurs d'attribut qui sont des chaînes ne doivent pas être entre guillemets.
- Si une valeur d'attribut contient une virgule, vous devez placer une barre oblique inverse (\) devant la virgule. La raison en est que les champs au sein d'un fichier CSV sont séparés par des virgules.
- Le contenu du fichier CSV doit être au format UTF-8 sans marque d'ordre d'octet.
- Le champ `cognito:username` est obligatoire et doit être unique au sein de votre groupe d'utilisateurs. Ce peut être n'importe quelle chaîne Unicode. Cependant, elle ne peut pas comporter d'espaces ou de tabulations.
- La valeur `birthdate`, si elle est présente, doit être au format `mm/jj/aaaa`. Cela signifie, par exemple, que la date du 1er février 1985 doit être codée sous la forme **02/01/1985**.
- Le champ `cognito:mfa_enabled` est obligatoire. Si vous avez défini l'authentification multi-facteur (MFA, Multi-Factor Authentication) comme obligatoire dans votre pool d'utilisateurs, ce champ doit avoir la valeur `true` pour tous les utilisateurs. Si vous avez désactivé l'authentification MFA, ce champ doit avoir la valeur `false` pour tous les utilisateurs. Si vous avez défini l'authentification MFA comme facultative, ce champ peut être `true` ou `false`, mais ne peut pas être vide.
- La longueur maximale de la ligne est de 16 000 caractères.
- La taille maximale du fichier CSV est de 100 Mo.
- Le nombre maximal de lignes (utilisateurs) du fichier est de 500 000. La ligne d'en-tête n'est pas comprise dans ce nombre maximal.
- La valeur du champ `updated_at` doit être une heure Posix exprimée en secondes, par exemple : **1471453471**.
- Les espaces de début ou de fin d'une valeur d'attribut seront supprimés.

La liste suivante est un exemple de fichier d'importation CSV pour un groupe d'utilisateurs sans attributs personnalisés. Le schéma de votre groupe d'utilisateurs peut être différent par rapport à cet exemple. Dans ce cas, vous devez fournir des valeurs de test dans le modèle CSV que vous téléchargez à partir de votre groupe d'utilisateurs.

```
cognito:username,name,given_name,family_name,middle_name,nickname,preferred_username,profile,pi  
John,,John,Doe,,,,,,,,,johndoe@example.com,TRUE,,02/01/1985,,,+12345550100,TRUE,123 Any  
Street,,FALSE  
Jane,,Jane,Roe,,,,,,,,,janeroe@example.com,TRUE,,01/01/1985,,,+12345550199,TRUE,100 Main  
Street,,FALSE
```

Création et exécution de la tâche d'importation de groupe d'utilisateurs Amazon Cognito

Cette section explique comment créer et exécuter la tâche d'importation de groupe d'utilisateurs à l'aide de la console Amazon Cognito et de l'AWS Command Line Interface (AWS CLI).

Rubriques

- [Importation d'utilisateurs à partir d'un fichier CSV \(console\)](#)
- [Importation d'utilisateurs \(AWS CLI\)](#)

Importation d'utilisateurs à partir d'un fichier CSV (console)

La procédure suivante montre comment importer les utilisateurs à partir du fichier CSV.

Pour importer les utilisateurs à partir du fichier CSV (console)

1. Accédez à la [console Amazon Cognito](#). Il se peut que vous soyez invité à saisir vos informations d'identification AWS.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste.
4. Sélectionnez l'onglet Utilisateurs.
5. Dans la section Import users (Importer des utilisateurs), choisissez Create an import job (Créer une tâche d'importation).
6. Sur la page Create import job (Créer une tâche d'importation), saisissez un nom de tâche.
7. Choisissez Create a new IAM role (Créer un nouveau rôle IAM) ou Use an existing IAM role (Utiliser un rôle IAM existant).
 - a. Si vous avez choisi Create a new IAM role (Créer un nouveau rôle IAM), attribuez un nom à votre nouveau rôle. Amazon Cognito crée alors automatiquement un rôle doté des autorisations et de la relation d'approbation adéquates. Le principal IAM qui crée la tâche d'importation doit disposer des autorisations permettant de créer des rôles IAM.

- b. Si vous avez choisi Use an existing IAM role (Utiliser un rôle IAM existant), choisissez un rôle dans la liste située en dessous de IAM role selection (Sélection du rôle IAM). Ce rôle doit disposer des autorisations et de la politique d'approbation décrites dans [Création du rôle IAM CloudWatch Logs](#).
8. Choisissez Create job (Créer une tâche) pour soumettre votre tâche, mais démarrez-la plus tard. Choisissez Create and start job (Créer et démarrer une tâche) pour soumettre votre tâche et la démarrer immédiatement.
9. Si vous avez créé votre tâche mais que vous ne l'avez pas démarrée, vous pouvez la démarrer ultérieurement. Dans l'onglet Users (Utilisateurs), sous Import users (Importer des utilisateurs), choisissez votre tâche d'importation, puis sélectionnez Start (Démarrer). Vous pouvez également soumettre une demande d'API [StartUserImportJob](#) à partir d'un kit SDK AWS.
10. Surveillez la progression de votre tâche d'importation d'utilisateurs dans l'onglet Users (Utilisateurs) sous Import users (Importer des utilisateurs). Si votre tâche échoue, vous pouvez sélectionner la valeur Status (État). Pour plus de détails, sélectionnez View the CloudWatch logs for more details (Consulter les journaux CloudWatch pour plus de détails) et passez en revue les problèmes éventuels dans la console CloudWatch Logs.

Importation d'utilisateurs (AWS CLI)

Les commandes de CLI suivantes sont disponibles pour l'importation des utilisateurs dans un pool d'utilisateurs :

- `create-user-import-job`
- `get-csv-header`
- `describe-user-import-job`
- `list-user-import-jobs`
- `start-user-import-job`
- `stop-user-import-job`

Pour obtenir la liste des options de ligne de commande pour ces commandes, utilisez l'option de ligne de commande `help`. Par exemple :

```
aws cognito-idp get-csv-header help
```

Création d'une tâche d'importation d'utilisateurs

Après avoir créé votre fichier CSV, créez une tâche d'importation d'utilisateurs en exécutant la commande d'interface de ligne de commande suivante, où *JOB_NAME* est le nom que vous choisissez pour la tâche, *USER_POOL_ID* est l'ID du groupe d'utilisateurs dans lequel les nouveaux utilisateurs seront ajoutés et *ROLE_ARN* est l'ARN de rôle que vous avez reçu dans [Création du rôle IAM CloudWatch Logs](#) :

```
aws cognito-idp create-user-import-job --job-name "JOB_NAME" --user-pool-id "USER_POOL_ID" --cloud-watch-logs-role-arn "ROLE_ARN"
```

La valeur de *PRE_SIGNED_URL* retournée dans la réponse est valide pendant 15 minutes. Au-delà de ce délai, la valeur expire et vous devez créer une nouvelle tâche d'importation des utilisateurs pour obtenir une nouvelle URL.

Exemple Exemple de réponse :

```
{
  "UserImportJob": {
    "Status": "Created",
    "SkippedUsers": 0,
    "UserPoolId": "USER_POOL_ID",
    "ImportedUsers": 0,
    "JobName": "JOB_NAME",
    "JobId": "JOB_ID",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CloudWatchLogsRoleArn": "ROLE_ARN",
    "FailedUsers": 0,
    "CreationDate": 1470957431.965
  }
}
```

Valeurs d'état pour une tâche d'importation d'utilisateurs

Dans les réponses à vos commandes d'importation d'utilisateurs, vous verrez l'une des valeurs Status suivantes :

- **Created** – La tâche a été créée, mais n'a pas démarré.
- **Pending** – Un état de transition. Vous avez démarré la tâche, mais elle n'a pas encore commencé à importer les utilisateurs.

- **InProgress** – La tâche a démarré et les utilisateurs sont en cours d'importation.
- **Stopping** – Vous avez arrêté la tâche, mais la tâche n'a pas encore cessé d'importer les utilisateurs.
- **Stopped** – Vous avez arrêté la tâche et la tâche a cessé d'importer les utilisateurs.
- **Succeeded** – La tâche a été bien exécutée.
- **Failed** – La tâche s'est arrêtée à la suite d'une erreur.
- **Expired** – Vous avez créé une tâche, mais n'avez pas commencé la tâche au cours des 24 ou 48 heures écoulées. Toutes les données associées à la tâche ont été supprimées et la tâche ne peut pas être démarrée.

Chargement du fichier CSV

Utilisez la commande `curl` suivante pour charger le fichier CSV contenant vos données utilisateur sur l'URL présignée que vous avez obtenue de la réponse de la commande `create-user-import-job`.

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"  
"PRE_SIGNED_URL"
```

Dans la sortie de cette commande, recherchez l'expression "We are completely uploaded and fine". Cette expression indique que le fichier a été téléchargé avec succès.

Description d'une tâche d'importation d'utilisateurs

Pour obtenir une description de la tâche d'importation d'utilisateurs, utilisez la commande suivante, où *USER_POOL_ID* est l'ID du groupe d'utilisateurs et *JOB_ID* l'ID de la tâche renvoyée lorsque vous avez créé la tâche d'importation d'utilisateurs.

```
aws cognito-idp describe-user-import-job --user-pool-id "USER_POOL_ID" --job-id  
"JOB_ID"
```

Exemple Exemple de réponse :

```
{  
  "UserImportJob": {  
    "Status": "Created",  
    "SkippedUsers": 0,  
    "UserPoolId": "USER_POOL_ID",
```

```
"ImportedUsers": 0,
"JobName": "JOB_NAME",
"JobId": "JOB_ID",
"PreSignedUrl": "PRE_SIGNED_URL",
"CloudWatchLogsRoleArn": "ROLE_ARN",
"FailedUsers": 0,
"CreationDate": 1470957431.965
}
}
```

Dans l'exemple de sortie précédent, la valeur de *PRE_SIGNED_URL* est l'URL sur laquelle vous avez chargé le fichier CSV. La valeur de *ROLE_ARN* correspond à l'ARN du rôle CloudWatch Logs que vous avez reçu lorsque vous avez créé le rôle.

Affichage des tâches d'importation d'utilisateurs

Pour afficher les tâches d'importation d'utilisateurs, utilisez la commande suivante :

```
aws cognito-idp list-user-import-jobs --user-pool-id "USER_POOL_ID" --max-results 2
```

Exemple Exemple de réponse :

```
{
  "UserImportJobs": [
    {
      "Status": "Created",
      "SkippedUsers": 0,
      "UserPoolId": "USER_POOL_ID",
      "ImportedUsers": 0,
      "JobName": "JOB_NAME",
      "JobId": "JOB_ID",
      "PreSignedUrl": "PRE_SIGNED_URL",
      "CloudWatchLogsRoleArn": "ROLE_ARN",
      "FailedUsers": 0,
      "CreationDate": 1470957431.965
    },
    {
      "CompletionDate": 1470954227.701,
      "StartDate": 1470954226.086,
      "Status": "Failed",
      "UserPoolId": "USER_POOL_ID",
      "ImportedUsers": 0,

```

```

        "SkippedUsers": 0,
        "JobName": "JOB_NAME",
        "CompletionMessage": "Too many users have failed or been skipped during the
import.",
        "JobId": "JOB_ID",
        "PreSignedUrl": "PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn": "ROLE_ARN",
        "FailedUsers": 5,
        "CreationDate": 1470953929.313
    }
],
    "PaginationToken": "PAGINATION_TOKEN"
}

```

Les tâches sont affichées par ordre chronologique, depuis la dernière créée jusqu'à la première créée. La chaîne `PAGINATION_TOKEN` après la deuxième tâche indique qu'il y a des résultats supplémentaires pour cette commande de liste. Pour afficher les résultats supplémentaires, utilisez l'option `--pagination-token` comme suit :

```
aws cognito-idp list-user-import-jobs --user-pool-id "USER_POOL_ID" --max-results 10 --
pagination-token "PAGINATION_TOKEN"
```

Démarrage d'une tâche d'importation d'utilisateurs

Pour démarrer une tâche d'importation d'utilisateurs, utilisez la commande suivante :

```
aws cognito-idp start-user-import-job --user-pool-id "USER_POOL_ID" --job-id "JOB_ID"
```

Une seule tâche d'importation peut être active à la fois par compte.

Exemple Exemple de réponse :

```

{
  "UserImportJob": {
    "Status": "Pending",
    "StartDate": 1470957851.483,
    "UserPoolId": "USER_POOL_ID",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "JobName": "JOB_NAME",
    "JobId": "JOB_ID",

```

```
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CloudWatchLogsRoleArn": "ROLE_ARN",
    "FailedUsers": 0,
    "CreationDate": 1470957431.965
  }
}
```

Arrêt d'une tâche d'importation d'utilisateurs

Pour arrêter une tâche d'importation d'utilisateur alors qu'elle est en cours, utilisez la commande suivante. Une fois que vous avez arrêté la tâche, elle ne peut pas être redémarrée.

```
aws cognito-idp stop-user-import-job --user-pool-id "USER_POOL_ID" --job-id "JOB_ID"
```

Exemple Exemple de réponse :

```
{
  "UserImportJob": {
    "CompletionDate": 1470958050.571,
    "StartDate": 1470958047.797,
    "Status": "Stopped",
    "UserPoolId": "USER_POOL_ID",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "JobName": "JOB_NAME",
    "CompletionMessage": "The Import Job was stopped by the developer.",
    "JobId": "JOB_ID",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CloudWatchLogsRoleArn": "ROLE_ARN",
    "FailedUsers": 0,
    "CreationDate": 1470957972.387
  }
}
```

Affichage des résultats de l'importation du groupe d'utilisateurs dans la console CloudWatch

Vous pouvez afficher les résultats de votre tâche d'importation dans la console Amazon CloudWatch.

Rubriques

- [Affichage des résultats](#)
- [Interprétation des résultats](#)

Affichage des résultats

Les étapes suivantes expliquent comment afficher les résultats de l'importation du pool d'utilisateurs.

Pour afficher les résultats de l'importation du pool d'utilisateurs

1. Connectez-vous à la AWS Management Console et ouvrez la console CloudWatch à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Choisissez Journaux.
3. Sélectionnez le pool de journaux pour vos tâches d'importation du pool d'utilisateurs. Le nom du groupe de journaux est sous la forme `/aws/cognito/userpools/USER_POOL_ID/USER_POOL_NAME`.
4. Choisissez le journal pour la tâche d'importation d'utilisateurs que vous venez juste d'exécuter. Le nom du journal est sous la forme `JOB_ID/JOB_NAME`. Les résultats du journal se réfèrent à vos utilisateurs par numéro de ligne. Aucune donnée utilisateur n'est écrite dans le journal. Pour chaque utilisateur, une ligne similaire à la suivante s'affiche :
 - [SUCCEEDED] Line Number 5956 - The import succeeded.
 - [SKIPPED] Line Number 5956 - The user already exists.
 - [FAILED] Line Number 5956 - The User Record does not set any of the auto verified attributes to true. (Example: email_verified to true).

Interprétation des résultats

Les utilisateurs importés avec succès ont leur état défini sur « PasswordReset ».

Dans les cas suivants, l'utilisateur n'est pas importé, mais la tâche d'importation se poursuit :

- Aucun attribut à vérification automatique n'est défini pour `true`.
- Les données utilisateur ne correspondent pas au schéma.
- L'utilisateur n'a pas pu être importé en raison d'une erreur interne.

Dans les cas suivants, la tâche d'importation échoue :

- Le rôle Amazon CloudWatch Logs ne peut pas être endossé, ne possède pas la stratégie d'accès appropriée ou a été supprimé.
- Le pool d'utilisateurs a été supprimé.

- Amazon Cognito ne peut pas analyser le fichier .csv.

Obligation pour les utilisateurs importés de réinitialiser leur mot de passe

Chaque utilisateur importé doit saisir un nouveau mot de passe lors de sa première connexion. La procédure suivante décrit l'expérience utilisateur dans une application personnalisée avec des utilisateurs locaux après l'importation d'un fichier CSV. Si vos utilisateurs se connectent à l'aide de l'interface utilisateur hébergée, Amazon Cognito les invite à définir un nouveau mot de passe lors de leur première connexion.

Obligation pour les utilisateurs importés de réinitialiser leur mot de passe

1. Dans votre application, essayez de vous connecter silencieusement pour l'utilisateur actuel avec `InitiateAuth` à l'aide d'un mot de passe aléatoire.
2. Amazon Cognito renvoie `NotAuthorizedException` quand `PreventUserExistenceErrors` est activé. Sinon, la valeur renvoyée est `PasswordResetRequiredException`.
3. Votre application effectue une demande d'API `ForgotPassword` et réinitialise le mot de passe de l'utilisateur.
 - a. L'application soumet le nom d'utilisateur dans une demande d'API `ForgotPassword`.
 - b. Amazon Cognito envoie un code à l'adresse e-mail ou au numéro de téléphone vérifiés. La destination dépend des valeurs que vous avez fournies pour `email_verified` et `phone_number_verified` dans votre fichier CSV. La réponse à la demande `ForgotPassword` indique la destination du code.

Note

Votre groupe d'utilisateurs doit être configuré pour vérifier les adresses e-mail ou les numéros de téléphone. Pour de plus amples informations, veuillez consulter [Inscription et confirmation des comptes d'utilisateur](#).

- c. Votre application affiche un message demandant à votre utilisateur de vérifier l'endroit où le code a été envoyé et l'invite à saisir le code et un nouveau mot de passe.
- d. L'utilisateur entre le code et le nouveau mot de passe dans l'application.
- e. L'application soumet le code et le nouveau mot de passe dans une demande d'API `ConfirmForgotPassword`.

- f. Votre application redirige votre utilisateur vers la connexion.

Attributs de groupe d'utilisateurs

Les attributs sont des éléments d'information, comme le nom, l'adresse e-mail et le numéro de téléphone, qui vous aident à identifier des utilisateurs individuels. Un nouveau groupe d'utilisateurs possède un ensemble d'attributs standard par défaut. Vous pouvez également ajouter des attributs personnalisés à la définition de votre groupe d'utilisateurs dans le AWS Management Console. Cette rubrique décrit ces attributs en détail et vous donne des conseils sur la configuration du groupe d'utilisateurs.

Ne stockez pas toutes les informations sur vos utilisateurs dans des attributs. Par exemple, conservez les données utilisateur qui changent fréquemment, telles que les statistiques d'utilisation ou les scores de jeu, dans un magasin de données distinct, comme Amazon Cognito Sync ou Amazon DynamoDB.

Note

Certains documents et normes font référence aux attributs en tant que membres.

Rubriques

- [Attributs standard](#)
- [Noms d'utilisateur et noms d'utilisateurs préférés](#)
- [Personnalisation des attributs de connexion](#)
- [Attributs personnalisés](#)
- [Autorisations d'attributs et de portées](#)

Attributs standard

Amazon Cognito attribue à tous les utilisateurs un ensemble d'attributs standard en fonction de la [Spécification OpenID Connect](#). Par défaut, les valeurs des attributs standard et personnalisés peuvent être des chaînes d'une longueur maximale de 2 048 caractères, mais certaines d'entre elles présentent des restrictions de format.

Les attributs standard sont les suivants :

- address
- birthdate
- email
- family_name
- gender
- given_name
- locale
- middle_name
- name
- nickname
- phone_number
- picture
- preferred_username
- profile
- sub
- updated_at
- website
- zoneinfo

À l'exception de sub, les attributs standard sont facultatifs par défaut pour tous les utilisateurs. Pour rendre un attribut obligatoire, pendant le processus de création du groupe d'utilisateurs, cochez la case Obligatoire en regard de l'attribut. Amazon Cognito attribue une valeur d'identifiant utilisateur unique à l'attribut sub de chaque utilisateur. Seuls les attributs e-mail et phone_number peuvent être vérifiés.

Note

Quand vous marquez un attribut standard comme étant Required (Obligatoire), un utilisateur ne peut pas s'enregistrer s'il ne fournit pas de valeur pour cet attribut. Pour créer des utilisateurs et ne pas attribuer de valeurs aux attributs requis, les administrateurs peuvent utiliser l'[AdminCreateUserAPI](#). Après avoir créé un groupe d'utilisateurs, vous ne pouvez pas changer un attribut obligatoire en attribut non obligatoire, et inversement.

Détails des attributs standard et restrictions de format

birthdate

La valeur doit être une date valide de 10 caractères au format AAAA-MM-JJ.

e-mail

Les utilisateurs et les administrateurs peuvent vérifier les valeurs des adresses e-mail.

Un administrateur disposant d'un Compte AWS des autorisations appropriées peut modifier l'adresse e-mail de l'utilisateur et la marquer comme vérifiée. Marquez une adresse e-mail comme vérifiée à l'aide de l'[AdminUpdateUserAttributes](#) API ou de la commande [admin-update-user-attributes](#) AWS Command Line Interface (AWS CLI). Avec cette commande, l'administrateur peut affecter à l'attribut `email_verified` la valeur `true`. Vous pouvez également modifier un utilisateur dans l'onglet Utilisateurs du AWS Management Console pour marquer une adresse e-mail comme vérifiée.

La valeur doit être une chaîne d'adresse e-mail valide conforme au format d'e-mail standard avec le symbole `@` et le domaine, d'une longueur maximale de 2 048 caractères.

phone_number

Un utilisateur doit fournir un numéro de téléphone si l'authentification multifacteur (MFA) par SMS est active. Pour plus d'informations, consultez [Ajout de l'authentification MFA à un groupe d'utilisateurs](#).

Les utilisateurs et les administrateurs peuvent vérifier les numéros de téléphone.

Un administrateur disposant des Compte AWS autorisations appropriées peut modifier le numéro de téléphone de l'utilisateur et le marquer comme vérifié. Marquez un numéro de téléphone comme vérifié à l'aide de l'[AdminUpdateUserAttributes](#) API ou de la [admin-update-user-attributes](#) AWS CLI commande. Avec cette commande, l'administrateur peut affecter à l'attribut `phone_number_verified` la valeur `true`. Vous pouvez également modifier un utilisateur dans l'onglet Utilisateurs du AWS Management Console pour marquer un numéro de téléphone comme vérifié.

Important

Les numéros de téléphone doivent suivre ces règles de formatage : un numéro de téléphone doit commencer par le signe plus (+), suivi immédiatement de l'indicatif du

pays. Ils peuvent uniquement contenir le signe + et des chiffres. Supprimez tous les autres caractères d'un numéro de téléphone, tels que des parenthèses, des espaces ou des tirets (-), avant de soumettre la valeur au service. Par exemple, un numéro de téléphone basé aux États-Unis doit respecter ce format : **+14325551212**.

preferred_username

Vous pouvez sélectionner preferred_username au besoin ou en tant qu'alias, mais pas les deux. S'il s'agit d'un alias, vous pouvez envoyer une demande à l'opération [UpdateUserAttributes](#) API et ajouter la valeur de l'attribut après avoir confirmé l'utilisateur.

sub

Indexez et recherchez vos utilisateurs en fonction de l'attribut sub. L'attribut sub est un identifiant utilisateur unique au sein de chaque groupe d'utilisateurs. Les utilisateurs peuvent modifier des attributs tels que phone_number et email. L'attribut sub a une valeur fixe. Pour plus d'informations sur comment trouver des utilisateurs, consultez [Gestion et recherche de comptes d'utilisateur](#).

Afficher les attributs requis

Procédez comme suit pour afficher les attributs obligatoires d'un groupe d'utilisateurs.

Note

Vous ne pouvez pas modifier les attributs obligatoires après avoir créé un groupe d'utilisateurs.

Pour afficher les attributs requis

1. Accédez à [Amazon Cognito](#) dans le AWS Management Console. Si la console vous y invite, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste.
4. Choisissez l'onglet Sign-up experience (Expérience d'inscription).

5. Dans la section Required attributes (Attributs obligatoires), consultez les attributs obligatoires de votre groupe d'utilisateurs.

Noms d'utilisateur et noms d'utilisateurs préférés

La valeur `username` est un attribut distinct, à ne pas confondre avec l'attribut `name`. Chaque utilisateur a un attribut `username`. Amazon Cognito génère automatiquement un nom d'utilisateur pour les utilisateurs fédérés. Vous devez fournir un attribut `username` pour créer un utilisateur local dans l'annuaire Amazon Cognito. Après avoir créé un utilisateur, vous ne pouvez pas modifier la valeur de l'attribut `username`.

Les développeurs peuvent utiliser l'attribut `preferred_username` pour donner aux utilisateurs des noms d'utilisateur qu'ils peuvent modifier. Pour plus d'informations, consultez [Personnalisation des attributs de connexion](#).

Si votre application ne nécessite pas de nom d'utilisateur, vous n'avez pas besoin de demander aux utilisateurs d'en fournir un. Votre application peut créer un nom d'utilisateur unique pour les utilisateurs en arrière-plan. Cela peut s'avérer utile si vous souhaitez que les utilisateurs s'enregistrent et se connectent avec une adresse e-mail et un mot de passe. Pour plus d'informations, consultez [Personnalisation des attributs de connexion](#).

L'attribut `username` doit être unique au sein d'un groupe d'utilisateurs. Un `username` peut être réutilisé, mais seulement après avoir été supprimé et que plus personne ne l'utilise. Pour plus d'informations sur les contraintes de chaîne appliquées aux `username` attributs, consultez la propriété `username` d'une demande d'[SignUpAPI](#).

Personnalisation des attributs de connexion

Lorsque vous créez un groupe d'utilisateurs, vous pouvez configurer des attributs de nom d'utilisateur si vous souhaitez que vos utilisateurs s'inscrivent et se connectent avec une adresse e-mail ou un numéro de téléphone en guise de nom d'utilisateur. Vous pouvez également configurer des attributs d'alias pour permettre à vos utilisateurs d'inclure plusieurs attributs au moment de s'inscrire pour se connecter ensuite avec un nom d'utilisateur, un nom d'utilisateur préféré, une adresse e-mail ou un numéro de téléphone.

Important

Après avoir créé un groupe d'utilisateurs, vous ne pouvez pas modifier ce paramètre.

Comment choisir entre attributs d'alias et attributs de nom d'utilisateur

Votre exigence	Attributs d'alias	Attributs de nom d'utilisateur
Les utilisateurs disposent de plusieurs attributs de connexion	Oui ¹	Non- ²
Les utilisateurs doivent vérifier leur adresse e-mail ou leur numéro de téléphone avant de pouvoir se connecter	Oui	Non
Inscrivez les utilisateurs avec des adresses e-mail ou des numéros de téléphone dupliqués et évitez les <code>UsernameExistsException</code> erreurs ³	Oui	Non
Possibilité d'attribuer la même valeur d'attribut d'adresse e-mail ou de numéro de téléphone à plusieurs utilisateurs	Oui ⁴	Non

¹ Les attributs de connexion disponibles sont le nom d'utilisateur, l'adresse e-mail, le numéro de téléphone et le nom d'utilisateur préféré.

² Possibilité de se connecter avec une adresse e-mail ou un numéro de téléphone.

³ Votre groupe d'utilisateurs ne génère pas des erreurs `UsernameExistsException` quand les utilisateurs s'enregistrent avec des adresses e-mail ou des numéros de téléphone potentiellement dupliqués, mais sans nom d'utilisateur. Ce comportement est indépendant de l'option Empêcher les erreurs d'existence de nom d'utilisateur, qui s'applique aux opérations de connexion, mais pas aux opérations d'inscription.

⁴ Seul le dernier utilisateur ayant vérifié l'attribut peut l'utiliser pour se connecter.

Option 1 : plusieurs attributs de connexion (attributs d'alias)

Vous pouvez activer les alias si vous souhaitez permettre à vos utilisateurs de choisir de saisir leur nom d'utilisateur ou d'autres valeurs d'attributs lorsqu'ils se connectent. Par défaut, les utilisateurs se connectent avec leur nom d'utilisateur et un mot de passe. Le nom d'utilisateur est une valeur fixe que les utilisateurs ne peuvent pas modifier. Si vous marquez un attribut en tant qu'alias, les utilisateurs peuvent l'utiliser pour se connecter à la place de leur nom d'utilisateur. Les attributs que vous pouvez marquer en tant qu'alias sont l'adresse e-mail, le numéro de téléphone et le nom d'utilisateur préféré. Par exemple, si vous sélectionnez l'adresse e-mail et le numéro de téléphone en tant qu'alias pour un groupe d'utilisateurs, les utilisateurs de ce groupe peuvent se connecter en indiquant leur nom d'utilisateur, leur adresse e-mail ou leur numéro de téléphone, suivi de leur mot de passe.

Pour choisir les attributs d'alias, sélectionnez User name (Nom d'utilisateur) et au moins une option de connexion supplémentaire lorsque vous créez votre groupe d'utilisateurs.

Note

Quand vous configurez votre groupe d'utilisateurs comme non sensible à la casse, un utilisateur peut utiliser des lettres minuscules ou majuscules pour s'inscrire ou se connecter avec son alias. Pour plus d'informations, consultez le manuel [CreateUserPool](#) de référence de l'API des groupes d'utilisateurs Amazon Cognito.

Si vous sélectionnez une adresse e-mail comme alias, Amazon Cognito n'accepte pas de nom d'utilisateur correspondant à un format d'adresse e-mail valide. De même, si vous sélectionnez un numéro de téléphone comme alias, Amazon Cognito n'accepte pas un nom d'utilisateur pour ce groupe d'utilisateurs qui correspond à un format de numéro de téléphone valide.

Note

Les valeurs d'alias doivent être uniques dans l'ensemble du groupe d'utilisateurs. Si vous configurez un alias pour un numéro de téléphone ou une adresse e-mail, la valeur que vous fournissez peut avoir un état vérifié dans un seul compte. Lors de l'inscription, si votre utilisateur fournit une adresse e-mail ou un numéro de téléphone comme valeur d'alias et qu'un autre utilisateur a déjà utilisé cette valeur d'alias, l'enregistrement aboutit. Toutefois, quand un utilisateur tente de confirmer le compte avec cette adresse e-mail (ou ce numéro de téléphone) et saisit le code valide, Amazon Cognito renvoie une erreur

`AliasExistsException`. Cette erreur indique à l'utilisateur qu'un compte avec cette adresse e-mail (ou ce numéro de téléphone) existe déjà. À ce stade, l'utilisateur peut abandonner sa tentative de création du compte et essayer plutôt de réinitialiser le mot de passe de l'ancien compte. Si l'utilisateur poursuit la création du nouveau compte, votre application doit appeler l'API `ConfirmSignUp` avec l'option `forceAliasCreation`. `ConfirmSignUp` avec `forceAliasCreation` déplace l'alias du compte précédent vers le compte nouvellement créé et marque l'attribut comme étant non vérifié dans le compte précédent.

Les numéros de téléphone et les adresses e-mail deviennent des alias actifs pour un utilisateur une fois seulement que l'utilisateur les a vérifiés. Nous vous recommandons d'opter pour la vérification automatique des adresses e-mail et des numéros de téléphone si vous les utilisez comme alias.

Optez pour les attributs d'alias pour éviter des erreurs `UsernameExistsException` en rapport avec les attributs d'adresse e-mail et de numéro de téléphone lors de l'inscription de vos utilisateurs.

Activez l'attribut `preferred_username` afin que votre utilisateur puisse modifier le nom d'utilisateur qu'il utilise pour se connecter sans que sa valeur d'attribut `username` change. Si vous souhaitez configurer cette expérience utilisateur, envoyez la nouvelle valeur `username` en tant que `preferred_username` et choisissez `preferred_username` comme alias. Les utilisateurs peuvent alors se connecter avec la nouvelle valeur qu'ils ont saisie. Si vous sélectionnez `preferred_username` comme alias, votre utilisateur peut fournir cette valeur uniquement quand il confirme un compte. Il ne peut pas fournir cette valeur au moment de l'inscription.

Vous pouvez décider ou non de permettre à l'utilisateur de se connecter avec un ou plusieurs des alias suivants lorsqu'il s'inscrit avec un nom d'utilisateur.

- Adresse e-mail vérifiée
- Numéro de téléphone vérifié
- Nom d'utilisateur préféré

Une fois l'utilisateur inscrit, il peut modifier ces alias.

Important

Lorsque votre groupe d'utilisateurs prend en charge la connexion à l'aide d'alias et que vous souhaitez autoriser ou rechercher un utilisateur, ne l'identifiez pas à l'aide de ses attributs de

connexion. L'identifiant utilisateur à valeur fixe sub est le seul indicateur cohérent de l'identité de votre utilisateur.

Incluez les étapes suivantes quand vous créez le groupe d'utilisateurs, afin que les utilisateurs puissent se connecter avec un alias.

Pour configurer un groupe d'utilisateurs afin que les utilisateurs puissent se connecter avec un nom d'utilisateur préféré

1. Accédez à [Amazon Cognito](#) dans AWS Management Console. Si la console vous y invite, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Dans l'angle supérieur droit de la page, choisissez Créer un groupe d'utilisateurs pour lancer l'assistant de création de groupe d'utilisateurs.
4. Dans Configure sign-in experience (Configurer l'expérience de connexion), choisissez l'identité Provider types (Types de fournisseurs) que vous souhaitez associer à votre groupe d'utilisateurs.
5. Sous Options de connexion du groupe d'utilisateurs Cognito, choisissez n'importe quelle combinaison de nom d'utilisateur, d'adresse de messagerie et de numéro de téléphone.
6. Sous Exigences relatives aux noms d'utilisateur, choisissez Autoriser les utilisateurs à se connecter avec un nom d'utilisateur préféré afin que vos utilisateurs puissent définir un autre nom d'utilisateur lorsqu'ils se connectent.
7. Choisissez Suivant, puis suivez toutes les étapes de l'assistant.

Option 2 : adresse e-mail ou numéro de téléphone en tant qu'attribut de connexion (attributs de nom d'utilisateur)

Quand l'utilisateur s'inscrit avec une adresse e-mail ou un numéro de téléphone en guise de nom d'utilisateur, vous pouvez choisir s'il peut s'inscrire uniquement avec une adresse e-mail, uniquement avec un numéro de téléphone, ou avec l'un des deux au choix.

Pour choisir des attributs de nom d'utilisateur, évitez de sélectionner Nom d'utilisateur comme option de connexion lorsque vous créez votre groupe d'utilisateurs.

L'adresse e-mail et le numéro de téléphone doivent être uniques et ne doivent pas être déjà utilisés par un autre utilisateur. Ils n'ont pas besoin d'être vérifiés. Une fois que l'utilisateur s'est inscrit avec une adresse e-mail ou un numéro de téléphone, il ne peut pas créer de nouveau compte avec la

même adresse e-mail ou le même numéro de téléphone. L'utilisateur peut uniquement réutiliser le compte existant et réinitialiser son mot de passe, si nécessaire. Toutefois, l'utilisateur peut modifier l'adresse e-mail ou le numéro de téléphone pour une nouvelle adresse e-mail ou un nouveau numéro de téléphone. Si l'adresse e-mail ou le numéro de téléphone ne sont pas déjà en cours d'utilisation, ils deviennent le nouveau nom d'utilisateur.

Note

Si un utilisateur s'inscrit avec une adresse e-mail comme nom d'utilisateur, il peut modifier ce dernier et utiliser une autre adresse e-mail. En revanche, il ne peut pas remplacer l'adresse e-mail par un numéro de téléphone. S'il s'inscrit avec un numéro de téléphone, il peut modifier le nom d'utilisateur en spécifiant un autre numéro de téléphone. En revanche, il ne peut remplacer le numéro de téléphone par une adresse e-mail.

Procédez comme suit pendant la création du groupe d'utilisateur pour configurer l'inscription et la connexion avec une adresse e-mail ou un numéro de téléphone.

Pour configurer un groupe d'utilisateurs de manière à autoriser les inscriptions et les connexions avec une adresse e-mail ou un numéro de téléphone

1. Accédez à [Amazon Cognito](#) dans AWS Management Console. Si la console vous y invite, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Dans le coin supérieur droit de la page, choisissez Créer un groupe d'utilisateurs pour lancer l'assistant de création de groupe d'utilisateurs.
4. Sous Cognito user pool sign-in options (Options de connexion au groupe d'utilisateurs Cognito), choisissez une combinaison quelconque des paramètres Email (Adresse e-mail) et Phone number (Numéro de téléphone) pour représenter les attributs dont l'utilisateur peut se servir pour se connecter.
5. Choisissez Next (Suivant), puis effectuez toutes les étapes restantes de l'assistant.

Note

Vous n'avez pas besoin de marquer une adresse e-mail ou un numéro de téléphone comme des attributs obligatoires pour votre groupe d'utilisateurs.

Pour implémenter l'option 2 dans votre application

1. Appelez l'API `CreateUserPool` pour créer votre groupe d'utilisateurs. Définissez le paramètre `UserNameAttributes` sur `phone_number`, `email` ou `phone_number | email`.
2. Appelez l'API `SignUp` et indiquez une adresse e-mail ou un numéro de téléphone dans le paramètre `username`. Cette API exécute alors les opérations suivantes :
 - Si la chaîne `username` a un format d'adresse e-mail valide, le groupe d'utilisateurs renseigne automatiquement l'attribut `email` de l'utilisateur avec la valeur `username`.
 - Si le format de numéro de téléphone de la chaîne `username` est valide, le groupe d'utilisateurs renseigne automatiquement l'attribut `phone_number` de l'utilisateur avec la valeur `username`.
 - Si le format de la chaîne `username` ne correspond pas à une adresse e-mail ou à un numéro de téléphone, l'API `SignUp` renvoie une exception.
 - L'API `SignUp` génère un UUID persistant pour votre utilisateur, qu'elle emploie en interne comme attribut de nom d'utilisateur inaltérable. Cet UUID a la même valeur que la demande sub du jeton d'identité de l'utilisateur.
 - Si la chaîne `username` contient une adresse e-mail ou un numéro de téléphone qui est déjà en cours d'utilisation, l'API `SignUp` renvoie une exception.

Vous pouvez utiliser une adresse e-mail ou un numéro de téléphone comme alias à la place du nom d'utilisateur dans toutes les API, à l'exception de l'API `ListUsers`. Quand vous appelez `ListUsers`, vous pouvez effectuer une recherche par l'attribut `email` ou `phone_number`. Si vous effectuez une recherche par `username`, vous devez indiquer le nom d'utilisateur réel, et non un alias.

Attributs personnalisés

Vous pouvez ajouter jusqu'à 50 attributs personnalisés à votre groupe d'utilisateurs. Vous pouvez leur appliquer une longueur minimale et/ou maximale. Toutefois, la longueur maximale pour un attribut personnalisé ne peut pas dépasser 2 048 caractères.

Chaque attribut personnalisé possède les caractéristiques suivantes :

- Vous pouvez le définir comme une chaîne ou un nombre. Amazon Cognito écrit des valeurs d'attribut personnalisées dans le jeton d'identification uniquement sous forme de chaînes.
- Vous ne pouvez pas exiger que les utilisateurs fournissent une valeur pour cet attribut.
- Vous ne pouvez pas le supprimer ni le modifier après l'avoir ajouté dans le groupe d'utilisateurs.

- Le nombre maximal de caractères du nom de l'attribut se situe dans la limite acceptée par Amazon Cognito. Pour plus d'informations, consultez [Quotas dans Amazon Cognito](#).
- Il peut être mutable ou inaltérable. Vous pouvez écrire une valeur dans un attribut inaltérable seulement quand vous créez un utilisateur. Vous pouvez modifier la valeur d'un attribut mutable si votre client d'application dispose d'une autorisation d'écriture sur cet attribut. Pour plus d'informations, consultez [Autorisations d'attributs et de portées](#).

Note

Dans votre code et dans les paramètres des règles pour [Utilisation du contrôle d'accès basé sur les rôles](#), les attributs personnalisés doivent être différenciés des attributs standard par le préfixe `custom:`.

Vous pouvez également ajouter des attributs de développeur lorsque vous créez des groupes d'utilisateurs, dans la `SchemaAttributes` propriété de [CreateUserPool](#). Les attributs de développeur ont un préfixe `dev:`. Vous ne pouvez modifier les attributs de développeur d'un utilisateur qu'à l'aide AWS d'informations d'identification. Les attributs de développeur sont une fonctionnalité héritée qu'Amazon Cognito a remplacée par des autorisations de lecture/écriture pour les clients de l'application.

Utilisez la procédure suivante pour créer un attribut personnalisé.

Pour ajouter un attribut personnalisé à l'aide de la console

1. Accédez à [Amazon Cognito](#) dans le AWS Management Console. Si la console vous y invite, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs dans la liste.
4. Choisissez l'onglet Sign-up experience (Expérience d'inscription) et dans l'onglet Custom attributes (Attributs personnalisés), choisissez Add custom attributes (Ajouter des attributs personnalisés).
5. Dans la page Ajouter des attributs personnalisés, fournissez les informations suivantes sur le nouvel attribut :
 - Entrez un Nom.
 - Sélectionnez un Type de Chaîne ou Numéro.

- Saisissez une longueur de chaîne ou valeur numérique Minimale.
- Saisissez une longueur de chaîne ou valeur numérique Maximale.
- Sélectionnez Mutable si vous souhaitez autoriser les utilisateurs à modifier la valeur d'un attribut personnalisé après qu'ils ont défini la valeur initiale.

6. Sélectionnez Enregistrer les modifications.

Autorisations d'attributs et de portées

Pour chaque client d'application, vous pouvez définir des autorisations de lecture et d'écriture pour chaque attribut utilisateur. De cette façon, vous pouvez contrôler l'accès dont dispose une application pour lire et modifier chaque attribut que vous stockez pour vos utilisateurs. Par exemple, vous pouvez avoir un attribut personnalisé qui indique si un utilisateur est un client payant ou non. Vos applications peuvent éventuellement voir cet attribut mais pas le modifier directement. Au lieu de cela, vous devez mettre à jour cet attribut à l'aide d'un outil d'administration ou d'un processus en arrière-plan. Vous pouvez définir des autorisations pour les attributs utilisateur à partir de la console Amazon Cognito, de l'API Amazon Cognito ou de l'interface AWS CLI. Par défaut, aucun nouvel attribut personnalisé n'est disponible tant que vous ne définissez pas les autorisations de lecture et d'écriture correspondantes. Par défaut, lorsque vous créez un nouveau client d'application, vous accordez à votre application des autorisations de lecture et d'écriture pour tous les attributs standard et personnalisés. Pour limiter votre application à la seule quantité d'informations dont elle a besoin, attribuez des autorisations spécifiques aux attributs dans la configuration de votre client d'application.

Il est recommandé de spécifier les autorisations de lecture et d'écriture des attributs lorsque vous créez un client d'application. Accordez à votre client d'application l'accès à l'ensemble minimal d'attributs utilisateur dont vous avez besoin pour le fonctionnement de votre application.

Note

[DescribeUserPoolClient](#) renvoie uniquement des valeurs pour `ReadAttributes` et `WriteAttributes` lorsque vous configurez des autorisations client de l'application autres que celles par défaut.

Pour mettre à jour les autorisations d'attribut (AWS Management Console)

1. Accédez à [Amazon Cognito](#) dans le AWS Management Console Si la console vous y invite, entrez vos AWS informations d'identification.

2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs dans la liste.
4. Choisissez l'onglet App integration (Intégration d'applications) et dans l'onglet App clients (Clients d'application), choisissez un client d'application dans la liste.
5. Dans Autorisation de lecture et d'écriture des attributs, choisissez Modifier.
6. Dans la page Modifier les autorisations de lecture et d'écriture des attributs, configurez vos autorisations de lecture et d'écriture, puis choisissez Enregistrez les modifications.

Répétez ces étapes pour chaque client d'application qui utilise l'attribut personnalisé.

Pour chaque application, vous pouvez marquer les attributs comme accessibles en lecture ou en écriture. Cette règle s'applique aux attributs standard et personnalisés. Votre application peut récupérer la valeur des attributs que vous marquez comme accessibles en lecture, et peut définir ou modifier la valeur des attributs que vous marquez comme accessibles en écriture. Si votre application essaie de définir une valeur pour un attribut qu'elle n'est pas autorisée à écrire, Amazon Cognito renvoie `NotAuthorizedException`. [GetUser](#) les demandes incluent un jeton d'accès associé à une réclamation de client d'application ; Amazon Cognito renvoie uniquement des valeurs pour les attributs que votre client d'application peut lire. Le jeton d'identification de votre utilisateur provenant d'une application contient uniquement des demandes correspondant aux attributs accessibles en lecture. Tous les clients d'application peuvent écrire les attributs requis par le groupe d'utilisateurs. Vous pouvez définir la valeur d'un attribut dans une demande d'API de groupes d'utilisateurs Amazon Cognito seulement lorsque vous fournissez également une valeur pour tous les attributs requis qui n'ont pas encore de valeur.

Les attributs personnalisés ont des fonctionnalités distinctes pour les autorisations de lecture et d'écriture. Vous pouvez les créer en tant qu'attributs modifiables ou inaltérables pour le groupe d'utilisateurs, et vous pouvez les définir en tant qu'attributs de lecture ou d'écriture pour un client d'application.

Un attribut personnalisé inaltérable peut être mis à jour une seule fois, pendant de la création de l'utilisateur. Vous pouvez renseigner un attribut inaltérable à l'aide des méthodes suivantes.

- `SignUp` : un utilisateur s'inscrit avec un client d'application qui dispose d'un accès en écriture à un attribut personnalisé inaltérable. Il fournit une valeur pour cet attribut.
- Connexion avec un IdP tiers : un utilisateur se connecte à un client d'application qui dispose d'un accès en écriture à un attribut personnalisé inaltérable. La configuration de votre groupe

d'utilisateurs pour son IdP comporte une règle pour mapper une demande fournie à un attribut inaltérable.

- `AdminCreateUser` : vous fournissez une valeur pour un attribut inaltérable.

Pour obtenir des informations sur les étendues que vous pouvez attribuer à vos clients d'application, consultez [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#).

Vous pouvez modifier les autorisations et les règles d'attribut une fois que vous avez créé le groupe d'utilisateurs.

Ajout d'exigences de mot de passe pour un groupe d'utilisateurs

Les mots de passe forts et complexes constituent une bonne pratique de sécurité pour votre groupe d'utilisateurs. En particulier dans les applications ouvertes sur Internet, les mots de passe faibles peuvent exposer les informations d'identification de vos utilisateurs à des systèmes qui devinent les mots de passe et tentent d'accéder à vos données. Plus un mot de passe est complexe, plus il est difficile à deviner. Amazon Cognito propose des outils supplémentaires pour les administrateurs soucieux de la sécurité, tels que des [fonctionnalités de sécurité avancées](#) et des [ACL AWS WAF Web](#), mais votre politique en matière de mots de passe est un élément central de la sécurité de votre annuaire d'utilisateurs.

Les mots de passe des utilisateurs locaux des groupes d'utilisateurs Amazon Cognito n'expirent pas automatiquement. Il est recommandé de consigner l'heure, la date et les métadonnées des réinitialisations du mot de passe utilisateur dans un système externe. Avec un journal externe de l'ancienneté du mot de passe, votre application ou un déclencheur Lambda peut rechercher l'âge du mot de passe d'un utilisateur et nécessiter une réinitialisation après une période donnée.

Vous pouvez configurer votre groupe d'utilisateurs pour exiger une complexité minimale des mots de passe conforme à vos normes de sécurité. Les mots de passe complexes ont une longueur minimale de huit caractères. Ils incluent également un mélange de majuscules, de chiffres et de caractères spéciaux.

Pour configurer une politique de mot de passe pour un groupe d'utilisateurs

1. Créez un groupe d'utilisateurs et accédez à l'étape Configurer les exigences de sécurité, ou accédez à un groupe d'utilisateurs existant et accédez à l'onglet Expérience de connexion.
2. Accédez à la Stratégie de mot de passe.

3. Choisissez un Mode de stratégie de mot de passe. Valeurs par défaut de Cognito configure votre groupe d'utilisateurs avec les paramètres minimaux recommandés. Vous pouvez également choisir une politique de mot de passe personnalisée.
4. Configurez une Longueur minimum du mot de passe. Tous les utilisateurs doivent s'inscrire ou être créés avec un mot de passe dont la longueur est supérieure ou égale à cette valeur. Vous pouvez définir cette valeur minimale jusqu'à 99, mais vos utilisateurs peuvent définir des mots de passe d'une longueur maximale de 256 caractères.
5. Configurez les règles de complexité des mots de passe sous Exigences relatives au mot de passe. Choisissez les types de caractères (chiffres, caractères spéciaux, lettres majuscules et minuscules) que vous souhaitez exiger (au moins un) dans le mot de passe de chaque utilisateur.

Vous pouvez exiger au moins l'un des caractères suivants dans les mots de passe. Une fois qu'Amazon Cognito a vérifié que les mots de passe contiennent le minimum de caractères requis, les mots de passe de vos utilisateurs peuvent contenir des caractères supplémentaires de n'importe quel type, jusqu'à la longueur maximale du mot de passe.

- Lettres [latines de base](#) majuscules et minuscules
- Nombres
- Les caractères spéciaux suivants.

```
^ $ * . [ ] { } ( ) ? " ! @ # % & / \ , > < ' : ; | _ ~ ` = + -
```

- Caractères espace ni au début ni à la fin.
6. Définissez une valeur pour Les mots de passe temporaires définis par les administrateurs expirent dans. Une fois ce délai écoulé, un nouvel utilisateur que vous avez créé dans la console Amazon Cognito ou avec `AdminCreateUser` ne peut pas se connecter ni définir un nouveau mot de passe. Une fois qu'ils se sont connectés avec leur mot de passe temporaire, leur compte utilisateur n'expire jamais. Pour mettre à jour la durée du mot de passe dans l'API des groupes d'utilisateurs Amazon Cognito, définissez une valeur pour [TemporaryPasswordValidityDays](#) dans votre demande [CreateUserPool](#) ou dans votre demande d'[UpdateUserPoolAPI](#).
- Pour réinitialiser l'accès à un compte utilisateur expiré, effectuez l'une des actions suivantes.
 - Supprimez le profil utilisateur et créez-en un nouveau.
 - Définissez un nouveau mot de passe permanent dans une demande d'[AdminSetUserPasswordAPI](#).

- Générez un nouveau code de confirmation dans une demande d'[AdminResetUserPassword](#) API.

Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito

Certains événements qui se produisent dans l'application cliente de votre groupe d'utilisateurs peuvent conduire Amazon Cognito à envoyer des e-mails à vos utilisateurs. Par exemple, si vous configurez votre groupe d'utilisateurs de façon à exiger la vérification des e-mails, Amazon Cognito envoie un e-mail au moment où un utilisateur crée un compte dans votre application ou réinitialise son mot de passe. En fonction de l'action qui déclenche l'e-mail, celui-ci contient un code de vérification ou un mot de passe temporaire.

Pour gérer la remise d'e-mails, vous pouvez utiliser l'une des options suivantes :

- [Configuration d'e-mail par défaut](#) intégrée au service Amazon Cognito.
- [Votre configuration Amazon Simple Email Service \(Amazon SES\)](#).

Vous pouvez modifier votre option de livraison après avoir créé votre groupe d'utilisateurs.

Amazon Cognito envoie des e-mails à vos utilisateurs avec un code qu'ils peuvent saisir ou un lien URL qu'ils peuvent sélectionner. Le tableau suivant indique les événements qui peuvent générer un e-mail.

Options de message

Activité	Opération API	Options de livraison	Options de format	Personnalisable	Modèle de message
Mot de passe oublié	ForgotPassword	Courrier électronique, SMS	code	Non	N/A
Invitation	AdminCreateUser	Courrier électronique, SMS	code	Oui	Message d'invitation

Activité	Opération API	Options de livraison	Options de format	Personnalisable	Modèle de message
Auto-enregistrement	SignUp	Courrier électronique, SMS	code, lien	Oui	Message de vérification
Vérification de l'adresse e-mail ou du numéro de téléphone	UpdateUserAttributes	Courrier électronique, SMS	code	Oui	Message de vérification
Authentification multifactorielle (MFA)	AdminInitiateAuth , InitiateAuth	SMS	code	Oui ¹	Message du MFA

¹ Pour les SMS.

Amazon SES facture les e-mails. Pour de plus amples informations, consultez la [tarification Amazon SES](#).

Configuration du courrier électronique par défaut

Amazon Cognito peut utiliser sa configuration d'e-mail par défaut pour gérer les envois d'e-mails à votre place. Lorsque vous utilisez l'option par défaut, Amazon Cognito limite le nombre d'e-mails envoyés par jour pour votre groupe d'utilisateurs. Pour plus d'informations sur les limites de service, consultez [Quotas dans Amazon Cognito](#). Pour les environnements de production classiques, la limite d'e-mails par défaut est inférieure au volume de remise requis. Pour permettre un volume de remise plus élevé, vous pouvez utiliser la configuration de votre e-mail Amazon SES.

Lorsque vous utilisez la configuration par défaut, vous utilisez les ressources Amazon SES gérées par AWS pour envoyer des e-mails. Amazon SES ajoute des adresses e-mail qui renvoient un [message d'erreur définitif](#) à une [liste de suppression au niveau du compte](#) ou à une [liste de suppression globale](#). Si une adresse e-mail non distribuable le devient ultérieurement, vous ne pouvez pas contrôler sa suppression de la liste de suppression tant que votre groupe d'utilisateurs est configuré pour utiliser la configuration par défaut. Une adresse e-mail peut rester indéfiniment sur

la liste de suppression AWS gérée par `-managed`. Pour gérer les adresses e-mail non distribuables, utilisez votre configuration de messagerie Amazon SES avec une liste de suppression au niveau du compte, comme cela est décrit dans la section suivante.

Quand vous utilisez la configuration de messagerie par défaut, vous pouvez utiliser l'une des adresses e-mail suivantes comme adresse d'envoi :

- L'adresse e-mail par défaut, `no-reply@verificationemail.com`.
- Une adresse e-mail personnalisée. Avant de pouvoir utiliser votre propre adresse e-mail, vous devez la vérifier avec Amazon SES et accorder à Amazon Cognito l'autorisation de l'utiliser.

Configuration de l'e-mail Amazon SES

Votre application peut avoir besoin d'un volume de remise plus élevé que celui offert par l'option par défaut. Pour augmenter le volume de remise possible, utilisez vos ressources Amazon SES avec votre groupe d'utilisateurs pour envoyer des e-mails à vos utilisateurs. Vous pouvez également [surveiller votre activité d'envoi d'e-mails](#) quand vous envoyez des e-mails avec votre propre configuration Amazon SES.

Avant de pouvoir utiliser votre configuration Amazon SES, vous devez vérifier une ou plusieurs adresses e-mail ou un domaine, avec Amazon SES. Utilisez une adresse e-mail vérifiée ou une adresse issue d'un domaine vérifié en guise d'adresse e-mail d'envoi que vous affectez à votre groupe d'utilisateurs. Quand Amazon Cognito envoie un e-mail à un utilisateur, il appelle Amazon SES pour vous et utilise votre adresse e-mail.

Quand vous utilisez votre configuration Amazon SES, les conditions suivantes s'appliquent :

- Les limites de remise d'e-mails pour votre groupe d'utilisateurs sont les mêmes que celles qui s'appliquent à votre adresse e-mail vérifiée Amazon SES dans votre Compte AWS.
- Vous pouvez gérer vos messages vers des adresses e-mail non distribuables à l'aide d'une liste de suppression au niveau du compte dans Amazon SES qui remplace la [liste de suppression globale](#). Quand vous utilisez une liste de suppression au niveau du compte, les retours à l'expéditeur des e-mails affectent la réputation de votre compte en tant qu'expéditeur. Pour plus d'informations, consultez [Utilisation de la liste de suppression au niveau du compte Amazon SES](#) dans le Guide du développeur Amazon Simple Email Service.

Configuration des régions de l'e-mail Amazon SES

L' Région AWS endroit où vous créez un groupe d'utilisateurs répondra à l'une des trois exigences relatives à la configuration des e-mails avec Amazon SES. Vous pouvez envoyer des e-mails depuis Amazon SES dans la même région que votre groupe d'utilisateurs, dans plusieurs régions, dont la même région, ou dans une ou plusieurs régions distantes. Pour de meilleures performances, envoyez des e-mails avec une identité vérifiée par Amazon SES dans la même région que votre groupe d'utilisateurs lorsque vous en avez le choix.

Catégories d'exigences régionales pour les identités vérifiées par Amazon SES

Dans la région uniquement

Vos groupes d'utilisateurs peuvent envoyer des e-mails avec des identités vérifiées au même Région AWS titre que le groupe d'utilisateurs. Dans la configuration d'e-mail par défaut sans adresse FROM e-mail personnalisée, Amazon Cognito utilise une identité `no-reply@verificationemail.com` vérifiée dans la même région.

Rétrocompatible

Vos groupes d'utilisateurs peuvent envoyer des e-mails avec des identités vérifiées dans la même région Région AWS ou dans l'une des autres régions suivantes :

- USA Est (Virginie du Nord)
- USA Ouest (Oregon)
- Europe (Irlande)

Cette fonctionnalité assure la continuité des ressources du pool d'utilisateurs que vous avez peut-être créées pour répondre aux exigences d'Amazon Cognito lors du lancement du service. Les groupes d'utilisateurs de cette période ne pouvaient envoyer des e-mails avec des identités vérifiées que dans un nombre limité de Régions AWS. Dans la configuration d'e-mail par défaut sans adresse FROM e-mail personnalisée, Amazon Cognito utilise une identité `no-reply@verificationemail.com` vérifiée dans la même région.

Région alternative

Vos groupes d'utilisateurs peuvent envoyer des e-mails avec des identités vérifiées dans une autre région Région AWS située en dehors de la région du groupe d'utilisateurs. Cette configuration se produit lorsqu'Amazon SES n'est pas disponible dans une région où Amazon Cognito est disponible.

La politique d'autorisation d'envoi d'Amazon SES relative à votre identité vérifiée dans la région alternative doit faire confiance au principal du service Amazon Cognito de la région d'origine. Pour plus d'informations, consultez [Pour accorder l'autorisation d'utiliser la configuration de messagerie par défaut](#).

Dans certaines de ces régions, Amazon Cognito divise les e-mails entre deux régions alternatives pour la configuration d'e-mail par défaut de COGNITO_DEFAULT. Dans ces cas, pour utiliser une adresse FROM e-mail personnalisée, la politique d'autorisation d'envoi d'Amazon SES pour votre identité vérifiée dans chaque région alternative doit faire confiance au principal du service Amazon Cognito de la région d'origine. Pour plus d'informations, consultez [Pour accorder l'autorisation d'utiliser la configuration de messagerie par défaut](#). Avec la configuration de messagerie Amazon SES DEVELOPER dans ces régions, vous devez utiliser une identité vérifiée dans la première région répertoriée et la configurer de manière à faire confiance au principal du service Amazon Cognito dans la région du groupe d'utilisateurs. Par exemple, dans un groupe d'utilisateurs au Moyen-Orient (Émirats arabes unis), configurez une identité vérifiée en Europe (Francfort) pour qu'elle soit fiable `cognito-idp.me-central-1.amazonaws.com`. Dans la configuration e-mail par défaut sans adresse FROM e-mail personnalisée, Amazon Cognito utilise une identité `no-reply@verificationemail.com` vérifiée dans chaque région.

Note

Dans la combinaison de conditions suivante, vous devez spécifier le `SourceArn` paramètre de [EmailConfiguration](#) avec un caractère générique dans l'élément `Region`, au format `arn:{{Partition}}:ses:*:{{Account}}:identity/{{IdentityName}}`. Cela permet à votre groupe d'utilisateurs d'envoyer des e-mails avec des identités vérifiées identiques Compte AWS dans les deux versions Régions AWS.

- `EmailSendingAccount` Le vôtre est `COGNITO_DEFAULT`.
- Vous souhaitez utiliser une FROM adresse personnalisée.
- Votre groupe d'utilisateurs envoie des e-mails dans une autre région.
- Votre groupe d'utilisateurs possède une deuxième région ¹alternative spécifiée dans le tableau des régions prises en charge par Amazon SES ci-dessous.

Si vous créez un groupe d'utilisateurs par programmation, à l'aide d'un SDK AWS , de l'API ou de la CLI Amazon Cognito AWS CDK, votre groupe d'utilisateurs envoie des e-mails avec l'

AWS CloudFormation identité Amazon SES SourceArn que le paramètre de spécifie pour votre groupe d'utilisateurs. [EmailConfiguration](#) L'identité Amazon SES doit être prise en charge Région AWS. Si votre EmailSendingAccount est COGNITO_DEFAULT et que vous ne spécifiez pas de paramètre SourceArn, Amazon Cognito envoie des messages électroniques depuis `no-reply@verificationemail.com` en utilisant des ressources dans la région où vous avez créé votre groupe d'utilisateurs.

Le tableau suivant indique Régions AWS où vous pouvez utiliser les identités Amazon SES avec Amazon Cognito.

Région du groupe d'utilisateurs	Option de région	Régions prises en charge par Amazon SES
USA Est (Virginie du Nord)	Rétrocompatible	USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
USA Est (Ohio)	Rétrocompatible	USA Est (Ohio), USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
USA Ouest (Californie du Nord)	Dans la région uniquement	USA Ouest (Californie du Nord)
US West (Oregon)	Rétrocompatible	USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
Canada (Centre)	Rétrocompatible	Canada (Centre), USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
Asie-Pacifique (Tokyo)	Rétrocompatible	Asie-Pacifique (Tokyo), USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
Asie-Pacifique (Séoul)	Rétrocompatible	Asie-Pacifique (Séoul), USA Est (Virginie du Nord), USA

Région du groupe d'utilisateurs	Option de région	Régions prises en charge par Amazon SES
		Ouest (Oregon), Europe (Irlande)
Asie-Pacifique (Mumbai)	Rétrocompatible	Asie-Pacifique (Mumbai), USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
Asie-Pacifique (Hyderabad)	Région alternative	Asie-Pacifique (Mumbai), Asie-Pacifique (Singapour) ¹
Asie-Pacifique (Singapour)	Rétrocompatible	Asie-Pacifique (Singapour), USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
Asie-Pacifique (Sydney)	Rétrocompatible	Asie-Pacifique (Sydney), USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
Asie-Pacifique (Osaka)	Dans la région uniquement	Asie-Pacifique (Osaka)
Asie-Pacifique (Jakarta)	Dans la région uniquement	Asie-Pacifique (Jakarta)
Asie-Pacifique (Melbourne)	Région alternative	Asie-Pacifique (Sydney), Asie-Pacifique (Singapour) ¹
Europe (Irlande)	Rétrocompatible	USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
Europe (Londres)	Rétrocompatible	Europe (Londres), USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)

Région du groupe d'utilisateurs	Option de région	Régions prises en charge par Amazon SES
Europe (Paris)	Dans la région uniquement	Europe (Paris)
Europe (Francfort)	Rétrocompatible	Europe (Francfort), USA Est (Virginie du Nord), USA Ouest (Oregon), Europe (Irlande)
Europe (Zurich)	Région alternative	Europe (Francfort), Europe (Londres) ¹
Europe (Stockholm)	Dans la région uniquement	Europe (Stockholm)
Europe (Milan)	Dans la région uniquement	Europe (Milan)
Europe (Espagne)	Région alternative	Europe (Paris), Europe (Stockholm) ¹
Moyen-Orient (Bahreïn)	Dans la région uniquement	Moyen-Orient (Bahreïn)
Moyen-Orient (EAU)	Région alternative	Europe (Francfort), Europe (Londres) ¹
Amérique du Sud (São Paulo)	Dans la région uniquement	Amérique du Sud (São Paulo)
Israël (Tel Aviv)	Dans la région uniquement	Israël (Tel Aviv)
Afrique (Le Cap)	Dans la région uniquement	Afrique (Le Cap)

¹ Utilisé dans les groupes d'utilisateurs avec la configuration de messagerie par défaut. Amazon Cognito distribue des e-mails aux identités vérifiées ayant la même adresse e-mail dans chaque région. Pour utiliser une FROM adresse personnalisée, EmailConfiguration configurez-la avec un SourceArn paramètre au format `arn:ses:*:${Account}:identity/${IdentityName}`.

Configuration de la messagerie pour votre groupe d'utilisateurs

Effectuez les étapes suivantes pour configurer les paramètres de messagerie pour votre groupe d'utilisateurs. Selon les paramètres que vous utilisez, vous pouvez avoir besoin de permissions IAM dans Amazon SES, AWS Identity and Access Management (IAM) et Amazon Cognito.

Note

Vous ne pouvez pas partager les ressources que vous créez au cours de ces étapes entre différents Comptes AWS. Par exemple, vous ne pouvez pas configurer un groupe d'utilisateurs dans un compte, puis l'utiliser avec une adresse e-mail Amazon SES dans un autre compte. Si vous utilisez Amazon Cognito dans plusieurs comptes, répétez ces étapes pour chaque compte.

Étape 1 : Vérifiez votre adresse e-mail ou votre domaine avec Amazon SES

Avant de configurer votre groupe d'utilisateurs, vous devez vérifier un ou plusieurs domaines ou adresses e-mail avec Amazon SES si vous souhaitez effectuer l'une des opérations suivantes :

- Utiliser votre propre adresse e-mail comme adresse d'envoi
- Utiliser votre configuration Amazon SES pour gérer la remise d'e-mails

En vérifiant votre adresse e-mail ou votre domaine, vous confirmez que vous en êtes propriétaire, ce qui contribue à empêcher toute utilisation non autorisée.

Pour plus d'informations sur la vérification d'une adresse e-mail avec Amazon SES, consultez [Vérifier une adresse e-mail](#) dans le Manuel du développeur Amazon Simple Email Service. Pour plus d'informations sur la vérification d'un domaine avec Amazon SES, consultez [Vérification des domaines](#).

Étape 2 : Sortie de votre compte de l'environnement de test (sandbox) Amazon SES

Omettez cette étape si vous utilisez la configuration d'e-mail par défaut d'Amazon Cognito.

Lorsque vous utilisez Amazon SES pour la première fois dans une région Région AWS, cela vous place Compte AWS dans le sandbox Amazon SES de cette région. Amazon SES utilise l'environnement de test (sandbox) pour éviter toute fraude ou abus. Si vous utilisez votre

configuration Amazon SES pour gérer la remise d'e-mails, vous devez sortir votre Compte AWS de l'environnement de test (sandbox) pour permettre à Amazon Cognito d'envoyer des e-mails à vos utilisateurs.

Dans l'environnement de test (sandbox), Amazon SES impose des restrictions eu égard au nombre d'e-mails que vous pouvez envoyer et à la destination de ces messages. Vous ne pouvez envoyer des e-mails qu'aux adresses et domaines que vous avez vérifiés avec Amazon SES, ou les envoyer à des adresses de simulateur de boîte aux lettres Amazon SES. Tant que Compte AWS vous êtes dans le sandbox, n'utilisez pas votre configuration Amazon SES pour des applications en production. Dans ce cas, Amazon Cognito ne peut pas envoyer de messages aux adresses e-mail de vos utilisateurs.

Pour vous Compte AWS retirer du sandbox, consultez [Moving out the Amazon SES sandbox](#) dans le manuel Amazon Simple Email Service Developer Guide.

Étape 3 : Octroi d'autorisations de remise d'e-mails à Amazon Cognito

Vous pouvez être amené à accorder des autorisations spécifiques à Amazon Cognito pour lui permettre d'adresser des e-mails à vos utilisateurs. Les autorisations que vous accordez et le processus que vous utilisez pour les accorder varient selon que vous utilisez la configuration de messagerie par défaut ou celle d'Amazon SES.

Pour accorder l'autorisation d'utiliser la configuration de messagerie par défaut

Effectuez cette étape uniquement si vous configurez votre groupe d'utilisateurs pour Envoyer des e-mails avec Cognito ou si vous le définissez sur `EmailSendingAccount.COGNITO_DEFAULT`

Avec la configuration de messagerie par défaut, votre groupe d'utilisateurs peut envoyer des messages électroniques avec l'une des adresses suivantes.

- L'adresse par défaut `no-reply@verificationemail.com`.
- Une adresse FROM personnalisée provenant de vos adresses e-mail ou domaines vérifiés dans Amazon SES.

Si vous utilisez une adresse personnalisée, Amazon Cognito a besoin de permissions supplémentaires pour envoyer des e-mails aux utilisateurs à partir de cette adresse. Ces autorisations sont accordées par une [politique d'autorisation d'envoi](#) pour l'adresse ou le domaine dans Amazon SES. Si vous utilisez la console Amazon Cognito pour ajouter une adresse personnalisée à votre groupe d'utilisateurs, la politique est automatiquement attachée à l'adresse e-

mail vérifiée Amazon SES. Toutefois, si vous configurez votre groupe d'utilisateurs en dehors de la console, par exemple à l'aide de l'API Amazon Cognito, vous devez joindre la politique à l'aide de la [console Amazon SES](#) ou de l'[PutIdentityPolicy](#) API. AWS CLI

Note

Vous pouvez configurer une adresse EXPÉDITEUR uniquement dans un domaine vérifié à l'aide de la commande AWS CLI ou l'API Amazon Cognito.

Une politique d'autorisation d'envoi autorise ou refuse l'accès en fonction des ressources du compte qui utilisent Amazon Cognito pour appeler Amazon SES. Pour plus d'informations sur les politiques basées sur les ressources, consultez le [manuel d'utilisateur IAM](#). Vous pouvez également trouver des exemples de politiques basées sur les ressources dans le [Manuel de développeur Amazon SES](#).

Exemple Politique d'autorisation d'envoi

L'exemple suivant de politique d'autorisation d'envoi accorde à Amazon Cognito la possibilité limitée d'utiliser une identité vérifiée Amazon SES. Amazon Cognito ne peut envoyer des messages électroniques que lorsqu'il le fait au nom du groupe d'utilisateurs dans le `aws:SourceArn` et du compte dans la condition `aws:SourceAccount`.

Regions with Amazon SES

Votre politique d'autorisation d'envoi dans la région du groupe d'utilisateurs ou dans une autre région doit autoriser le principal du service Amazon Cognito à envoyer des e-mails. Reportez-vous au [tableau des régions](#) pour plus d'informations. Si la région de votre groupe d'utilisateurs correspond à au moins une valeur de la région Amazon SES, configurez votre politique d'autorisation d'envoi avec le principal de service mondial dans l'exemple suivant.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "stmnt1234567891234",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "email.cognito-idp.amazonaws.com"
        ]
      }
    }
  ]
}
```

```
    },
    "Action": [
      "SES:SendEmail",
      "SES:SendRawEmail"
    ],
    "Resource": "<your SES identity ARN>",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<your account number>"
      },
      "ArnLike": {
        "aws:SourceArn": "<your user pool ARN>"
      }
    }
  }
]
}
```

Opt-in Regions without Amazon SES

Amazon SES n'est pas disponible dans tous les opt-in Régions AWS où Amazon Cognito est disponible. Le Moyen-Orient (EAU) en est un exemple et ne peut envoyer des e-mails avec des identités vérifiées qu'en Europe (Francfort) (`eu-central-1`). Dans les groupes d'utilisateurs dotés de la configuration d'e-mail par défaut, Amazon Cognito envoie également des e-mails avec une identité vérifiée dans chacune des deux régions. Dans le cas du Moyen-Orient (EAU), la région supplémentaire est l'Europe (Londres). Vous devez mettre à jour la politique d'autorisation d'envoi dans les deux régions.

Votre politique d'autorisation d'envoi dans chacune des régions alternatives doit autoriser le principal du service Amazon Cognito de la région optionnelle du groupe d'utilisateurs à envoyer des e-mails. Reportez-vous au [tableau des régions](#) pour plus d'informations. Si votre région est marquée comme région alternative, configurez vos politiques d'autorisation d'envoi avec le principal de service régional, comme dans l'exemple suivant. Remplacez l'exemple d'identifiant de région `me-central-1` par l'ID de région requis selon les besoins.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
```

```
        "cognito-idp.me-central-1.amazonaws.com"
    ],
    "Action": [
        "SES:SendEmail",
        "SES:SendRawEmail"
    ],
    "Resource": "<your SES identity ARN>",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "<your account number>"
        },
        "ArnLike": {
            "aws:SourceArn": "<your user pool ARN>"
        }
    }
}
]
```

Pour plus d'informations sur la syntaxe de la politique, consultez [Politiques d'autorisation d'envoi Amazon SES](#) dans le Manuel du développeur Amazon Simple Email Service.

Pour d'autres exemples, consultez [Exemples de politiques d'autorisation d'envoi Amazon SES](#) dans le Manuel du développeur Amazon Simple Email Service.

Pour accorder des autorisations d'utilisation de votre configuration Amazon SES

Si vous configurez votre groupe d'utilisateurs afin d'utiliser votre configuration Amazon SES, Amazon Cognito a besoin d'autorisations supplémentaires pour appeler Amazon SES en votre nom au moment d'envoyer des e-mails à vos utilisateurs. Cette autorisation est accordée avec le service IAM.

Quand vous configurez votre groupe d'utilisateurs avec cette option, Amazon Cognito crée un rôle lié à un service, qui est un type de rôle IAM dans votre Compte AWS. Ce rôle contient les autorisations qui permettent à Amazon Cognito d'accéder à Amazon SES et d'envoyer des e-mails avec votre adresse.

Amazon Cognito crée votre rôle lié à un service avec les AWS informations d'identification de la session utilisateur qui définit la configuration. Les autorisations IAM de cette session doivent inclure l'action `iam:CreateServiceLinkedRole`. Pour plus d'informations sur les autorisations dans IAM, consultez la section [Gestion de l'accès aux AWS ressources](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur le rôle lié à un service créé par Amazon Cognito, consultez [Utilisation de rôles liés à un service pour Amazon Cognito](#).

Étape 4 : Configuration de votre groupe d'utilisateurs

Effectuez les étapes ci-après si vous souhaitez configurer votre groupe d'utilisateurs avec l'un des éléments suivants :

- Une adresse d'envoi personnalisée qui apparaît comme étant l'expéditeur de l'e-mail
- Une adresse de réponse personnalisée qui reçoit les messages que vos utilisateurs envoient à votre adresse d'envoi
- Votre configuration Amazon SES

Note

Si votre identité vérifiée est une adresse e-mail, Amazon Cognito définit cette adresse e-mail en tant qu'adresse e-mail FROM et REPLY-TO par défaut. Toutefois, si votre identité vérifiée est un domaine, vous devez fournir une valeur pour les adresses e-mail FROM et REPLY-TO. Par exemple, si votre domaine vérifié est example.com, vous pouvez définir no-reply@example.com comme adresses e-mail FROM et REPLY-TO.

Omettez cette procédure si vous souhaitez utiliser la configuration et l'adresse e-mail par défaut d'Amazon Cognito.

Pour configurer votre groupe d'utilisateurs pour qu'il utilise une adresse e-mail personnalisée

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste.
4. Choisissez l'onglet Messaging (Messagerie), localisez Email configuration (Configuration de la messagerie), choisissez Edit (Modifier).
5. Dans la page Modifier la configuration de la messagerie, sélectionnez Envoyer un e-mail depuis Amazon SES ou Envoyer un e-mail avec Amazon Cognito. Vous pouvez personnaliser la Région SES, Jeu de configurations, et Nom de l'expéditeur DE uniquement lorsque vous choisissez Envoyer un e-mail depuis Amazon SES.

6. Pour utiliser une adresse DE personnalisée, procédez comme suit :
 - a. Sous Région SES, choisissez la région qui contient votre adresse e-mail vérifiée.
 - b. Sous adresse e-mail DE, choisissez votre adresse e-mail. Utilisez une adresse e-mail que vous avez vérifiée avec Amazon SES.
 - c. (Facultatif) Sous Configuration set (Jeu de configurations), choisissez un jeu de configurations qu'Amazon SES utilisera. L'apport et l'enregistrement de cette modification crée un rôle lié à un service.
 - d. (Facultatif) Sous Adresse de l'expéditeur DE, saisissez une adresse e-mail. Vous pouvez fournir uniquement une adresse e-mail, ou une adresse e-mail et un nom générique au format Jane Doe <janedoe@example.com>.
 - e. (Facultatif) Sous Adresse e-mail RÉPONDRE À, saisissez l'adresse e-mail à laquelle vous souhaitez recevoir les messages que vos utilisateurs envoient à votre adresse DE.
7. Sélectionnez Enregistrer les modifications.

Rubriques connexes

- [Personnalisation des messages de vérification d'adresse e-mail](#)
- [Personnalisation des messages d'invitation des utilisateurs](#)

Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito

Certains événements Amazon Cognito pour votre groupe d'utilisateurs peuvent amener Amazon Cognito à environnemeenvoyer des SMS à vos utilisateurs. Par exemple, si vous configurez votre groupe d'utilisateurs de façon à exiger la vérification par téléphone, Amazon Cognito envoie un SMS au moment où un utilisateur crée un compte dans votre application ou réinitialise son mot de passe. En fonction de l'action qui déclenche le SMS, celui-ci contient un code de vérification, un mot de passe temporaire ou un message de bienvenue.

Amazon Cognito utilise Amazon Simple Notification Service (Amazon SNS) pour l'envoi des SMS. Si vous envoyez un SMS via Amazon Cognito ou Amazon SNS pour la première fois, Amazon SNS vous place dans un environnement de test (sandbox). Dans l'environnement de test (sandbox), vous pouvez tester vos applications pour les SMS. Dans l'environnement de test (sandbox), les messages ne peuvent être envoyés qu'à des numéros de téléphone vérifiés.

Amazon SNS facture les SMS. Pour en savoir plus, consultez [Tarification Amazon SNS](#).

Note

En raison du volume du trafic de SMS non sollicités dans le monde entier, certains gouvernements mettent en place des obstacles entre les expéditeurs et les destinataires des SMS. Lorsque vous utilisez des SMS pour la MFA et les mises à jour des utilisateurs, vous devez prendre des mesures supplémentaires pour garantir la réception de vos messages. Vous devez également surveiller les réglementations relatives aux SMS dans les pays de résidence de vos utilisateurs et maintenir la configuration de vos SMS à jour. Pour plus d'informations, consultez [Messagerie texte mobile \(SMS\)](#) dans le Guide du développeur Amazon Simple Notification Service.

L'utilisation de SMS pour authentifier et vérifier les utilisateurs n'est pas une bonne pratique en matière de sécurité. Les numéros de téléphone peuvent changer de propriétaire et peuvent ne pas représenter de manière fiable un facteur de quelque chose que vous avez pour la MFA de vos utilisateurs. Mettez plutôt en place la MFA par TOTP dans votre application ou avec votre utilisateur de fournisseur d'identité (IdP) tiers. Vous pouvez également créer des facteurs d'authentification supplémentaires personnalisés avec [Déclencheurs Lambda création d'une stimulation d'authentification personnalisée](#).

Amazon Cognito envoie des SMS à vos utilisateurs avec un code qu'ils peuvent saisir. Le tableau suivant indique les événements qui peuvent générer un message SMS.

Options de message

Activité	Opération API	Options de livraison	Options de format	Personnalisable	Modèle de message
Mot de passe oublié	ForgotPassword	Courrier électronique, SMS	code	Non	N/A
Invitation	AdminCreateUser	Courrier électronique, SMS	code	Oui	Message d'invitation

Activité	Opération API	Options de livraison	Options de format	Personnalisable	Modèle de message
Auto-enregistrement	SignUp	Courrier électronique, SMS	code, lien	Oui	Message de vérification
Vérification de l'adresse e-mail ou du numéro de téléphone	UpdateUserAttributes	Courrier électronique, SMS	code	Oui	Message de vérification
Authentification multifactorielle (MFA)	AdminInitiateAuth , InitiateAuth	SMS, application d'authentification	code	Oui ¹	Message du MFA

¹ Pour les SMS.

Configuration de l'envoi de SMS pour la première fois dans des groupes d'utilisateurs Amazon Cognito

Amazon Cognito utilise Amazon SNS pour envoyer des SMS à vos groupes d'utilisateurs. Vous pouvez également utiliser un [Déclencheur Lambda de l'expéditeur de SMS personnalisé](#) pour utiliser vos propres ressources afin d'envoyer des SMS. La première fois que vous configurez Amazon SNS pour envoyer des SMS dans une région donnée Région AWS, Amazon SNS vous place Compte AWS dans le sandbox SMS de cette région. Amazon SNS utilise le sandbox pour prévenir les fraudes et les abus et pour répondre aux exigences de conformité. [Lorsque vous êtes dans le sandbox, Amazon SNS impose certaines restrictions.](#) Par exemple, vous pouvez envoyer des SMS à un maximum de 10 numéros de téléphone que vous avez vérifiés avec Amazon SNS. Tant que Compte AWS vous êtes dans le sandbox, n'utilisez pas votre configuration Amazon SNS pour les applications en production. Lorsque vous êtes dans l'environnement de test (sandbox), Amazon Cognito ne peut pas envoyer de SMS aux numéros de téléphone de vos utilisateurs.

Pour envoyer des SMS aux utilisateurs du groupe d'utilisateurs

1. [Préparer un rôle IAM qu'Amazon Cognito peut utiliser pour envoyer des SMS avec Amazon SNS](#)
2. [Choisissez le Région AWS pour les messages SMS Amazon SNS](#)
3. [Obtenir une identité d'origine pour envoyer des SMS à des numéros de téléphone aux États-Unis](#)
4. [Vérifier que vous vous trouvez dans l'environnement de test \(sandbox\) SMS](#)
5. [Sortir votre compte de l'environnement de test \(sandbox\) Amazon SNS](#)
6. [Vérifier les numéros de téléphone pour Amazon Cognito dans Amazon SNS](#)
7. [Terminer la configuration du groupe d'utilisateurs dans Amazon Cognito](#)

Préparer un rôle IAM qu'Amazon Cognito peut utiliser pour envoyer des SMS avec Amazon SNS

Lorsque vous envoyez un SMS depuis votre groupe d'utilisateurs, Amazon Cognito endosse un rôle IAM dans votre compte. Amazon Cognito utilise l'autorisation `sns:Publish` attribuée à ce rôle pour envoyer des SMS à vos utilisateurs. Dans la console Amazon Cognito, vous pouvez définir une sélection de rôle IAM à partir de l'onglet Messaging (Messagerie) de votre groupe d'utilisateurs, dans SMS ou effectuer cette sélection par l'intermédiaire de l'assistant de création de groupe d'utilisateurs.

L'exemple suivant de politique d'approbation de rôle IAM permet à un groupe d'utilisateurs Amazon Cognito une capacité limitée à assumer le rôle. Amazon Cognito peut assumer le rôle uniquement lorsqu'il le fait au nom du groupe d'utilisateurs dans la condition `aws:SourceArn` et du Compte AWS dans la condition `aws:SourceAccount`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "cognito-idp.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<your account number>"
      },
      "ArnLike": {
        "aws:SourceArn": "<your user pool ARN>"
      }
    }
  ]
}
```



```
    }  
  }  
}]  
}
```

Vous pouvez spécifier un [ARN de groupe d'utilisateurs](#) exact ou un ARN générique dans la valeur de la condition `aws:SourceArn`. Recherchez les ARN de vos groupes d'utilisateurs dans AWS Management Console ou avec une demande d'[DescribeUserPoolAPI](#).

Pour plus d'informations sur les rôles IAM et les stratégies d'approbation, veuillez consulter [Termes et concepts relatifs aux rôles](#) dans le Guide de l'utilisateur AWS Identity and Access Management .

Choisissez le Région AWS pour les messages SMS Amazon SNS

Dans certains Régions AWS cas, vous pouvez choisir la région qui contient les ressources Amazon SNS que vous souhaitez utiliser pour les SMS Amazon Cognito. Partout Région AWS où Amazon Cognito est disponible, à l'exception de la région Asie-Pacifique (Séoul), vous pouvez utiliser les ressources Amazon SNS là où vous avez créé votre groupe d'utilisateurs. Région AWS Pour rendre l'envoi de SMS plus rapide et plus fiable lorsque vous avez le choix entre les régions, utilisez les ressources Amazon SNS dans la même région que votre groupe d'utilisateurs.

Note

Dans le AWS Management Console, vous ne pouvez modifier la région pour les ressources SMS qu'une fois que vous êtes passé à la nouvelle expérience de console Amazon Cognito.

Choisissez une région pour les ressources SMS dans l'étape Configure message delivery (Configurer la diffusion des messages) de l'assistant Nouveau groupe d'utilisateurs. Vous pouvez également choisir Edit (Modifier) sous SMS, dans l'onglet Messaging (Messagerie) d'un groupe d'utilisateurs existant.

Lors du lancement, pour certains Régions AWS, Amazon Cognito envoyait des SMS contenant des ressources Amazon SNS dans une autre région. Pour définir votre région préférée, utilisez le `SnsRegion` paramètre de l'[SmsConfigurationType](#) objet de votre groupe d'utilisateurs. Lorsque vous créez par programme une ressource de groupes d'utilisateurs Amazon Cognito dans une région Amazon Cognito dans le tableau suivant et que vous ne fournissez pas de paramètre `SnsRegion`, votre groupe d'utilisateurs peut envoyer des SMS avec des ressources Amazon SNS dans une région Amazon SNS héritée.

Les groupes d'utilisateurs Amazon Cognito en Asie-Pacifique (Séoul) Région AWS doivent utiliser votre configuration Amazon SNS dans la région Asie-Pacifique (Tokyo).

Amazon SNS définit le quota des dépenses pour tous les nouveaux comptes à 1,00 USD par mois. Vous avez peut-être augmenté votre limite de dépenses dans un produit Région AWS que vous utilisez avec Amazon Cognito. Avant de modifier les SMS Région AWS pour Amazon SNS, ouvrez une demande d'augmentation de quota dans le AWS Support Center pour augmenter votre limite dans la nouvelle région. Pour plus d'informations, consultez [Demande d'augmentations de votre quota de dépenses mensuelles pour l'envoi de SMS pour Amazon SNS](#) dans le Guide du développeur Amazon Simple Notification Service.

Vous pouvez envoyer des SMS pour toute région Amazon Cognito figurant dans le tableau suivant avec les ressources Amazon SNS dans la région Amazon SNS correspondante.

Région Amazon Cognito	Région Amazon SNS
USA Est (Ohio)	USA Est (Ohio), USA Est (Virginie du Nord)
Canada (Centre)	Canada (Centre), USA Est (Virginie du Nord)
Europe (Francfort)	Europe (Francfort), Europe (Irlande)
Europe (Londres)	Europe (Londres), Europe (Irlande)
Asie-Pacifique (Séoul)	Asie-Pacifique (Tokyo)
USA Est (Virginie du Nord)	USA Est (Virginie du Nord)
USA Ouest (Californie du Nord)	USA Ouest (Californie du Nord)
US West (Oregon)	USA Ouest (Oregon)
Asie-Pacifique (Mumbai)	Asie-Pacifique (Mumbai), Asie-Pacifique (Singapour)
Asie-Pacifique (Hyderabad)	Asie-Pacifique (Hyderabad)
Asie-Pacifique (Singapour)	Asie-Pacifique (Singapour)
Asie-Pacifique (Sydney)	Asie-Pacifique (Sydney)

Région Amazon Cognito	Région Amazon SNS
Asia Pacific (Tokyo)	Asie-Pacifique (Tokyo)
Asie-Pacifique (Jakarta)	Asie-Pacifique (Jakarta)
Asie-Pacifique (Osaka)	Asie-Pacifique (Osaka)
Asie-Pacifique (Melbourne)	Asie-Pacifique (Melbourne)
Europe (Irlande)	Europe (Irlande)
Europe (Paris)	Europe (Paris)
Europe (Stockholm)	Europe (Stockholm)
Europe (Milan)	Europe (Milan)
Europe (Espagne)	Europe (Espagne)
Moyen-Orient (Bahreïn)	Moyen-Orient (Bahreïn)
Amérique du Sud (Sao Paulo)	Amérique du Sud (São Paulo)
Israël (Tel Aviv)	Israël (Tel Aviv)
Afrique (Le Cap)	Afrique (Le Cap)
Moyen-Orient (EAU)	Moyen-Orient (EAU)
Europe (Zurich)	Europe (Zurich)

Obtenir une identité d'origine pour envoyer des SMS à des numéros de téléphone aux États-Unis

Si vous envisagez d'envoyer des SMS à des numéros de téléphone aux États-Unis, vous devez obtenir une identité d'origine, que vous optiez pour un environnement de test (sandbox) pour SMS ou pour un environnement de production.

Depuis le 1er juin 2021, les opérateurs aux États-Unis exigent une identité d'origine pour envoyer des messages à des numéros de téléphone aux États-Unis. Si vous ne disposez pas encore d'une identité

d'origine, vous devez en obtenir une. Pour savoir comment obtenir une identité d'origine, consultez [Demande de numéro](#) dans le Guide de l'utilisateur Amazon Pinpoint.

Si vous opérez dans les pays suivants Régions AWS, vous devez ouvrir un AWS Support ticket pour obtenir une identité d'origine. Pour obtenir des instructions, consultez [Demande de prise en charge des SMS](#) dans le Manuel du développeur Amazon Simple Notification Service.

- USA Est (Ohio)
- Europe (Stockholm)
- Europe (Paris)
- Europe (Milan)
- Middle East (Bahrain)
- Amérique du Sud (Sao Paulo)
- USA Ouest (Californie du Nord)

Lorsque vous avez plusieurs identités d'origine identiques Région AWS, Amazon SNS choisit un type d'identité d'origine dans l'ordre de priorité suivant : code abrégé, 10 DLC, numéro gratuit. Vous ne pouvez pas modifier cette priorité. Pour plus d'informations, consultez [FAQ Amazon SNS](#).

Vérifier que vous vous trouvez dans l'environnement de test (sandbox) SMS

Suivez la procédure ci-dessous pour confirmer que vous êtes dans l'environnement de test (sandbox) pour SMS. Répétez cette procédure pour Région AWS chaque groupe d'utilisateurs Amazon Cognito en production.

Vérification du statut de l'environnement de test (sandbox) pour SMS dans la console Amazon Cognito

Pour vérifier que vous vous trouvez dans l'environnement de test (sandbox) SMS

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos informations d'identification AWS .
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs dans la liste.
4. Choisissez l'onglet Messaging (Messagerie).

5. Dans la section Configuration SMS, développez Passer à l'environnement de production Amazon SNS. Si votre compte se trouve dans l'environnement de test (sandbox) SMS, le message suivant s'affiche :

```
You are currently in the SMS Sandbox and cannot send SMS messages to unverified numbers.
```

Si ce message ne s'affiche pas, cela signifie que quelqu'un a déjà configuré les SMS dans votre compte. Passez à [Terminer la configuration du groupe d'utilisateurs dans Amazon Cognito](#).

6. Choisissez le lien [Amazon SNS](#) dans le message. Cela ouvre la console Amazon SNS dans un nouvel onglet.
7. Vérifiez que vous vous trouvez dans l'environnement de test (sandbox). Le message de console indique l'état de votre sandbox et Région AWS, comme suit :

```
This account is in the SMS sandbox in US East (N. Virginia).
```

Sortir votre compte de l'environnement de test (sandbox) Amazon SNS

Si vous testez votre application et que vous avez seulement besoin d'envoyer des SMS à des numéros de téléphone que vos administrateurs peuvent vérifier, ignorez cette étape.

Pour utiliser votre application en production, sortez votre compte de l'environnement de test (sandbox) pour SMS et faites-le entrer en production. Après avoir configuré une identité d'origine dans le fichier contenant les ressources Amazon SNS Région AWS que vous souhaitez qu'Amazon Cognito utilise, vous pouvez vérifier les numéros de téléphone américains tout en restant dans le sandbox Compte AWS SMS. Quand votre environnement Amazon SNS est en production, vous n'avez pas besoin de vérifier les numéros de téléphone des utilisateurs dans Amazon SNS pour envoyer des SMS à vos utilisateurs.

Pour des instructions détaillées, consultez [Sortie de l'environnement de test \(sandbox\)](#) dans le Guide du développeur Amazon Simple Notification Service.

Vérifier les numéros de téléphone pour Amazon Cognito dans Amazon SNS

Si vous avez sorti votre compte de l'environnement de test (sandbox) pour SMS, ignorez cette étape.

Lorsque vous vous trouvez dans l'environnement de test (sandbox) pour SMS, vous pouvez envoyer des messages à n'importe quel numéro de téléphone que vous avez vérifié avec Amazon SNS.

Pour vérifier un numéro de téléphone, procédez comme suit :

1. Ajoutez un numéro de téléphone de destination de l'environnement de test (sandbox) dans la section Text messaging (SMS) (SMS) de la console Amazon SNS.
2. Recevez un SMS avec un code au numéro de téléphone que vous avez fourni.
3. Saisissez le code de vérification à partir du SMS dans la console Amazon SNS.

Pour obtenir des instructions détaillées, consultez [Ajout et vérification de numéros de téléphone dans l'environnement de test \(sandbox\) SMS](#) dans le Manuel du développeur Amazon Simple Notification Service.

Note

Amazon SNS limite le nombre de numéros de téléphone de destination que vous pouvez vérifier quand vous êtes dans l'environnement de test (sandbox) pour SMS. Consultez [Environnement de test \(sandbox\) pour SMS](#) dans le Guide du développeur Amazon Simple Notification Service.

Terminer la configuration du groupe d'utilisateurs dans Amazon Cognito

Revenez à l'onglet du navigateur sous lequel vous avez [créé](#) ou [modifié](#) votre groupe d'utilisateurs. Exécutez la procédure . Une fois que vous avez ajouté une configuration SMS à votre groupe d'utilisateurs, Amazon Cognito envoie un message de test à un numéro de téléphone interne pour vérifier que votre configuration fonctionne. Amazon SNS facture chaque SMS de test.

Utilisation des jetons avec des groupes d'utilisateurs

Authentifiez les utilisateurs et accordez l'accès aux ressources avec des jetons. Les champs standard (claims) présents dans les jetons sont des informations sur l'utilisateur. Le jeton d'identification contient des champs standard sur l'identité, tels que le nom d'utilisateur, le nom de famille et l'adresse e-mail. Le jeton d'accès contient des champs standard comme scope que l'utilisateur authentifié peut utiliser pour accéder à des API tierces, aux opérations d'API en libre-service des utilisateurs Amazon Cognito et au [Point de terminaison UserInfo](#). Le jeton d'accès et le jeton d'identification incluent tous deux une demande `cognito:groups` qui contient l'appartenance du groupe de l'utilisateur à votre groupe d'utilisateurs. Pour de plus amples informations sur les groupes de groupes d'utilisateurs, consultez [Ajout de groupes à un groupe d'utilisateurs](#).

Amazon Cognito inclut également des jetons que vous pouvez utiliser pour en obtenir des nouveaux ou révoquer des jetons existants. [Actualisez un jeton](#) pour récupérer de nouveaux jetons d'identification et d'accès. [Révoquez un jeton](#) pour stopper un accès d'utilisateur autorisé par des jetons d'actualisation.

Amazon Cognito émet des jetons sous forme de chaînes codées en Base64. Vous pouvez décoder n'importe quel ID Amazon Cognito ou jeton d'accès du format base64 au format JSON en texte brut. Les jetons d'actualisation Amazon Cognito sont chiffrés, opaques pour les utilisateurs et les administrateurs des groupes d'utilisateurs et ne peuvent être lus que par votre groupe d'utilisateurs.

Authentification avec des jetons

Quand un utilisateur se connecte à votre application, Amazon Cognito vérifie les informations de connexion. Si la connexion est établie, Amazon Cognito crée une session et renvoie un jeton d'identification, d'accès et d'actualisation pour l'utilisateur authentifié. Vous pouvez utiliser les jetons pour accorder à vos utilisateurs l'accès à des ressources et à des API en aval, comme Amazon API Gateway. Vous pouvez également les échanger contre des informations d'identification AWS temporaires pour accéder à d'autres services Services AWS.



Stockage de jetons

Votre application doit pouvoir stocker des jetons de différentes tailles. La taille des jetons peut changer pour des raisons incluant, sans s'y limiter, des revendications supplémentaires, des changements dans les algorithmes de codage et des changements dans les algorithmes de chiffrement. Lorsque vous activez la révocation de jetons dans votre groupe d'utilisateurs, Amazon Cognito ajoute des champs standard supplémentaires aux jetons Web JSON, ce qui augmente leur taille. Les nouvelles revendications `origin_jti` et `jti` sont ajoutées aux jetons d'accès et d'identification. Pour plus d'informations sur la révocation de jetons, consultez [Révocation de jetons](#).

Important

Une bonne pratique consiste à sécuriser tous les jetons en transit et en stockage dans le cadre de votre application. Les jetons peuvent contenir des informations d'identification

personnelle sur vos utilisateurs et des informations sur le modèle de sécurité que vous employez pour votre groupe d'utilisateurs.

Personnalisation des jetons

Vous pouvez personnaliser les jetons d'accès et d'identité qu'Amazon Cognito transmet à votre application. Dans un [Déclencheur Lambda avant génération de jeton](#), vous pouvez ajouter, modifier et supprimer des demandes de jetons. Le déclencheur de pré-génération du jeton est une fonction Lambda à laquelle Amazon Cognito envoie un ensemble de demandes par défaut. Les revendications incluent les champs d'application OAuth 2.0, l'appartenance à un groupe de groupes d'utilisateurs, les attributs des utilisateurs, etc. La fonction peut ensuite profiter de l'occasion pour apporter des modifications lors de l'exécution et renvoyer les demandes de jetons mises à jour à Amazon Cognito.

Des coûts supplémentaires s'appliquent à la personnalisation des jetons d'accès avec les événements de la version 2. Pour plus d'informations, consultez [Tarification d'Amazon Cognito](#).

Rubriques

- [Utilisation du jeton d'identification](#)
- [Utilisation du jeton d'accès](#)
- [Utilisation du jeton actualisation](#)
- [Révocation de jetons](#)
- [Vérification d'un jeton web JSON](#)
- [Jetons de mise en cache](#)

Utilisation du jeton d'identification

Le jeton d'identification est un [jeton Web JSON \(JWT\)](#) qui contient des champs standard sur l'identité de l'utilisateur authentifié, comme `name`, `email` et `phone_number`. Vous pouvez utiliser ces informations d'identité dans votre application. Le jeton d'identification peut aussi servir à authentifier des utilisateurs auprès de vos serveurs de ressources ou applications de serveur. Vous pouvez également utiliser un jeton d'identification en dehors de l'application avec vos opérations d'API web. Dans ce cas, vous devez vérifier la signature du jeton d'identification avant de pouvoir approuver les revendications qu'il contient. veuillez consulter [Vérification d'un jeton web JSON](#).

Vous pouvez définir l'expiration d'un jeton d'identification sur toute valeur comprise entre 5 minutes et 1 jour. Vous pouvez définir cette valeur par client d'application.

Important

Lorsque votre utilisateur se connecte avec l'interface utilisateur hébergée ou un fournisseur d'identité fédérée (IdP), Amazon Cognito définit des cookies de session valides pendant 1 heure. Si vous utilisez l'interface utilisateur hébergée ou la fédération et que vous spécifiez une durée minimale de moins d'une heure pour vos jetons d'accès et d'identification, vos utilisateurs disposent d'une session valide jusqu'à l'expiration du cookie. Si l'utilisateur a des jetons qui expirent pendant la session d'une heure, l'utilisateur peut actualiser ses jetons sans avoir besoin de se réauthentifier.

En-tête de jeton d'identification

L'en-tête contient deux éléments d'information : l'ID de clé (`kid`) et l'algorithme (`alg`).

```
{
  "kid" : "1234example=",
  "alg" : "RS256"
}
```

kid

ID de la clé . Sa valeur indique quelle clé a été utilisée pour sécuriser la signature web JSON (JWS) du jeton. Vous pouvez consulter les identifiants de clé de signature de votre groupe d'utilisateurs sur le point de terminaison `jwtks_uri`.

Pour plus d'informations sur le paramètre `kid`, consultez [Paramètre d'en-tête Key identifier \(kid\)](#).

alg

Algorithme de chiffrement utilisé par Amazon Cognito pour sécuriser le jeton d'accès. Les groupes d'utilisateurs se servent d'un algorithme de chiffrement RS256, qui est une signature RSA avec SHA-256.

Pour plus d'informations sur le paramètre `alg`, consultez [Paramètre d'en-tête Algorithme \(alg\)](#).

Charge utile par défaut du jeton d'identification

Il s'agit d'un exemple de charge utile provenant d'un jeton d'identification. Il contient les demandes sur l'utilisateur authentifié. Pour plus d'informations sur les revendications standard OpenID Connect (OIDC), consultez la liste des revendications standard [OIDC](#). Vous pouvez ajouter des revendications de votre propre design avec un [Déclencheur Lambda avant génération de jeton](#).

```
<header>.{
  "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "cognito:groups": [
    "test-group-a",
    "test-group-b",
    "test-group-c"
  ],
  "email_verified": true,
  "cognito:preferred_role": "arn:aws:iam::111122223333:role/my-test-role",
  "iss": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_example",
  "cognito:username": "my-test-user",
  "middle_name": "Jane",
  "nonce": "abcdefg",
  "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "cognito:roles": [
    "arn:aws:iam::111122223333:role/my-test-role"
  ],
  "aud": "xxxxxxxxxxxxexample",
  "identities": [
    {
      "userId": "amzn1.account.EXAMPLE",
      "providerName": "LoginWithAmazon",
      "providerType": "LoginWithAmazon",
      "issuer": null,
      "primary": "true",
      "dateCreated": "1642699117273"
    }
  ],
  "event_id": "64f513be-32db-42b0-b78e-b02127b4f463",
  "token_use": "id",
  "auth_time": 1676312777,
  "exp": 1676316377,
  "iat": 1676312777,
  "jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "email": "my-test-user@example.com"
}
```

```
.<token signature>
```

sub

Identifiant unique (UUID), ou sujet, de l'utilisateur authentifié. Le nom d'utilisateur n'est peut-être pas unique dans votre groupe d'utilisateurs. Le champ standard sub est le meilleur moyen d'identifier un utilisateur donné.

cognito:groups

Tableau des noms des groupes de groupes d'utilisateurs dont l'utilisateur est membre. Les groupes peuvent être un identifiant que vous présentez à votre application ou ils peuvent générer une demande pour un rôle IAM préféré à partir d'une réserve d'identités.

cognito:preferred_role

ARN du rôle IAM que vous avez associé au groupe de groupes d'utilisateurs de plus haute priorité de l'utilisateur. Pour plus d'informations sur la manière dont votre groupe d'utilisateurs sélectionne la demande de rôle, consultez [Affectation de valeurs de priorité à des groupes](#).

iss

Fournisseur d'identité qui a émis le jeton. Le champ standard a le format suivant :

```
https://cognito-idp.<Region>.amazonaws.com/<your user pool ID>
```

cognito:username

Nom d'utilisateur de l'utilisateur dans votre groupe d'utilisateurs.

nonce

La revendication nonce provient d'un paramètre du même nom que vous pouvez ajouter aux requêtes de votre point de terminaison authorize OAuth 2.0. Lorsque vous ajoutez le paramètre, la revendication nonce est incluse dans le jeton d'identification émis par Amazon Cognito, et vous pouvez l'utiliser pour vous protéger contre les attaques de relecture. Si vous ne fournissez pas de valeur nonce dans votre demande, Amazon Cognito génère et valide automatiquement un nonce lorsque vous vous authentifiez via un fournisseur d'identité tiers, puis l'ajoute en tant que réclamation nonce au jeton d'identification. L'implémentation de la revendication nonce dans Amazon Cognito est basée sur les [normes OIDC](#).

origin_jti

Un identifiant de révocation de jeton associé au jeton d'actualisation de votre utilisateur. Amazon Cognito fait référence à la origin_jti réclamation lorsqu'il vérifie si vous avez

révoqué le jeton de votre utilisateur lors de l'opération [Point de terminaison de révocation](#) ou de l'[RevokeToken](#) API. Lorsque vous révoquez un jeton, Amazon Cognito invalide tous les jetons d'accès et d'identification ayant la même valeur `origin_jti`.

cognito:roles

Tableau des noms des rôles IAM associés aux groupes de l'utilisateur. Un rôle IAM peut être associé à chaque groupe de groupes d'utilisateurs. Ce tableau représente tous les rôles IAM pour les groupes de vos utilisateurs, quelle que soit leur priorité. Pour plus d'informations, consultez [Ajout de groupes à un groupe d'utilisateurs](#).

aud

Client d'application du groupe d'utilisateurs qui a authentifié l'utilisateur. Amazon Cognito affiche la même valeur dans le champ standard `client_id` du jeton d'accès.

identities

Contenu de l'attribut `identities` de l'utilisateur. L'attribut contient des informations sur chaque profil de fournisseur d'identité tiers que vous avez lié à un utilisateur, soit par une connexion fédérée, soit en ayant [lié un utilisateur fédéré à un profil local](#). Ces informations contiennent le nom de leur fournisseur, l'identifiant unique de leur fournisseur et d'autres métadonnées.

token_use

Objectif prévu du jeton. Dans un jeton d'identification, sa valeur est `id`.

auth_time

Date et heure d'authentification, au format horaire Unix, auxquelles l'utilisateur a terminé l'authentification.

exp

Date et heure d'expiration, au format horaire Unix, auxquelles le jeton de l'utilisateur expire.

iat

Date et heure, au format horaire Unix, auxquelles Amazon Cognito a émis le jeton de l'utilisateur.

jti

Identifiant unique du jeton JWT.

Le jeton d'identification peut contenir des revendications standard OIDC définies dans les [Revendications standard OIDC](#). Le jeton d'identification peut également contenir des attributs

personnalisés que vous définissez dans votre groupe d'utilisateurs. Amazon Cognito écrit des valeurs d'attribut personnalisées dans le jeton d'ID sous forme de chaînes quel que soit le type d'attribut.

Note

Les attributs personnalisés du groupe d'utilisateurs sont toujours préfixés par `custom:`.

Signature de jeton d'identité

La signature du jeton d'identification est calculée en fonction de l'en-tête et de la charge utile du jeton JWT. Avant d'accepter les champs standard figurant dans un jeton d'identification reçu par votre application, vérifiez la signature du jeton. Pour plus d'informations, consultez [Vérification d'un jeton Web JSON](#).

Utilisation du jeton d'accès

Le jeton d'accès de groupe d'utilisateurs contient les demandes sur l'utilisateur authentifié, une liste des groupes de l'utilisateur et une liste des portées. L'objectif du jeton d'accès est d'autoriser des opérations d'API. Votre groupe d'utilisateurs accepte les jetons d'accès pour autoriser les opérations en libre-service des utilisateurs. Par exemple, vous pouvez utiliser le jeton d'accès pour accorder à votre utilisateur un accès lui permettant d'ajouter, de modifier ou de supprimer des attributs utilisateur.

Avec les [étendues OAuth 2.0](#) dans un jeton d'accès, dérivées des étendues personnalisées que vous ajoutez à votre groupe d'utilisateurs, vous pouvez autoriser l'utilisateur à récupérer des informations à partir d'une API. Par exemple, Amazon API Gateway prend en charge l'autorisation avec les jetons d'accès Amazon Cognito. Vous pouvez renseigner un mécanisme d'autorisation d'API REST avec des informations provenant de votre groupe d'utilisateurs, ou utiliser Amazon Cognito comme mécanisme d'autorisation de jetons Web JSON (JWT) pour une API HTTP. Pour générer un jeton d'accès avec des étendues personnalisées, vous devez le demander via les [points de terminaison publics](#) de votre groupe d'utilisateurs.

Le jeton d'accès de l'utilisateur est une autorisation de demander plus d'informations sur les attributs de l'utilisateur auprès du [Point de terminaison UserInfo](#). Le jeton d'accès de l'utilisateur permet également de lire et d'écrire les attributs de l'utilisateur. Le niveau d'accès aux attributs que votre jeton d'accès accorde dépend des autorisations que vous attribuez à votre client d'application et des étendues que vous accordez dans le jeton.

Le jeton d'accès est un [jeton Web JSON \(JWT\)](#). L'en-tête du jeton d'accès a la même structure que celui du jeton d'identification. Amazon Cognito signe les jetons d'accès avec une clé différente de celle utilisée pour signer les jetons d'identification. La valeur d'un champ standard d'identification de clé d'accès (`kid`) ne correspond pas à la valeur du champ standard `kid` d'un jeton d'identification provenant de la même session utilisateur. Dans le code de votre application, vérifiez indépendamment les jetons d'identification et les jetons d'accès. Ne faites pas confiance aux champs standard d'un jeton d'accès tant que vous n'avez pas vérifié la signature. Pour plus d'informations, consultez [Vérification d'un jeton web JSON](#). Vous pouvez définir l'expiration d'un jeton d'accès sur toute valeur comprise entre 5 minutes et 1 jour. Vous pouvez définir cette valeur par client d'application.

Important

Pour les jetons d'accès et d'identification, ne spécifiez pas une valeur minimale inférieure à une heure si vous utilisez l'interface utilisateur hébergée. Amazon Cognito HostedUI utilise des cookies valides pendant une heure. Si vous entrez un minimum de moins d'une heure, vous n'obtiendrez pas de temps d'expiration inférieur.

En tête de jeton d'accès

L'en-tête contient deux éléments d'information : l'ID de clé (`kid`) et l'algorithme (`alg`).

```
{
  "kid" : "1234example="
  "alg" : "RS256",
}
```

kid

ID de la clé . Sa valeur indique quelle clé a été utilisée pour sécuriser la signature web JSON (JWS) du jeton. Vous pouvez consulter les identifiants de clé de signature de votre groupe d'utilisateurs sur le point de terminaison `jwtks_uri`.

Pour plus d'informations sur le paramètre `kid`, consultez [Paramètre d'en-tête Key identifier \(kid\)](#).

alg

Algorithme de chiffrement utilisé par Amazon Cognito pour sécuriser le jeton d'accès. Les groupes d'utilisateurs se servent d'un algorithme de chiffrement RS256, qui est une signature RSA avec SHA-256.

Pour plus d'informations sur le paramètre `alg`, consultez [Paramètre d'en-tête Algorithme \(alg\)](#).

Charge utile par défaut du jeton d'accès

Voici un exemple de la charge utile d'un jeton d'accès. Pour plus d'informations, consultez [Demandes JWT](#). Vous pouvez ajouter des revendications de votre propre design avec un [Déclencheur Lambda avant génération de jeton](#).

```
<header>.
{
  "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "device_key": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "cognito:groups": [
    "testgroup"
  ],
  "iss": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_example",
  "version": 2,
  "client_id": "xxxxxxxxxxxxexample",
  "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "event_id": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "token_use": "access",
  "scope": "phone openid profile resourceserver.1/appclient2 email",
  "auth_time": 1676313851,
  "exp": 1676317451,
  "iat": 1676313851,
  "jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "username": "my-test-user"
}
.<token signature>
```

sub

Identifiant unique (UUID), ou sujet, de l'utilisateur authentifié. Le nom d'utilisateur n'est peut-être pas unique dans votre groupe d'utilisateurs. Le champ standard `sub` est le meilleur moyen d'identifier un utilisateur donné.

cognito:groups

Tableau des noms des groupes de groupes d'utilisateurs dont l'utilisateur est membre.

iss

Fournisseur d'identité qui a émis le jeton. Le champ standard a le format suivant :

`https://cognito-idp.<Region>.amazonaws.com/<your user pool ID>`

client_id

Client d'application du groupe d'utilisateurs qui a authentifié l'utilisateur. Amazon Cognito affiche la même valeur dans le champ standard au-dessus du jeton d'identification.

origin_jti

Un identifiant de révocation de jeton associé au jeton d'actualisation de votre utilisateur. Amazon Cognito fait référence à la `origin_jti` réclamation lorsqu'il vérifie si vous avez révoqué le jeton de votre utilisateur lors de l'opération [Point de terminaison de révocation](#) ou de l'[RevokeToken](#) API. Lorsque vous révoquez un jeton, Amazon Cognito invalide tous les jetons d'accès et d'identification ayant la même valeur `origin_jti`.

token_use

Objectif prévu du jeton. Dans un jeton d'accès, sa valeur est `access`.

scope

Liste des étendues OAuth 2.0 qui définissent l'accès fourni par le jeton. Un jeton provenant du [Point de terminaison de jeton](#) peut contenir toutes les étendues prises en charge par votre client d'application. Un jeton provenant de la connexion à l'API Amazon Cognito contient uniquement l'étendue `aws.cognito.signin.user.admin`.

auth_time

Date et heure d'authentification, au format horaire Unix, auxquelles l'utilisateur a terminé l'authentification.

exp

Date et heure d'expiration, au format horaire Unix, auxquelles le jeton de l'utilisateur expire.

iat

Date et heure, au format horaire Unix, auxquelles Amazon Cognito a émis le jeton de l'utilisateur.

jti

Identifiant unique du jeton JWT.

username

Nom d'utilisateur de l'utilisateur dans votre groupe d'utilisateurs.

Signature du jeton d'accès

La signature du jeton d'accès est calculée en fonction de l'en-tête et de la charge utile du jeton JWT. Lorsqu'elle est utilisée à l'extérieur d'une application dans vos API web, vous devez toujours vérifier cette signature avant d'accepter le jeton. Pour plus d'informations, consultez [Vérification d'un jeton web JSON](#).

Utilisation du jeton actualisation

Vous pouvez utiliser le jeton d'actualisation pour récupérer de nouveaux jetons d'identification et d'accès. Par défaut, le jeton d'actualisation expire 30 jours après que l'utilisateur de votre application s'est connecté à votre groupe d'utilisateurs. Lorsque vous créez une application pour votre groupe d'utilisateurs, vous pouvez définir le délai d'expiration de son jeton d'actualisation sur une valeur comprise entre 60 minutes et 10 jours.

Les kits SDK Mobile pour iOS et Android, ainsi qu'Amplify pour iOS, Android et Flutter actualisent automatiquement votre identifiant et vos jetons d'accès si un jeton d'actualisation valide (non expiré) est présent. Les jetons d'identification et d'accès ont une validité restante minimale de 2 minutes. Si le jeton d'actualisation est expiré, l'utilisateur de votre application doit se réauthentifier en se connectant à nouveau à votre groupe d'utilisateurs. Si la valeur minimale des jetons d'accès et d'identification est 5 minutes et que vous utilisez le kit SDK, le jeton d'actualisation est utilisé en continu pour récupérer les nouveaux jetons d'accès et d'identification. Vous verrez le comportement attendu au bout de 7 minutes plutôt que 5 minutes.

Le compte de l'utilisateur proprement dit n'expire pas, à condition que l'utilisateur se soit connecté au moins une fois avant la limite de temps définie par `UnusedAccountValidityDays` pour les nouveaux comptes.

Obtenir de nouveaux jetons d'accès et d'identité à l'aide d'un jeton d'actualisation

Utiliser l'API ou l'interface utilisateur hébergée pour lancer l'authentification des jetons d'actualisation.

Pour utiliser le jeton d'actualisation afin d'obtenir un nouvel identifiant et des jetons d'accès avec l'API des groupes d'utilisateurs, utilisez les opérations de l'[InitiateAuth](#) API [AdminInitiateAuthor](#). Transmettez `REFRESH_TOKEN_AUTH` pour le paramètre `AuthFlow`. Dans la propriété `AuthParameters` de `AuthFlow`, transmettez le jeton d'actualisation de votre utilisateur en tant que valeur de `"REFRESH_TOKEN"`. Amazon Cognito renvoie des nouveaux jetons d'identification et d'accès une fois que votre demande d'API a réussi tous les défis.

Note

Pour utiliser l'API des groupes d'utilisateurs Amazon Cognito afin d'actualiser les jetons pour un utilisateur de l'interface utilisateur hébergée, générez une demande `InitiateAuth`.

Vous pouvez également envoyer des jetons d'actualisation au [Point de terminaison de jeton](#) dans un groupe d'utilisateurs où vous avez configuré un domaine. Dans le corps de la demande, incluez une valeur `grant_type` de `refresh_token` et une valeur `refresh_token` du jeton d'actualisation de votre utilisateur.

Révocation de jetons d'actualisation

Vous pouvez révoquer des jetons d'actualisation appartenant à un utilisateur. Pour plus d'informations sur la révocation de jetons, consultez [Révocation de jetons](#).

Note

La révocation du jeton d'actualisation révoquera tous les identifiants et jetons d'accès qu'Amazon Cognito a émis à la suite de demandes d'actualisation avec ce jeton.

Il se peut que les utilisateurs soient déconnectés de tous les appareils auxquels ils sont connectés lorsque vous révoquez les jetons de tous les autorisateurs à l'aide des opérations de l'API `GlobalSignOut` et `AdminUserGlobalSignOut`. Une fois que l'utilisateur est déconnecté, les effets suivants se produisent.

- Le jeton d'actualisation ne peut pas obtenir de nouveaux jetons pour l'utilisateur.
- Le jeton d'accès de l'utilisateur ne peut pas effectuer de demandes d'API autorisées par jeton.
- L'utilisateur doit s'authentifier à nouveau pour obtenir de nouveaux jetons. Étant donné que les cookies de session de l'interface utilisateur hébergée n'expirent pas automatiquement, votre

utilisateur peut s'authentifier à nouveau à l'aide d'un cookie de session, sans qu'il soit nécessaire de saisir des informations d'identification supplémentaires. Après avoir déconnecté les utilisateurs de votre interface utilisateur hébergée, redirigez-les vers [Point de terminaison de déconnexion](#), où Amazon Cognito effacera son cookie de session.

Les jetons d'actualisation vous permettent de maintenir les sessions des utilisateurs dans votre application pendant une longue période. Au fil du temps, vos utilisateurs souhaiteront peut-être annuler l'autorisation de certains appareils sur lesquels ils se sont connectés, en actualisant continuellement leur session. Pour déconnecter votre utilisateur d'un seul appareil, révoquez son jeton d'actualisation. Lorsque votre utilisateur souhaite se déconnecter de toutes les sessions authentifiées, générez une demande d'[GlobalSignOut](#) API. Votre application peut proposer à votre utilisateur un choix tel que Se déconnecter de tous les appareils. `GlobalSignOut` accepte le jeton d'accès valide, non modifié, non expiré et non révoqué, d'un utilisateur. Cette API étant autorisée par jeton, un utilisateur ne peut pas l'utiliser pour initier la déconnexion d'un autre utilisateur.

Vous pouvez toutefois générer une demande d'[AdminUserGlobalSignOut](#) API que vous autorisez à l'aide de vos AWS informations d'identification pour déconnecter un utilisateur de tous ses appareils. L'application d'administration doit appeler cette opération d'API avec les informations d'identification du AWS développeur et transmettre l'ID du groupe d'utilisateurs et le nom d'utilisateur de l'utilisateur en tant que paramètres. L'API `AdminUserGlobalSignOut` permet de déconnecter tous les utilisateurs du groupe.

Pour plus d'informations sur les demandes que vous pouvez autoriser à l'aide des AWS informations d'identification ou du jeton d'accès d'un utilisateur, consultez [Opérations d'API authentifiées et non authentifiées des groupes d'utilisateurs Amazon Cognito](#).

Révocation de jetons

Vous pouvez révoquer un jeton d'actualisation pour un utilisateur à l'aide de l' AWS API. Lorsque vous révoquez un jeton d'actualisation, tous les jetons d'accès précédemment émis par celui-ci deviennent non valides. Les autres jetons d'actualisation émis à l'utilisateur ne sont pas affectés.

Note

Les [jetons JWT](#) sont autonomes avec une signature et une heure d'expiration assignée lors de la création du jeton. Il n'est pas possible d'utiliser des jetons révoqués avec des appels

d'API Amazon Cognito nécessitant un jeton. Cependant, les jetons révoqués restent valides s'ils sont vérifiés à l'aide d'une bibliothèque JWT qui vérifie leur signature et leur expiration.

Vous pouvez révoquer un jeton d'actualisation pour un client de groupe d'utilisateurs pour lequel la révocation de jeton est activée. Lorsque vous créez un client de groupe d'utilisateurs, la révocation de jetons est activée par défaut.

Activer la révocation de jetons

Pour pouvoir révoquer un jetons pour un client de groupe d'utilisateurs existant, vous devez activer la révocation de jetons. Vous pouvez activer la révocation des jetons pour les clients du groupe d'utilisateurs existants à l'aide de l'API AWS CLI ou de l' AWS API. Pour ce faire, appelez la Commande de l'interface de ligne de commande `aws cognito-idp describe-user-pool-client` ou l'opération de l'API `DescribeUserPoolClient` pour récupérer les paramètres actuels de votre client d'application. Appelez ensuite la commande de l'interface de ligne de commande `aws cognito-idp update-user-pool-client` ou l'opération de l'API `UpdateUserPoolClient`. Incluez les paramètres actuels de votre client d'application et définissez le paramètre `EnableTokenRevocation` sur `true`.

Lorsque vous créez un nouveau client de groupe d'utilisateurs à l'aide de l' AWS Management Console API AWS CLI, de la ou de l' AWS API, la révocation des jetons est activée par défaut.

Une fois que vous avez activé la révocation de jetons, de nouveaux champs standard sont ajoutés dans les jetons Web JSON Amazon Cognito. Les revendications `origin_jti` et `jti` sont ajoutées aux jetons d'accès et d'identification. Ces revendications augmentent la taille des jetons d'accès et d'identification du client d'application.

Pour créer ou modifier un client d'application avec la révocation des jetons activée, incluez le paramètre suivant dans votre demande [CreateUserPoolClient](#) ou dans votre demande [UpdateUserPoolClient](#) d'API.

```
"EnableTokenRevocation": true
```

Révocation d'un jeton

Vous pouvez révoquer un jeton d'actualisation à l'aide d'une demande d'[RevokeToken](#) API, par exemple à l'aide de la commande `aws cognito-idp revoke-token` CLI. Vous pouvez

également révoquer des jetons à l'aide du [Point de terminaison de révocation](#). Ce point de terminaison est disponible après l'ajout d'un domaine à votre groupe d'utilisateurs. Vous pouvez utiliser le point de terminaison de révocation sur un domaine hébergé par Amazon Cognito ou sur votre propre domaine personnalisé.

Note

Votre demande de révocation d'un jeton d'actualisation doit inclure l'identifiant client utilisé pour obtenir le jeton.

L'exemple qui suit illustre un exemple de demande d'API RevokeToken.

```
{
  "ClientId": "1example23456789",
  "ClientSecret": "abcdef123456789ghijklexample",
  "Token": "eyJjdHkiOiJKV1QiEXAMPLE"
}
```

Voici un exemple de demande cURL adressée au point de terminaison `/oauth2/revoke` d'un groupe d'utilisateurs avec un domaine personnalisé.

```
curl --location 'auth.mydomain.com/oauth2/revoke' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic Base64Encode(client_id:client_secret)' \
--data-urlencode 'token=abcdef123456789ghijklexample' \
--data-urlencode 'client_id=1example23456789'
```

L'opération RevokeToken et le point de terminaison `/oauth2/revoke` ne nécessitent aucune autorisation supplémentaire, sauf si votre client d'application possède un secret client.

Vérification d'un jeton web JSON

Ces étapes décrivent la vérification d'un jeton web JSON (JWT) de groupe d'utilisateurs.

Rubriques

- [Prérequis](#)
- [Validez les jetons avec aws-jwt-verify](#)
- [Comprendre et inspecter les jetons](#)

Prérequis

Les tâches décrites dans cette section sont peut-être déjà traitées par votre bibliothèque, votre kit SDK ou votre infrastructure logicielle. AWS Les SDK fournissent des outils pour le traitement et la gestion des jetons du pool d'utilisateurs Amazon Cognito dans votre application. AWS Amplify inclut des fonctions permettant de récupérer et d'actualiser les jetons Amazon Cognito.

Pour plus d'informations, consultez les pages suivantes.

- [Intégration de l'authentification et de l'autorisation Amazon Cognito avec des applications Web et mobiles](#)
- [Exemples de code pour le fournisseur d'identité Amazon Cognito utilisant des SDK AWS](#)
- [Advanced workflows](#) (Flux de travail avancés) dans le Centre de développement Amplify

De nombreuses bibliothèques sont disponibles pour le décodage et la vérification d'un jeton web JSON (JWT). De telles bibliothèques peuvent être utiles si vous voulez traiter manuellement des jetons pour le traitement d'API côté serveur, ou si vous utilisez d'autres langages de programmation. Consultez la [liste de bibliothèques OpenID Foundation pour la gestion des jetons JWT](#).

Validez les jetons avec aws-jwt-verify

Dans une application Node.js, AWS recommande à la [aws-jwt-verifybibliothèque](#) de valider les paramètres du jeton que l'utilisateur transmet à votre application. Avec `aws-jwt-verify`, vous pouvez renseigner un `CognitoJwtVerifier` avec les valeurs des champs standard que vous souhaitez vérifier pour un ou plusieurs groupes d'utilisateurs. Parmi les valeurs qu'il peut vérifier figurent les suivantes :

- Les jetons d'accès ou d'identification ne sont pas mal formés ou n'ont pas expiré et possèdent une signature valide.
- Les jetons d'accès provenaient des [groupes d'utilisateurs et des clients d'application appropriés](#).
- Les champs standard de jeton d'accès contiennent les [étendues OAuth 2.0 correctes](#).
- Les clés qui ont signé vos jetons d'accès et d'identification [correspondent à une clé de signature kid provenant de l'URI JWKS de vos groupes d'utilisateurs](#).

L'URI JWKS contient des informations publiques sur la clé privée utilisée pour signer le jeton de l'utilisateur. Vous trouverez l'URI JWKS de votre groupe d'utilisateurs à l'adresse `https://cognito-idp.<Region>.amazonaws.com/<userPoolId>/well-known/jwks.json`.

Pour plus d'informations et des exemples de code que vous pouvez utiliser dans une application Node.js ou un AWS Lambda autorisateur, reportez-vous [aws-jwt-verify](#) à la section suivante. GitHub

Comprendre et inspecter les jetons

Avant d'intégrer l'inspection des jetons à votre application, réfléchissez à la manière dont Amazon Cognito assemble les jetons JWT. Récupérez des exemples de jetons de votre groupe d'utilisateurs. Décodez et examinez-les en détail pour comprendre leurs caractéristiques et déterminer ce que vous souhaitez vérifier et à quel moment. Par exemple, vous pouvez examiner l'appartenance à un groupe dans un scénario et les étendues dans un autre.

Les sections suivantes décrivent un processus permettant d'inspecter manuellement les jetons JWT Amazon Cognito lorsque vous préparez votre application.

Vérification de la structure du jeton JWT.

Un jeton Web JSON (JWT) comprend trois sections séparées par un séparateur `.` (point).

En-tête

L'ID de clé, `kid`, et l'algorithme RSA, `alg`, utilisés par Amazon Cognito pour signer le jeton. Amazon Cognito signe les jetons avec un `alg` ayant pour valeur `RS256`.

Charge utile

Champs standard du jeton. Dans un jeton d'identification, les champs standard incluent des attributs de l'utilisateur et des informations sur le groupe d'utilisateurs, `iss`, et le client d'application, `aud`. Dans un jeton d'accès, la charge utile inclut les étendues, l'appartenance à un groupe, votre groupe d'utilisateurs en tant que `iss`, et votre client d'application en tant que `client_id`.

Signature

La signature n'est pas décodable en base64 comme l'en-tête et la charge utile. Il s'agit d'un identifiant RSA256 dérivé d'une clé de signature et de paramètres que vous pouvez observer sur votre URI JWKS.

L'en-tête et la charge utile sont au format JSON codé en base64. Vous pouvez les identifier grâce aux premiers caractères `eyJ` dont la forme déchiffrée correspond au caractère ouvrant `{`. Si l'utilisateur présente un jeton JWT codé en base64 à votre application et qu'il n'est pas au format `[JSON Header].[JSON Payload].[Signature]`, ce n'est pas un jeton Amazon Cognito valide et vous pouvez l'ignorer.

Validation du jeton JWT

La signature du jeton JWT est une combinaison hachée de l'en-tête et de la charge utile. Amazon Cognito génère deux paires de clés de chiffrement RSA pour chaque groupe d'utilisateurs. Une clé privée signe les jetons d'accès et l'autre signe les jetons d'identification.

Pour vérifier la signature d'un jeton JWT

1. Décodez le jeton d'identification.

OpenID Foundation gère également une [liste de bibliothèques pour la gestion des jetons JWT](#).

Vous pouvez également l'utiliser AWS Lambda pour décoder les JWT du groupe d'utilisateurs. Pour plus d'informations, consultez [Décoder et vérifier les jetons Amazon Cognito JWT](#) à l'aide de. AWS Lambda

2. Comparez l'ID de clé local (kid) à l'identifiant kid public.
 - a. Téléchargez et stockez la clé web JSON (JWK) publique correspondante pour votre groupe d'utilisateurs. Elle est disponible dans le cadre d'un ensemble de clés web JSON (JWKS). Vous pouvez le localiser en construisant l'URI `jwtks_uri` suivant pour votre environnement :

```
https://cognito-idp.<Region>.amazonaws.com/<userPoolId>/well-known/jwks.json
```

Pour plus d'informations sur les clés JWK et les ensembles de clés JWK, consultez [JSON Web Key \(JWK\)](#).

Note

Amazon Cognito peut effectuer une rotation des clés de signature dans votre groupe d'utilisateurs. Une bonne pratique consiste à mettre en cache les clés publiques de votre application, en utilisant l'identifiant `kid` comme clé de cache, et à actualiser régulièrement le cache. Comparez l'identifiant `kid` dans les jetons que votre application reçoit à votre cache.

Si vous recevez un jeton avec l'émetteur approprié, mais un identifiant `kid` différent, Amazon Cognito a peut-être effectué la rotation de la clé de signature. Actualisez le cache à partir du point de terminaison `jwtks_uri` de votre groupe d'utilisateurs.

Ceci est un exemple de fichier `jwtks.json` :


```
{
  "keys": [{
    "kid": "1234example=",
    "alg": "RS256",
    "kty": "RSA",
    "e": "AQAB",
    "n": "1234567890",
    "use": "sig"
  }, {
    "kid": "5678example=",
    "alg": "RS256",
    "kty": "RSA",
    "e": "AQAB",
    "n": "987654321",
    "use": "sig"
  }]
}
```

ID de clé (**kid**)

L'identifiant `kid` indique quelle clé a été utilisée pour sécuriser la signature Web JSON (JWS) du jeton.

Algorithme (**alg**)

Le paramètre d'en-tête `alg` indique l'algorithme de chiffrement utilisé pour sécuriser le jeton d'identification. Les groupes d'utilisateurs se servent d'un algorithme de chiffrement RS256, qui est une signature RSA avec SHA-256. Pour plus d'informations sur RSA, consultez [Chiffrement RSA](#).

Type de clé (**kty**)

Le paramètre `kty` identifie la famille d'algorithmes de chiffrement utilisée avec la clé, comme « RSA » dans cet exemple.

Exposant RSA (**e**)

Le paramètre `e` contient la valeur d'exposant pour la clé publique RSA. Il est représenté sous la forme d'une valeur encodée en Base64urlUInt.

Modulo RSA (**n**)

Le paramètre `n` contient la valeur de module pour la clé publique RSA. Il est représenté sous la forme d'une valeur encodée en Base64urlUInt.

Use (**use**)

Le paramètre `use` décrit l'utilisation prévue de la clé publique. Dans cet exemple, la valeur `use sig` représente la signature.

- b. Recherchez dans la clé Web JSON publique un identifiant `kid` correspondant à l'identifiant `kid` de votre jeton JWT.
3. Utilisez une bibliothèque JWT pour comparer la signature de l'émetteur à celle du jeton. La signature du diffuseur est dérivée de la clé publique (le module RSA)"`n`") du `kid` dans `jwt.json` qui correspond au jeton `kid`. Vous pouvez avoir besoin de convertir d'abord la clé JWK au format PEM. L'exemple suivant utilise le jeton JWT, la clé JWK et la bibliothèque Node.js, [jsonwebtoken](#), pour vérifier la signature du jeton JWT :

Node.js

```
var jwt = require('jsonwebtoken');
var jwkToPem = require('jwk-to-pem');
var pem = jwkToPem(jwk);
jwt.verify(token, pem, { algorithms: ['RS256'] }, function(err, decodedToken) {
});
```

Vérification des demandes.

Pour vérifier les demandes de jeton JWT.

1. À l'aide de l'une des méthodes suivantes, vérifiez que le jeton n'a pas expiré.
 - a. Décodez le jeton et comparez le champ standard `exp` à l'heure actuelle.
 - b. Si votre jeton d'accès inclut une `aws.cognito.signin.user.admin` réclamation, envoyez une demande à une API telle que [GetUser](#). Les demandes d'API que vous [autorisez avec un jeton d'accès](#) renvoient une erreur si votre jeton a expiré.
 - c. Présentez votre jeton d'accès dans une demande adressée au [Point de terminaison UserInfo](#). Votre demande renvoie une erreur si votre jeton a expiré.

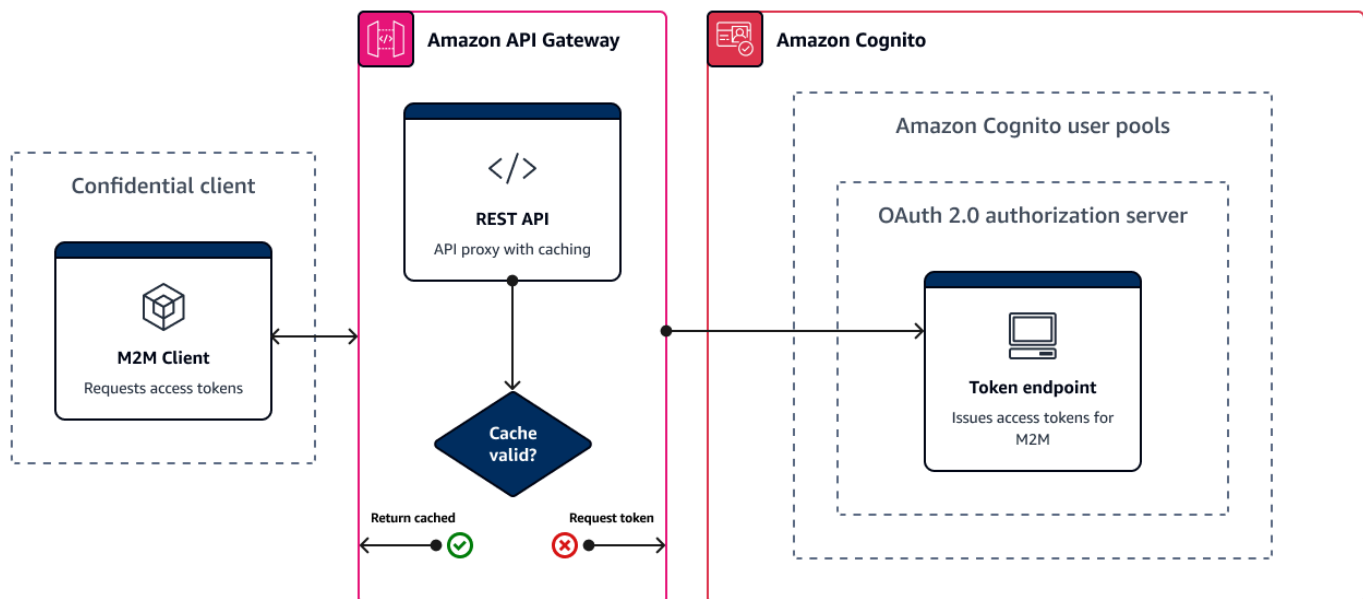
- La demande `aud` dans un jeton d'identification et la demande `client_id` dans un jeton d'accès doivent correspondre à l'ID de client d'application créé dans le groupe d'utilisateurs Amazon Cognito.
- La demande du diffuseur (`iss`) doit correspondre à votre groupe d'utilisateurs. Par exemple, un groupe d'utilisateurs créé dans la région `us-east-1` a la valeur `iss` suivante :

`https://cognito-idp.us-east-1.amazonaws.com/<userpoolID>`.

- Vérifiez la demande `token_use`.
 - Si vous acceptez uniquement le jeton d'accès dans vos opérations de l'API web, sa valeur doit être `access`.
 - Si vous utilisez uniquement le jeton d'identification, sa valeur doit indiquer `id`.
 - Si vous utilisez des jetons d'identification et d'accès, la demande `token_use` doit être `id` ou `access`.

Vous pouvez désormais approuver les demandes à l'intérieur du jeton.

Jetons de mise en cache



Votre application doit exécuter avec succès l'une des demandes suivantes chaque fois que vous souhaitez obtenir un nouveau jeton Web JSON (JWT).

- Demandez des informations d'identification client ou un [octroi](#) de code d'autorisation à partir du [Point de terminaison de jeton](#).
- Demandez un octroi implicite à partir de votre interface utilisateur hébergée.
- Authentifiez un utilisateur local dans une demande d'API Amazon Cognito telle que. [InitiateAuth](#)

Vous pouvez configurer votre groupe d'utilisateurs de manière à ce que les jetons expirent en fonction d'une valeur exprimée en minutes, en heures ou en jours. Pour garantir les performances et la disponibilité de votre application, utilisez les jetons Amazon Cognito jusqu'à leur expiration, puis récupérez de nouveaux jetons. Une solution de cache que vous créez pour votre application permet de conserver les jetons disponibles et empêche le rejet des demandes par Amazon Cognito lorsque votre taux de demandes est trop élevé. Une application côté client doit stocker les jetons dans un cache mémoire. Une application côté serveur peut ajouter un mécanisme de cache chiffré pour stocker les jetons.

Lorsque votre groupe d'utilisateurs génère un volume élevé d'utilisateurs ou d' machine-to-machine activités, vous pouvez rencontrer les limites fixées par Amazon Cognito quant au nombre de demandes de jetons que vous pouvez effectuer. Pour réduire le nombre de demandes que vous envoyez aux points de terminaison Amazon Cognito, vous pouvez stocker et réutiliser les données d'authentification de manière sécurisée ou mettre en œuvre un backoff exponentiel et des nouvelles tentatives.

Les données d'authentification proviennent de deux classes de point de terminaison. Les [points de terminaison OAuth 2.0](#) Amazon Cognito comprennent le point de terminaison du jeton, qui gère les informations d'identification du client et les demandes de code d'autorisation de l'interface utilisateur hébergée. Les [points de terminaison de service](#) répondent à des demandes d'API de groupes d'utilisateurs comme `InitiateAuth` et `RespondToAuthChallenge`. Chaque type de demande possède sa propre limite. Pour en savoir plus sur les limites, consultez [Quotas dans Amazon Cognito](#).

Mise en cache des jetons machine-to-machine d'accès avec Amazon API Gateway

Grâce à la mise en cache des jetons API Gateway, votre application peut faire l'objet d'une mise à l'échelle en réponse à des événements plus importants que le quota de taux de demandes par défaut des points de terminaison OAuth d'Amazon Cognito.

Vous pouvez mettre en cache les jetons d'accès afin que votre application demande un nouveau jeton d'accès uniquement si un jeton mis en cache a expiré. Sinon, votre point de terminaison de mise en cache renvoie un jeton depuis le cache. Cela empêche tout appel supplémentaire vers un point de terminaison d'API Amazon Cognito. Lorsque vous utilisez Amazon API Gateway en tant

que proxy pour le [Point de terminaison de jeton](#), votre API répond à la majorité des demandes qui, dans le cas contraire, contribueraient à votre quota de demandes. Cela permet d'éviter les demandes infructueuses liées à la limitation du taux.

La solution suivante, basée sur l'API Gateway, propose une mise en œuvre de la mise en cache des jetons à faible latence, à codage faible ou sans codage. Les API API Gateway sont chiffrées en transit et, éventuellement, au repos. Un cache API Gateway est idéal pour l'octroi d'informations d'[identification au client OAuth 2.0, un type de subvention](#) souvent important qui produit des jetons d'accès pour autoriser machine-to-machine des sessions de microservice. Dans un cas tel qu'une augmentation du trafic entraînant une mise à l'échelle horizontale de vos microservices, vous pouvez vous retrouver avec de nombreux systèmes utilisant les mêmes informations d'identification client à un volume supérieur à la limite de AWS taux de demandes de votre groupe d'utilisateurs ou de votre client d'application. Pour préserver la disponibilité des applications et une faible latence, une solution de mise en cache est la bonne pratique à appliquer dans de tels scénarios.

Dans cette solution, vous définissez un cache dans votre API afin de stocker un jeton d'accès distinct pour chaque combinaison de portées OAuth et de client d'application que vous souhaitez demander dans votre application. Lorsque votre application fait une demande qui correspond à la clé de cache, votre API répond avec un jeton d'accès qu'Amazon Cognito a émis à la première demande correspondant à la clé de cache. Lorsque la durée de votre clé de cache expire, votre API transmet la demande au point de terminaison de votre jeton et met en cache un nouveau jeton d'accès.

Note

La durée de votre clé de cache doit être inférieure à la durée du jeton d'accès de votre client d'application.

La clé de cache est une combinaison des portées OAuth que vous demandez dans le paramètre d'URL scope et l'en-tête `Authorization` de la demande. L'en-tête `Authorization` contient l'identifiant et le secret de votre client d'application. Vous n'avez pas besoin de mettre en œuvre une logique supplémentaire dans votre application pour appliquer cette solution. Vous devez uniquement mettre à jour votre configuration pour modifier le chemin d'accès au point de terminaison du jeton de votre groupe d'utilisateurs.

Vous pouvez également implémenter la mise en cache des jetons avec [ElastiCache for Redis](#). Pour un contrôle précis avec les politiques AWS Identity and Access Management (IAM), envisagez un cache [Amazon DynamoDB](#).

 Note

La mise en cache dans API Gateway est soumise à des frais supplémentaires. [Consultez la tarification pour plus d'informations.](#)

Pour configurer un proxy de mise en cache avec API Gateway

1. Ouvrez la [console API Gateway](#) et créez une API REST.
2. Dans Ressources (Ressources), créez une méthode POST.
 - a. Choisissez le type d'intégration HTTP.
 - b. Sélectionnez Use HTTP proxy integration (Utiliser une intégration proxy HTTP).
 - c. Saisissez une URL de point de terminaison de `https://<your user pool domain>/oauth2/token`.
3. Dans Ressources (Ressources), configurez la clé de cache.
 - a. Modifiez la demande de méthode de votre méthode POST.
 - b. Définissez votre paramètre scope et votre en-tête Authorization comme clé de mise en cache.
 - i. Ajoutez une chaîne de requête dans URL query string parameters (Paramètres de chaîne de requête d'URL) et choisissez Caching (Mise en cache) pour la chaîne scope.
 - ii. Ajoutez un en-tête à HTTP request headers (En-têtes de demande HTTP) et choisissez Caching (Mise en cache) pour l'en-tête Authorization.
4. Dans Stages (Étapes), configurez la mise en cache.
 - a. Choisissez l'étape que vous souhaitez modifier.
 - b. Sous Settings (Paramètres), sélectionnez Enable API cache (Activer le cache d'API).
 - c. Choisissez une capacité de cache.
 - d. Choisissez un cache time-to-live (TTL) d'au moins 3 600 secondes.
 - e. Décochez la case Exiger une autorisation.
5. Dans Stages (Étapes), notez l'URL d'appel.
6. Mettez à jour votre application pour envoyer des demandes de jetons POST à l'URL d'appel de votre API plutôt qu'au point de terminaison /oauth2/token de votre groupe d'utilisateurs.

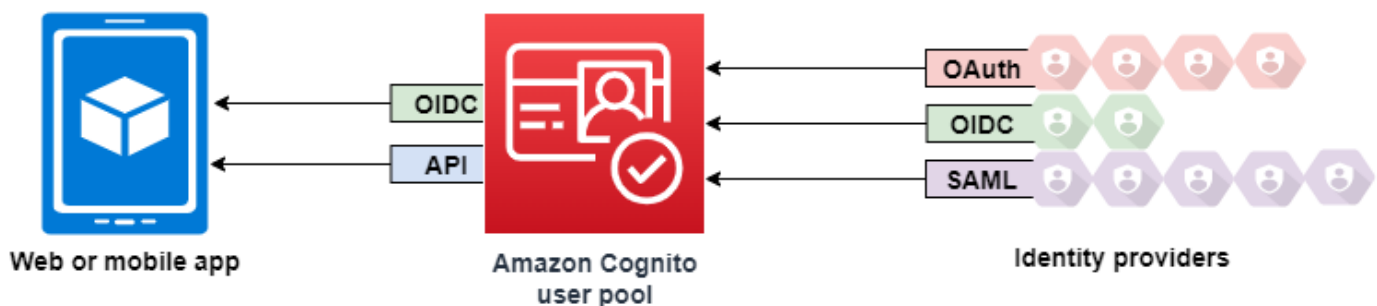
Accès aux ressources après avoir réussi l'authentification d'un groupe d'utilisateurs

Les utilisateurs de votre application peuvent soit se connecter directement via un groupe d'utilisateurs, soit fédérer via un fournisseur d'identité (IdP) tiers. Le groupe d'utilisateurs gère les frais généraux liés à la gestion des jetons renvoyés lors de la connexion aux réseaux sociaux via Facebook, Google, Amazon et Apple, et depuis OpenID Connect (OIDC) et SAML. IdPs Pour plus d'informations, consultez [Utilisation des jetons avec des groupes d'utilisateurs](#).

Après authentification, votre application recevra d'Amazon Cognito des jetons de groupe d'utilisateurs. Vous pouvez utiliser les jetons du groupe d'utilisateurs pour :

- Récupérez les AWS informations d'identification qui autorisent les demandes de ressources d'application dans Services AWS Amazon DynamoDB et Amazon S3, par exemple.
- Fournissez une preuve d'authentification temporaire et révocable.
- Renseignez les données d'identité d'un profil utilisateur dans votre application.
- Autorisez les modifications du profil de l'utilisateur connecté dans le répertoire du groupe d'utilisateurs.
- Autorisez les demandes d'informations sur les utilisateurs à l'aide d'un jeton d'accès.
- Autorisez les demandes adressées aux données qui se trouvent derrière des API externes protégées par des jetons d'accès.
- Autorisez l'accès aux ressources de l'application stockées sur le client ou le serveur avec des autorisations vérifiées par Amazon.

Pour plus d'informations, consultez [Flux d'authentification de groupe d'utilisateurs](#) et [Utilisation des jetons avec des groupes d'utilisateurs](#).



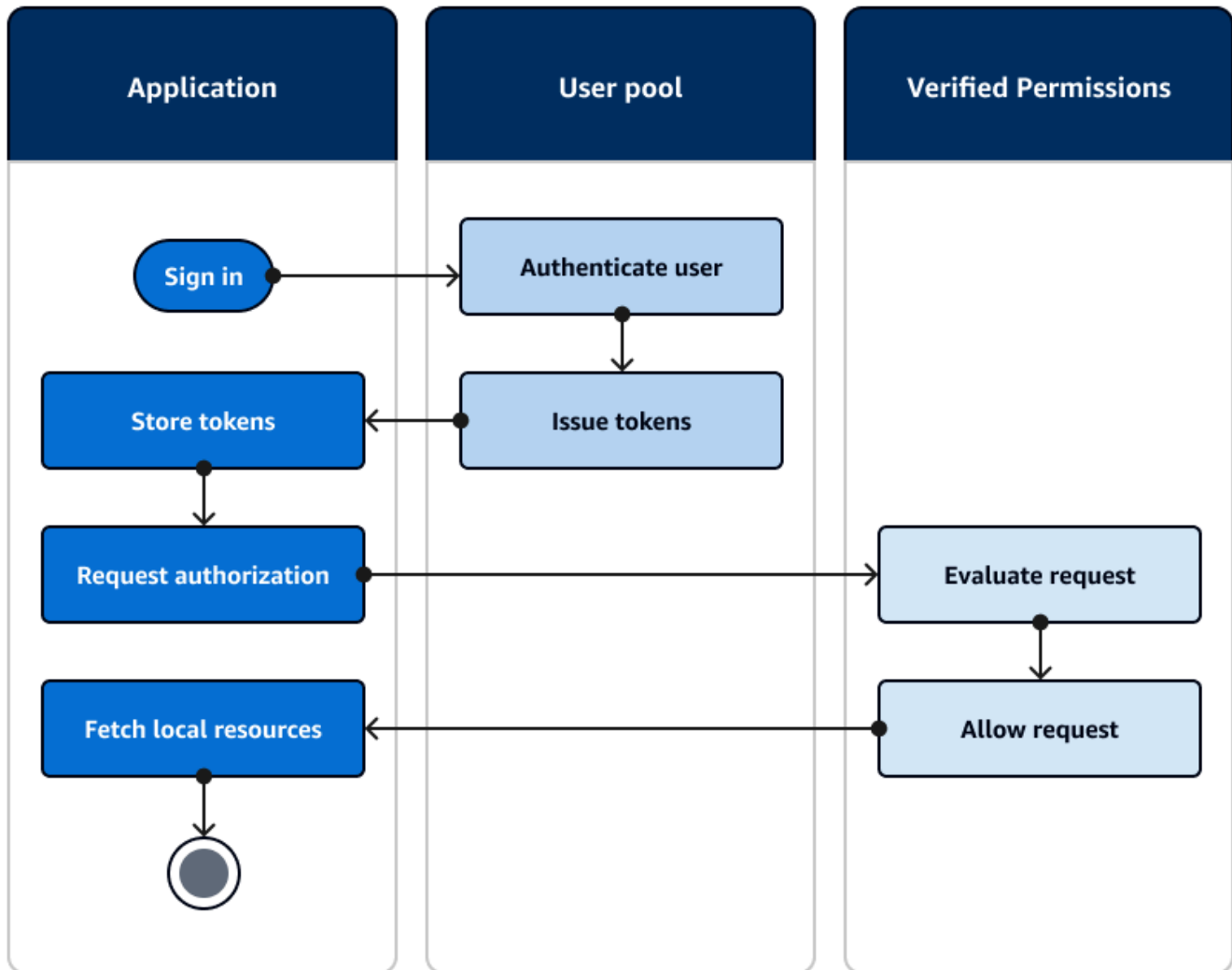
Rubriques

- [Autoriser l'accès aux ressources du client ou du serveur avec les autorisations Amazon Verified](#)
- [Accès aux ressources avec API Gateway après connexion](#)
- [Accès à Services AWS l'aide d'un pool d'identités après la connexion](#)

Autoriser l'accès aux ressources du client ou du serveur avec les autorisations Amazon Verified

Votre application peut transmettre les jetons d'un utilisateur connecté à [Amazon](#) Verified Permissions. Verified Permissions est un service de gestion des autorisations et d'autorisation évolutif et précis pour les applications personnalisées que vous avez créées. Un groupe d'utilisateurs Amazon Cognito peut être une source d'identité pour un magasin de politiques d'autorisations vérifiées. Verified Permissions prend les décisions d'autorisation pour les actions et les ressources demandées `premium_badge.png`, par exemple `GetPhoto` pour le principal et ses attributs dans les jetons du pool d'utilisateurs.

Le schéma suivant montre comment votre application peut transmettre le jeton d'un utilisateur à Verified Permissions dans le cadre d'une demande d'autorisation.



Commencez avec les autorisations vérifiées d'Amazon

Après avoir intégré les autorisations vérifiées à votre groupe d'utilisateurs, vous bénéficiez d'une source centrale d'autorisations granulaires pour toutes vos applications Amazon Cognito. Vous n'avez donc plus besoin d'une logique de sécurité précise que vous auriez autrement à coder et à répliquer entre toutes vos applications. Pour plus d'informations sur l'autorisation avec autorisations vérifiées, consultez [Autorisation avec Amazon Verified Permissions](#).

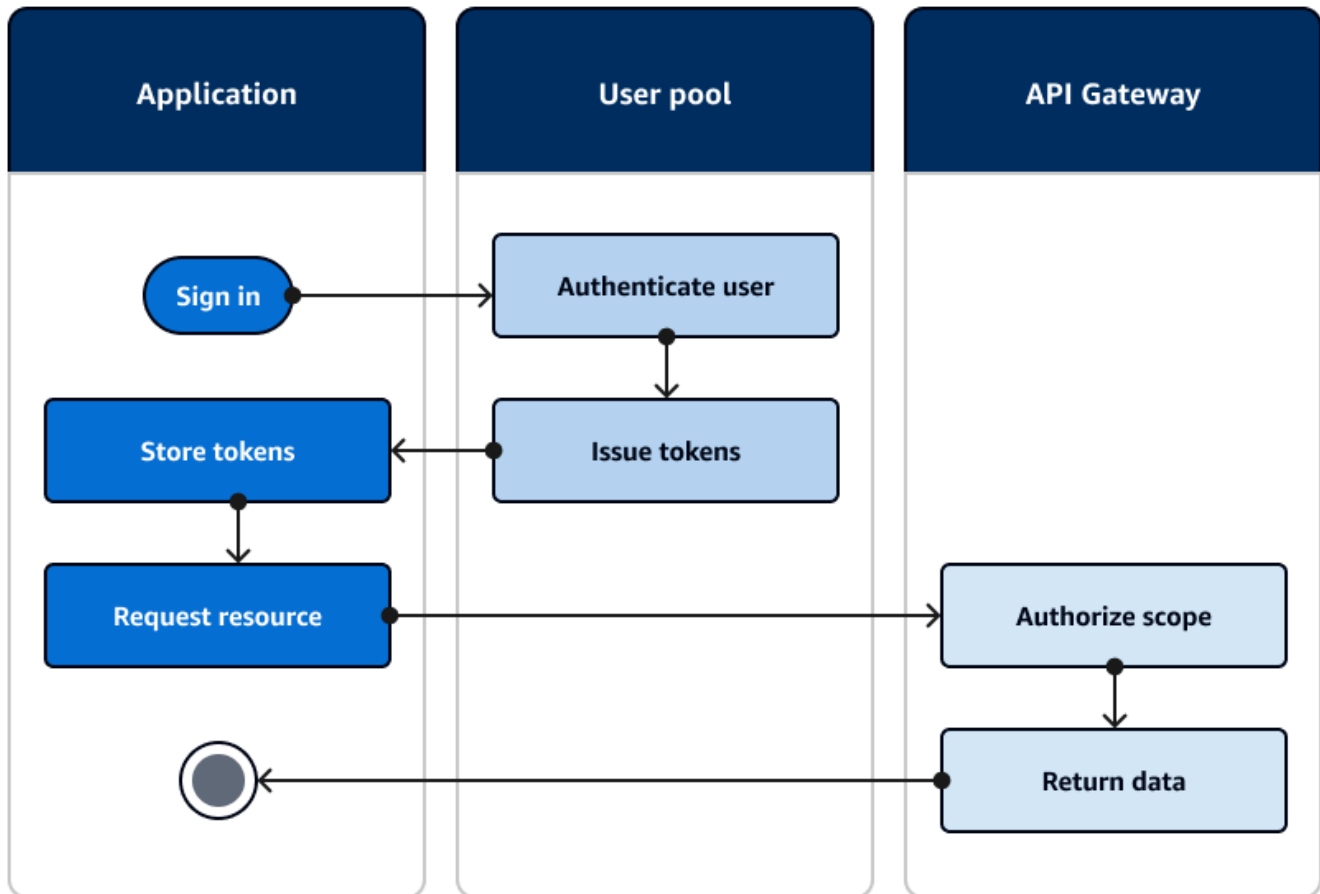
Les demandes d'autorisation d'autorisations vérifiées nécessitent des AWS informations d'identification. Vous pouvez mettre en œuvre certaines des techniques suivantes pour appliquer en toute sécurité des informations d'identification aux demandes d'autorisation.

- Utilisez une application Web capable de stocker des secrets dans le backend du serveur.
- Obtenez des informations d'identification authentifiées pour le pool d'identités.
- Proxy les demandes des utilisateurs par le biais d'une access-token-authorized API et ajout AWS d'informations d'identification à la demande.

Accès aux ressources avec API Gateway après connexion

Les jetons des groupes d'utilisateurs Amazon Cognito sont couramment utilisés pour autoriser les demandes adressées à une [API REST API Gateway](#). Les champs d'application OAuth 2.0 contenus dans les jetons d'accès peuvent autoriser une méthode et un chemin, comme pour. HTTP GET / app_assets Les jetons d'identification peuvent servir d'authentification générique à une API et peuvent transmettre des attributs utilisateur au service principal. API Gateway propose des options d'autorisation personnalisées supplémentaires, telles que les [autorisateurs JWT pour les API HTTP](#) et les [autorisateurs Lambda](#) qui peuvent appliquer une logique plus fine.

Le schéma suivant illustre une application qui accède à une API REST avec les étendues OAuth 2.0 dans un jeton d'accès.



Votre application doit collecter les jetons provenant de sessions authentifiées et les ajouter en tant que jetons porteurs à un `Authorization` en-tête de la demande. Configurez l'autorisateur que vous avez configuré pour l'API, le chemin et la méthode afin d'évaluer le contenu du jeton. API Gateway renvoie des données uniquement si la demande répond aux conditions que vous avez définies pour votre autorisateur.

L'API API Gateway peut éventuellement approuver l'accès à partir d'une application de différentes manières :

- Le jeton d'accès contient la portée OAuth 2.0 correcte. L'[autorisateur de groupes d'utilisateurs Amazon Cognito pour une API REST](#) est une implémentation courante avec un faible obstacle à l'entrée. Vous pouvez également évaluer le corps, les paramètres de la chaîne de requête et les en-têtes d'une demande adressée à ce type d'autorisateur.

- Le jeton d'identification est valide et n'a pas expiré. Lorsque vous transmettez un jeton d'identification à un autorisateur Amazon Cognito, vous pouvez effectuer une validation supplémentaire du contenu du jeton d'identification sur votre serveur d'applications.
- Un groupe, une réclamation, un attribut ou un rôle dans un jeton d'accès ou d'identification répond aux exigences que vous définissez dans une fonction Lambda. Un [autorisateur Lambda](#) analyse le jeton dans l'en-tête de la demande et l'évalue pour une décision d'autorisation. Vous pouvez créer une logique personnalisée dans votre fonction ou envoyer une demande d'API à [Amazon Verified Permissions](#).

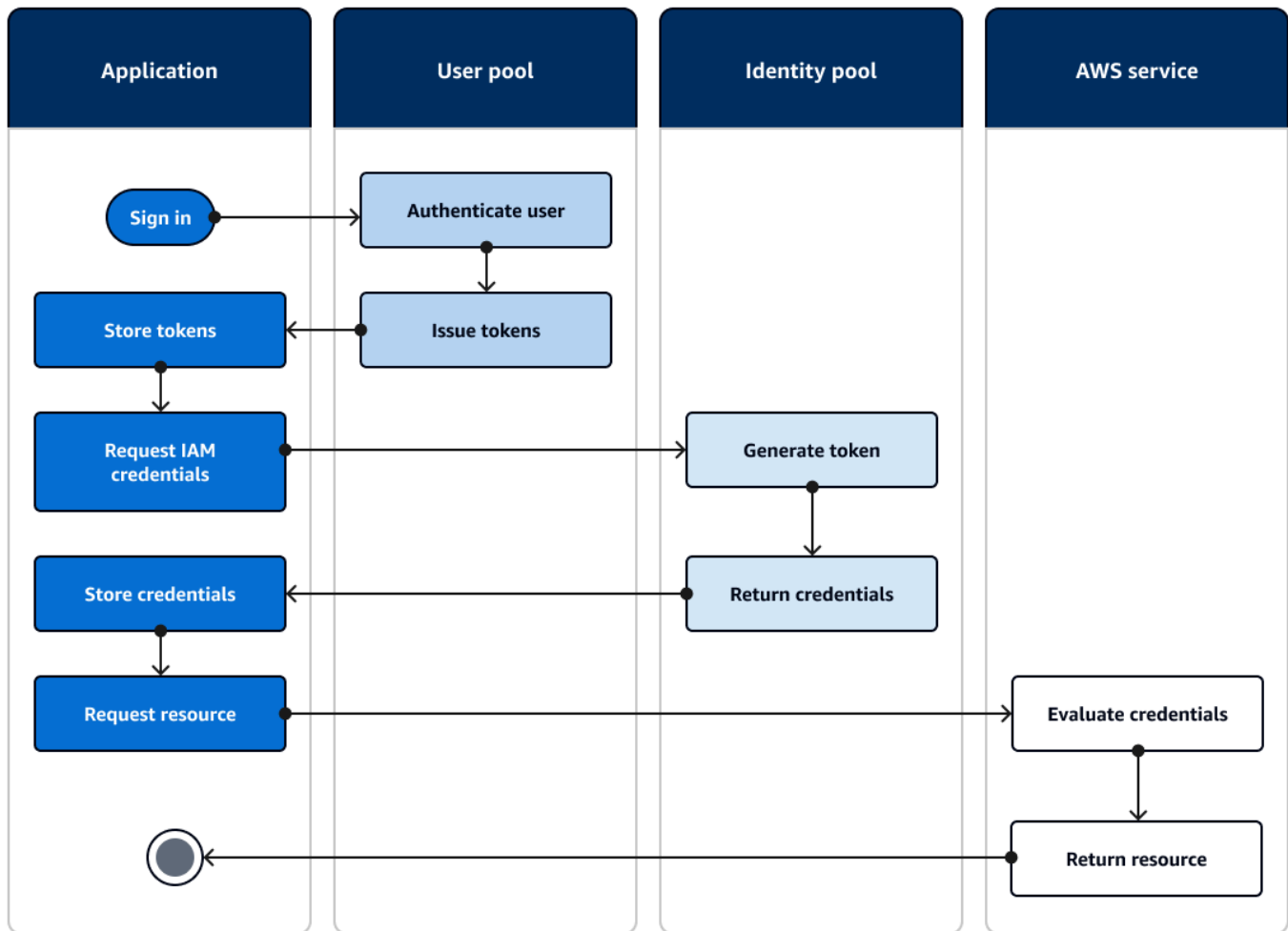
Vous pouvez également autoriser les requêtes adressées à une [API AWS AppSync GraphQL](#) à l'aide de jetons provenant d'un groupe d'utilisateurs.

Accès à Services AWS l'aide d'un pool d'identités après la connexion

Une fois que vos utilisateurs se sont connectés à un groupe d'utilisateurs, ils peuvent y accéder à l'API Services AWS aide d'informations d'identification API temporaires émises par un groupe d'identités.

Votre application Web ou mobile reçoit des jetons d'un groupe d'utilisateurs. Lorsque vous configurez votre groupe d'utilisateurs en tant que fournisseur d'identité pour votre groupe d'identités, le groupe d'identités échange des jetons contre des AWS informations d'identification temporaires. Ces informations d'identification peuvent être étendues aux rôles IAM et à leurs politiques qui permettent aux utilisateurs d'accéder à un ensemble limité de AWS ressources. Pour plus d'informations, consultez [Flux d'authentification de groupes d'identités \(identités fédérées\)](#).

Le schéma suivant montre comment une application se connecte à un groupe d'utilisateurs, récupère les informations d'identification du pool d'identités et demande un actif à un Service AWS.



Vous pouvez utiliser les informations d'identification du pool d'identités pour :

- Soumettez des demandes d'autorisation détaillées à Amazon Verified Permissions avec les informations d'identification de votre utilisateur.
- Connectez-vous à une API REST Amazon API Gateway ou à une API AWS AppSync GraphQL qui autorise les connexions avec IAM.
- Connectez-vous à un backend de base de données tel qu'Amazon DynamoDB ou Amazon RDS qui autorise les connexions avec IAM.
- Récupérez les ressources de l'application depuis un compartiment Amazon S3.
- Lancez une session avec un bureau WorkSpaces virtuel Amazon.

Les groupes d'identités ne fonctionnent pas exclusivement au sein d'une session authentifiée avec un groupe d'utilisateurs. Ils acceptent également l'authentification directement auprès de fournisseurs

d'identité tiers et peuvent générer des informations d'identification pour les utilisateurs invités non authentifiés.

Pour plus d'informations sur l'utilisation de groupes d'identités avec des groupes de groupes d'utilisateurs pour contrôler l'accès à vos AWS ressources, consultez [Ajout de groupes à un groupe d'utilisateurs](#) et [Utilisation du contrôle d'accès basé sur les rôles](#). En outre, pour plus d'informations sur les pools d'identités et AWS Identity and Access Management, voir [Concepts de groupes d'identités](#).

Configuration d'un groupe d'utilisateurs à l'aide du AWS Management Console

Créez un groupe d'utilisateurs Amazon Cognito et prenez note des identifiants indiqués dans les champs User Pool ID (ID du groupe d'utilisateurs) et App Client ID (ID du client d'application) pour chacune de vos applications client. Pour plus d'informations sur la création de groupes d'utilisateurs, consultez [Démarrage avec les groupes d'utilisateurs](#).

Configuration d'un pool d'identités à l'aide du AWS Management Console

La procédure suivante décrit comment utiliser le AWS Management Console pour intégrer un pool d'identités à un ou plusieurs groupes d'utilisateurs et applications clientes.

Pour ajouter un fournisseur d'identité (IdP) aux groupes d'utilisateurs Amazon Cognito

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Groupe d'utilisateurs Amazon Cognito.
5. Saisissez un ID de groupe d'utilisateurs et un ID de client d'application.
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - a. Vous pouvez attribuer aux utilisateurs de cet IdP le rôle par défaut que vous avez défini lorsque vous avez configuré votre rôle authentifié, ou vous pouvez choisir un rôle avec des règles. Avec un fournisseur d'identité de groupe d'utilisateurs Amazon Cognito, vous pouvez également sélectionner Choisir le rôle avec l'enregistrement `preferred_role` dans les jetons. Pour plus d'informations sur le champ standard `cognito:preferred_role`, consultez [Affectation de valeurs de priorité à des groupes](#).

- i. Si vous avez choisi Choisir un rôle avec des règles, entrez la réclamation source issue de l'authentification de votre utilisateur, l'opérateur que vous souhaitez utiliser pour comparer la réclamation à la règle, la valeur qui provoquera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer lorsque l'attribution du rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Si vous avez choisi Choose role with preferred_role claim in tokens, Amazon Cognito émet des informations d'identification pour le rôle indiqué dans la réclamation de votre utilisateur. `cognito:preferred_role` Si aucun enregistrement de rôle préféré n'est présent, Amazon Cognito émet des informations d'identification en fonction de la Résolution du rôle.
 - b. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
- Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
8. Sélectionnez Enregistrer les modifications.

Intégration d'un groupe d'utilisateurs à un groupe d'identités

Une fois que l'utilisateur de l'application est authentifié, ajoutez son jeton d'identité à la carte de connexions dans le fournisseur d'informations d'identification. Le nom de ce fournisseur dépend de l'ID du groupe d'utilisateurs Amazon Cognito. Sa structure est la suivante :

```
cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>
```

Vous pouvez déduire la valeur <region> de l'ID du groupe d'utilisateurs. Par exemple, si l'ID du groupe d'utilisateurs est `estus-east-1_EXAMPLE1`, c'<region>est le casus-`east-1`. Si l'ID du groupe d'utilisateurs est `estus-west-2_EXAMPLE2`, c'<region>est le casus-`west-2`.

JavaScript

```
var cognitoUser = userPool.getCurrentUser();

if (cognitoUser != null) {
  cognitoUser.getSession(function(err, result) {
    if (result) {
      console.log('You are now logged in.');
```

```

      // Add the User's Id Token to the Cognito credentials login map.
      AWS.config.credentials = new AWS.CognitoIdentityCredentials({
        IdentityPoolId: 'YOUR_IDENTITY_POOL_ID',
        Logins: {
          'cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>':
result.getIdToken().getJwtToken()
        }
      });
    }
  });
}
```

Android

```
cognitoUser.getSessionInBackground(new AuthenticationHandler() {
  @Override
  public void onSuccess(CognitoUserSession session) {
    String idToken = session.getIdToken().getJWTToken();

    Map<String, String> logins = new HashMap<String, String>();
    logins.put("cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>",
session.getIdToken().getJWTToken());
    credentialsProvider.setLogins(logins);
  }
});
```


iOS - objective-C

```
AWSServiceConfiguration *serviceConfiguration = [[AWSServiceConfiguration alloc]
initWithRegion:AWSRegionUSEast1 credentialsProvider:nil];
AWSCognitoIdentityUserPoolConfiguration *userPoolConfiguration =
[[AWSCognitoIdentityUserPoolConfiguration alloc] initWithClientId:@"YOUR_CLIENT_ID"
clientSecret:@"YOUR_CLIENT_SECRET" poolId:@"YOUR_USER_POOL_ID"];
[AWSCognitoIdentityUserPool
registerCognitoIdentityUserPoolWithConfiguration:serviceConfiguration
userPoolConfiguration:userPoolConfiguration forKey:@"UserPool"];
AWSCognitoIdentityUserPool *pool = [AWSCognitoIdentityUserPool
CognitoIdentityUserPoolForKey:@"UserPool"];
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider
alloc] initWithRegionType:AWSRegionUSEast1 identityPoolId:@"YOUR_IDENTITY_POOL_ID"
identityProviderManager:pool];
```

iOS - swift

```
let serviceConfiguration = AWSServiceConfiguration(region: .USEast1,
credentialsProvider: nil)
let userPoolConfiguration = AWSCognitoIdentityUserPoolConfiguration(clientId:
"YOUR_CLIENT_ID", clientSecret: "YOUR_CLIENT_SECRET", poolId: "YOUR_USER_POOL_ID")
AWSCognitoIdentityUserPool.registerCognitoIdentityUserPoolWithConfiguration(serviceConfiguration,
userPoolConfiguration: userPoolConfiguration, forKey: "UserPool")
let pool = AWSCognitoIdentityUserPool(forKey: "UserPool")
let credentialsProvider = AWSCognitoCredentialsProvider(regionType: .USEast1,
identityPoolId: "YOUR_IDENTITY_POOL_ID", identityProviderManager:pool)
```

Utiliser les fonctions de sécurité des groupes d'utilisateurs Amazon Cognito

Vous pouvez ajouter l'authentification multi-facteur (MFA) à un groupe d'utilisateurs pour protéger l'identité de vos utilisateurs. La MFA ajoute un second facteur d'authentification pour que votre groupe d'utilisateurs ne s'appuie pas uniquement sur le nom d'utilisateur et le mot de passe. Vous pouvez utiliser les SMS ou les mots de passe uniques à durée limitée (TOTP) comme deuxième facteur pour connecter vos utilisateurs. Vous pouvez également utiliser l'authentification adaptative avec son modèle basé sur le risque pour prévoir quand vous pouvez avoir besoin d'un autre facteur d'authentification. Les fonctionnalités de sécurité avancée de groupe d'utilisateurs incluent l'authentification adaptative et les protections contre les informations d'identification compromises.

Rubriques

- [Ajout de l'authentification MFA à un groupe d'utilisateurs](#)
- [Ajout de la sécurité avancée à un groupe d'utilisateurs](#)
- [Associer une ACL AWS WAF Web à un groupe d'utilisateurs](#)
- [Sensibilité à la casse du groupe d'utilisateurs](#)
- [Protection contre la suppression du groupe d'utilisateurs](#)
- [Gestion des réponses aux erreurs liées à l'existence des utilisateurs](#)

Ajout de l'authentification MFA à un groupe d'utilisateurs

L'authentification multifacteur (MFA) renforce la sécurité pour votre application. Elle ajoute un facteur d'authentification correspondant à quelque chose que vous avez au facteur correspondant à quelque chose que vous savez de nom d'utilisateur et de mot de passe. Vous pouvez choisir les SMS ou les mots de passe uniques à durée limitée (TOTP) comme second facteur pour connecter vos utilisateurs.

Note

La première fois qu'un nouvel utilisateur se connecte à votre application, Amazon Cognito émet des jetons OAuth 2.0, même si votre groupe d'utilisateurs nécessite l'authentification MFA. Le deuxième facteur d'authentification, lorsque votre utilisateur se connecte pour la première fois, est sa confirmation du message de vérification qu'Amazon Cognito lui envoie. Si votre groupe d'utilisateurs nécessite l'authentification MFA, Amazon Cognito invite votre utilisateur à enregistrer un facteur de connexion supplémentaire à utiliser lors de chaque tentative de connexion après la première.

Avec l'authentification adaptative, vous pouvez configurer votre groupe d'utilisateurs pour exiger un deuxième facteur d'authentification en réponse à un niveau de risque accru. Pour ajouter l'authentification adaptative à votre groupe d'utilisateurs, consultez [Ajout de la sécurité avancée à un groupe d'utilisateurs](#).

Quand vous définissez l'authentification MFA sur `required` pour un groupe d'utilisateurs, tous les utilisateurs doivent l'utiliser pour se connecter. Pour se connecter, chaque utilisateur a besoin d'au moins un facteur d'authentification MFA, tel que les SMS ou les mots de passe TOTP. Lorsque

vous définissez l'authentification MFA sur `required`, vous devez inclure la configuration MFA dans l'intégration des utilisateurs afin que votre groupe d'utilisateurs leur permette de se connecter.

Si vous activez les SMS comme facteur d'authentification MFA, vous pouvez exiger que les utilisateurs fournissent des numéros de téléphone et les vérifient dans le cadre de l'inscription. Si la MFA est définie sur `required` et ne prend en charge que les SMS comme facteur, les utilisateurs doivent fournir des numéros de téléphone. Les utilisateurs dépourvus de numéro de téléphone ont besoin de votre assistance pour ajouter un numéro de téléphone à leur profil avant de pouvoir se connecter. Vous pouvez utiliser des numéros de téléphone non vérifiés pour la MFA par SMS. Ces numéros recevront un statut vérifié une fois l'authentification MFA réussie.

Si vous avez rendu obligatoire l'authentification MFA et que vous avez activé les SMS et les mots de passe TOTP comme méthodes de vérification prises en charge, Amazon Cognito invite les nouveaux utilisateurs sans numéro de téléphone à configurer l'authentification MFA par TOTP. Si vous avez rendu obligatoire l'authentification MFA et que la seule méthode MFA que vous avez activée est TOTP, Amazon Cognito invite tous les nouveaux utilisateurs à configurer l'authentification MFA par TOTP la deuxième fois qu'ils se connectent. Amazon Cognito pose le défi de configurer le TOTP MFA en réponse [InitiateAuth](#) à des opérations d'API. [AdminInitiateAuth](#)

L'interface utilisateur hébergée invite les utilisateurs à configurer l'authentification MFA lorsque vous rendez obligatoire l'authentification MFA. Lorsque vous rendez facultative l'authentification MFA dans votre groupe d'utilisateurs, l'interface utilisateur hébergée n'invite pas les utilisateurs. Pour utiliser une authentification MFA facultative, vous devez créer une interface dans votre application qui invite vos utilisateurs à choisir de configurer l'authentification MFA, puis qui les guide via les entrées d'API pour vérifier leur facteur de connexion supplémentaire.

Après cinq tentatives infructueuses de présentation d'un code MFA, Amazon Cognito lance le processus de verrouillage par temporisation exponentielle décrit dans le [Flux d'authentification de groupe d'utilisateurs](#).

Rubriques

- [Prérequis](#)
- [Configuration de l'authentification multifacteur](#)
- [MFA par SMS](#)
- [Authentification MFA par jeton logiciel TOTP](#)

Prérequis

Avant de configurer MFA, prenez en compte les éléments suivants :

- Lorsque vous activez l'authentification MFA dans votre groupe d'utilisateurs et choisissez SMS text message (SMS) comme second facteur, vous pouvez envoyer des SMS à un attribut de numéro de téléphone que vous n'avez pas vérifié dans Amazon Cognito. Une fois que votre utilisateur a terminé l'authentification MFA par SMS, Amazon Cognito définit son attribut `phone_number_verified` sur `true`.
- Si votre compte se trouve dans le sandbox SMS Région AWS qui contient les ressources Amazon Simple Notification Service (Amazon SNS) pour votre groupe d'utilisateurs, vous devez vérifier les numéros de téléphone dans Amazon SNS avant de pouvoir envoyer un SMS. Pour plus d'informations, consultez [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#).
- Les fonctions de sécurité avancées exigent que vous activiez l'authentification MFA et la définissiez comme facultative dans la console du groupe d'utilisateurs Amazon Cognito. Pour plus d'informations, consultez [Ajout de la sécurité avancée à un groupe d'utilisateurs](#).

Configuration de l'authentification multifacteur

Vous pouvez configurer la MFA dans la console Amazon Cognito.

Pour configurer l'authentification MFA dans la console Amazon Cognito.

1. Connectez-vous à la [console Amazon Cognito](#).
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet Sign-in experience (Expérience de connexion). retrouvez Authentification multifacteur et choisissez Modifier.
5. Choisissez la méthode MFA enforcement que vous souhaitez utiliser avec votre groupe d'utilisateurs.

Edit multi-factor authentication (MFA) [Info](#)

Amazon Cognito provides your app users with additional authentication factors using SMS messages and time-based one-time passwords (TOTP).

Multi-factor authentication

Configure secure access to your app by enforcing multi-factor authentication (MFA) during the user sign-in process. MFA settings are applied to all app clients.

MFA enforcement [Info](#)

Require MFA -

Recommended

Users must provide an additional authentication factor when signing in.

Optional MFA

Users can sign in with a single authentication factor, and can choose to add additional authentication factors.

No MFA

Users can only sign in with a single authentication factor. This is the least secure option.


MFA methods [Info](#)

Choose the MFA methods that are allowed in your user pool. TOTP-based MFA offers a higher level of security. Recipient message and data rates apply.

Authenticator apps

Users can authenticate with a TOTP from an authenticator app such as Authy or Google Authenticator.

SMS message

Users can authenticate with a code sent by SMS message to a verified phone number. SMS messages are charged separately by Amazon SNS. [Learn more about pricing](#)  This option must be selected because SMS is configured.

Cancel

Save changes

- a. **Require MFA (Demander l'authentification MFA).** Tous les utilisateurs de votre groupe d'utilisateurs doivent se connecter avec un code SMS ou un mot de passe unique à durée limitée (TOTP) supplémentaire.
 - b. **Optional MFA (Authentification MFA facultative)** – Vous pouvez donner à vos utilisateurs la possibilité d'enregistrer un facteur de connexion supplémentaire tout en autorisant la connexion par les utilisateurs qui n'ont pas configuré MFA. Choisissez cette option si vous utilisez l'authentification adaptative. Pour plus d'informations sur l'authentification adaptative, consultez [Ajout de la sécurité avancée à un groupe d'utilisateurs](#).
 - c. **No MFA (Aucune authentification MFA).** Vos utilisateurs ne peuvent pas enregistrer un facteur de connexion supplémentaire.
6. Choisissez les méthodes MFA que vous allez prendre en charge dans votre application. Vous pouvez définir SMS message (SMS) ou Authenticator apps (Applications d'authentification) avec des mots de passe TOTP comme deuxième facteur. Nous vous recommandons d'implémenter

une authentification MFA par mot de passe unique à durée déterminée afin que la récupération de compte puisse utiliser les SMS.

7. Si vous utilisez les SMS comme deuxième facteur et que vous n'avez pas configuré de rôle IAM à utiliser avec Amazon Simple Notification Service (Amazon SNS) pour les SMS, créez-en un dans la console. Dans l'onglet Messaging (Messagerie) pour votre groupe d'utilisateurs, localisez SMS et choisissez Edit (Modifier). Vous pouvez également utiliser un rôle existant permettant à Amazon Cognito d'envoyer des SMS à vos utilisateurs à votre place. Pour en savoir plus, consultez [Rôles IAM](#).
8. Sélectionnez Enregistrer les modifications.

MFA par SMS

Quand un utilisateur se connecte avec la MFA activée, il commence par saisir et envoyer son nom d'utilisateur et son mot de passe. L'appli cliente reçoit une réponse `getMFA` indiquant où le code d'autorisation a été envoyé. L'appli cliente doit indiquer à l'utilisateur où rechercher le code (par exemple, le numéro de téléphone auquel le code a été envoyé). Ensuite, elle fournit un formulaire pour entrer le code. Enfin, l'appli cliente envoie le code pour achever le processus de connexion. La destination étant masquée, tous les chiffres du numéro de téléphone sont cachés, à l'exception des quatre derniers. Si une application utilise l'interface utilisateur hébergée Amazon Cognito, elle affiche une page permettant à l'utilisateur de saisir le code MFA.

Le code d'autorisation des SMS est valide pour la durée d'une session de flux d'authentification que vous avez définie pour votre client d'application.

Définissez la durée d'une session de flux d'authentification dans la console Amazon Cognito à partir de l'onglet App integration (Intégration d'applications) lorsque vous modifiez votre client d'application sous App clients and analytics (Clients d'application et analyses). Vous pouvez également définir la durée d'une session de flux d'authentification dans une demande d'API `CreateUserPoolClient` ou `UpdateUserPoolClient`. Pour plus d'informations, consultez [Flux d'authentification de groupe d'utilisateurs](#).

Si un utilisateur n'a plus accès à son appareil auquel sont envoyés les codes MFA par SMS, il doit contacter le service client pour obtenir de l'aide. Un administrateur disposant des Comptes AWS autorisations nécessaires peut modifier le numéro de téléphone de l'utilisateur, mais uniquement via l'API AWS CLI ou l'API.

Lorsqu'un utilisateur sort du flux MFA par SMS avec succès, son numéro de téléphone est également marqué comme vérifié.

Note

Le SMS pour MFA est facturé séparément. (L'envoi des codes de vérification aux adresses e-mail est réalisé sans frais.) Pour plus d'informations sur la tarification Amazon SNS, consultez [Tarifs SMS internationaux](#). Pour obtenir la liste des pays où la messagerie SMS est disponible, consultez la page [Régions et pays pris en charge](#).

Important

Pour vous assurer que des SMS sont envoyés pour vérifier les numéros de téléphone et pour la MFA par SMS, vous devez demander à Amazon SNS une augmentation de la limite des dépenses.

Amazon Cognito utilise Amazon SNS pour l'envoi de SMS aux utilisateurs. Le nombre de SMS que livre Amazon SNS est soumis à des limites de dépenses. Les limites de dépenses peuvent être spécifiées pour un AWS compte et pour des messages individuels, et les limites s'appliquent uniquement au coût d'envoi de SMS.

La valeur de limite de dépenses par défaut (si aucune valeur n'est précisée) est de 1,00 USD par mois. Si vous souhaitez augmenter la limite, soumettez un [dossier d'augmentation de limite SNS](#) au AWS Support Centre. Dans New limit value (Valeur de la nouvelle limite), saisissez la limite de dépenses mensuelles que vous souhaitez. Dans le champ Use Case Description (Description du cas d'utilisation), expliquez que vous demandez une augmentation de la limite de dépenses mensuelles SMS.

Pour ajouter l'authentification MFA à votre groupe d'utilisateurs, consultez [Ajout de l'authentification MFA à un groupe d'utilisateurs](#). Pour plus d'informations sur les SMS contenant Amazon SNS dans votre groupe d'utilisateurs, consultez [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#)

Authentification MFA par jeton logiciel TOTP

Lorsque vous configurez la MFA par jeton logiciel TOTP dans votre groupe d'utilisateurs, votre utilisateur se connecte avec un nom d'utilisateur et un mot de passe, puis utilise un mot de passe TOTP pour terminer l'authentification. Une fois que votre utilisateur a défini et vérifié un nom d'utilisateur et un mot de passe, il peut activer un jeton logiciel TOTP pour la MFA. Si votre application utilise l'interface utilisateur hébergée d'Amazon Cognito pour la connexion des

utilisateurs, l'utilisateur envoie son nom d'utilisateur et son mot de passe, puis envoie le mot de passe TOTP sur une page de connexion supplémentaire.

Vous pouvez activer l'authentification MFA par TOTP pour votre groupe d'utilisateurs dans la console Amazon Cognito. Vous pouvez également utiliser les opérations d'API Amazon Cognito. Au niveau du groupe d'utilisateurs, vous pouvez appeler [SetUserPoolMfaConfig](#) pour configurer le MFA et activer le MFA TOTP.

Note

Si vous n'avez pas activé l'authentification MFA par jeton logiciel TOTP pour le groupe d'utilisateurs, Amazon Cognito ne peut pas utiliser le jeton pour associer ou vérifier les utilisateurs. Dans ce cas, les utilisateurs reçoivent une exception `SoftwareTokenMFANotFoundException` avec la description `Software Token MFA has not been enabled by the userPool`. Si vous désactivez ultérieurement l'authentification MFA par jeton logiciel pour le groupe d'utilisateurs, les utilisateurs qui ont précédemment associé et vérifié un jeton TOTP peuvent continuer à l'utiliser pour l'authentification MFA.

La configuration du mot de passe TOTP pour votre utilisateur est un processus en plusieurs étapes au cours duquel l'utilisateur reçoit un code secret qu'il valide en entrant un mot de passe à usage unique. Ensuite, vous pouvez activer l'authentification MFA par TOTP pour l'utilisateur ou définir TOTP comme méthode MFA préférée pour l'utilisateur.

Lorsque vous configurez votre groupe d'utilisateurs de manière à exiger une connexion avec authentification MFA par TOTP et que vos utilisateurs s'inscrivent à votre application dans l'interface utilisateur hébergée, Amazon Cognito automatise le processus utilisateur. Amazon Cognito invite votre utilisateur à choisir une méthode MFA, affiche un code QR pour configurer son application d'authentification et vérifie son enregistrement MFA. Dans les groupes d'utilisateurs auxquels vous avez permis de choisir entre une authentification par SMS et une authentification MFA par TOTP, Amazon Cognito propose également à vos utilisateurs de choisir une méthode. Pour plus d'informations sur l'expérience d'inscription à l'interface utilisateur hébergée, veuillez consulter [Comment créer un nouveau compte dans l'interface utilisateur hébergée par Amazon Cognito](#).

Important

Lorsqu'une ACL AWS WAF Web est associée à un groupe d'utilisateurs et qu'une règle de votre ACL Web présente un CAPTCHA, cela peut provoquer une erreur irrécupérable lors de

l'enregistrement TOTP de l'interface utilisateur hébergée. Pour créer une règle qui comporte une action CAPTCHA et qui n'affecte pas l'enregistrement TOTP de l'interface utilisateur hébergée, veuillez consulter [Configuration de votre ACL AWS WAF Web pour l'interface utilisateur hébergée \(TOTP MFA\)](#). Pour plus d'informations sur les ACL AWS WAF Web et Amazon Cognito, consultez [Associer une ACL AWS WAF Web à un groupe d'utilisateurs](#)

Pour mettre en œuvre une authentification MFA TOTP dans une interface utilisateur personnalisée dans laquelle vous utilisez l'[API Amazon Cognito](#), veuillez consulter [Configuration de la MFA pour un utilisateur dans l'API de groupes d'utilisateurs Amazon Cognito](#).

Pour ajouter l'authentification MFA à votre groupe d'utilisateurs, consultez [Ajout de l'authentification MFA à un groupe d'utilisateurs](#).

Considérations et restrictions liées à l'authentification MFA par TOTP

1. Amazon Cognito prend en charge l'authentification MFA par jeton logiciel via une application d'authentification qui génère des codes TOTP. Amazon Cognito ne prend pas en charge l'authentification MFA matérielle.
2. Lorsque votre groupe d'utilisateurs nécessite un mot de passe TOTP pour un utilisateur qui ne l'a pas configuré, l'utilisateur reçoit un jeton d'accès unique que votre application peut utiliser pour activer l'authentification MFA par TOTP pour l'utilisateur. Les tentatives de connexion suivantes échouent tant que l'utilisateur n'a pas enregistré un facteur de connexion TOTP supplémentaire.
 - Un utilisateur qui s'inscrit dans votre groupe d'utilisateurs avec l'opération d'API `SignUp` ou via l'interface utilisateur hébergée reçoit des jetons uniques lorsque l'utilisateur termine son inscription.
 - Une fois que vous avez créé un utilisateur et que l'utilisateur a défini son mot de passe initial, Amazon Cognito émet des jetons uniques à partir de l'interface utilisateur hébergée vers l'utilisateur. Si vous définissez un mot de passe permanent pour l'utilisateur, Amazon Cognito émet des jetons uniques lorsque l'utilisateur se connecte pour la première fois.
 - Amazon Cognito n'émet pas de jetons à usage unique à un utilisateur créé par un administrateur qui se connecte via les opérations de l'API or. [InitiateAuthAdminInitiateAuth](#) Une fois que votre utilisateur a réussi à définir son mot de passe initial ou si vous avez défini un mot de passe permanent pour l'utilisateur, Amazon Cognito met immédiatement l'utilisateur au défi de configurer l'authentification MFA.
3. Si un utilisateur d'un groupe d'utilisateurs nécessitant l'authentification MFA a déjà reçu un jeton d'accès unique mais n'a pas configuré l'authentification MFA par TOTP, l'utilisateur ne peut pas se

connecter avec l'interface utilisateur hébergée tant qu'il n'a pas configuré l'authentification MFA. Au lieu du jeton d'accès, vous pouvez utiliser la valeur de session réponse d'un MFA_SETUP défini envoyé à une [AssociateSoftwareToken](#) demande [InitiateAuth](#) ou contenue [AdminInitiateAuth](#) dans celle-ci.

4. Si vos utilisateurs ont configuré un mot de passe TOTP, ils peuvent l'utiliser pour l'authentification MFA même si vous désactivez le mot de passe TOTP pour le groupe d'utilisateurs par la suite.
5. Amazon Cognito accepte uniquement les TOTP provenant d'applications d'authentification qui génèrent les codes avec la fonction de hachage SHA-1. Les codes générés avec le hachage SHA-256 renvoient une erreur. Code mismatch

Configuration de la MFA pour un utilisateur dans l'API de groupes d'utilisateurs Amazon Cognito

Lorsqu'un utilisateur se connecte pour la première fois, votre application utilise son jeton d'accès unique pour générer la clé privée TOTP et la présenter à l'utilisateur au format texte ou sous la forme d'un code QR. L'utilisateur configure son application d'authentification et fournit un mot de passe TOTP pour les tentatives de connexion suivantes. Votre application ou l'interface utilisateur hébergée présente le mot de passe TOTP à Amazon Cognito dans les réponses aux demandes d'authentification MFA.

Rubriques

- [Associer le jeton logiciel TOTP](#)
- [Vérifier le jeton TOTP](#)
- [Connexion avec l'authentification MFA par TOTP](#)
- [Supprimer le jeton TOTP](#)

Associer le jeton logiciel TOTP

Pour associer le jeton TOTP, envoyez à l'utilisateur un code secret qu'il doit valider avec un mot de passe à usage unique. L'association du jeton nécessite trois étapes.

1. Lorsque votre utilisateur choisit le jeton logiciel TOTP MFA, [AssociateSoftwareToken](#) appelez pour renvoyer un code secret partagé unique généré pour le compte utilisateur. Vous pouvez autoriser à l'AssociateSoftwareToken aide d'un jeton d'accès ou d'une chaîne de session.
2. Votre application présente à l'utilisateur la clé privée ou un code QR généré à partir de la clé privée. Votre utilisateur doit saisir la clé dans une application de génération de mot de passe TOTP, telle que Google Authenticator. Vous pouvez utiliser [libqrencode](#) pour générer un code QR.

3. Votre utilisateur saisit la clé ou scanne le code QR dans une application d'authentification telle que Google Authenticator, et l'application commence à générer des codes.

Vérifier le jeton TOTP

Ensuite, vérifiez le jeton TOTP. Demandez des exemples de codes à votre utilisateur et fournissez-les au service Amazon Cognito pour confirmer que l'utilisateur parvient à générer des codes TOTP, comme suit.

1. Votre application demande à votre utilisateur de fournir un code pour démontrer qu'il a correctement configuré son application d'authentification.
2. L'application d'authentification de l'utilisateur affiche un mot de passe temporaire. L'application d'authentification base le mot de passe sur la clé secrète que vous avez donnée à l'utilisateur.
3. Votre utilisateur saisit son mot de passe temporaire. Votre application transmet le mot de passe temporaire à Amazon Cognito dans une demande d'API [VerifySoftwareToken](#).
4. Amazon Cognito a conservé la clé secrète associée à l'utilisateur, génère un mot de passe TOTP et le compare à celui fourni par votre utilisateur. S'ils correspondent, `VerifySoftwareToken` renvoie une réponse SUCCESS.
5. Amazon Cognito associe le facteur TOTP à l'utilisateur.
6. Si l'opération `VerifySoftwareToken` renvoie une réponse ERROR, assurez-vous que l'horloge de l'utilisateur est correcte et que ce dernier n'a pas dépassé le nombre maximal de nouvelles tentatives. Amazon Cognito accepte les jetons TOTP qui se trouvent dans les 30 secondes précédant ou suivant la tentative, pour tenir compte d'un décalage mineur d'horloge. Une fois le problème résolu, recommencez l'opération `VerifySoftwareToken`.

Connexion avec l'authentification MFA par TOTP

À ce stade, l'utilisateur se connecte à l'aide du mot de passe à usage unique à durée limitée. Le processus est le suivant.

1. L'utilisateur saisit son nom d'utilisateur et son mot de passe pour se connecter à votre application cliente.
2. La stimulation de la MFA par TOTP est appelée, et l'utilisateur est invité par votre appli à saisir un mot de passe temporaire.
3. Votre utilisateur obtient le mot de passe temporaire auprès d'une application de génération de mot de passe TOTP associée.

4. L'utilisateur entre le code TOTP dans votre application cliente. L'appli demande au service Amazon Cognito de le vérifier. Pour chaque connexion, vous [RespondToAuthChallenge](#) devez être appelé pour obtenir une réponse au nouveau défi d'authentification TOTP.
5. Si le jeton est vérifié par Amazon Cognito, la connexion est réussie et l'utilisateur poursuit le flux d'authentification.

Supprimer le jeton TOTP

Enfin, votre application doit permettre à votre utilisateur de désactiver sa configuration TOTP. Actuellement, vous ne pouvez pas supprimer le jeton logiciel TOTP d'un utilisateur. Pour remplacer le jeton logiciel de votre utilisateur, associez et vérifiez un nouveau jeton logiciel. Pour désactiver le MFA TOTP pour un utilisateur, appelez [SetUserMFAPreference](#) pour modifier votre utilisateur afin qu'il n'utilise pas le MFA ou uniquement le MFA par SMS.

1. Créez une interface dans votre application pour les utilisateurs qui souhaitent réinitialiser la MFA. Dans cette interface, invitez un utilisateur à saisir son mot de passe.
2. [Si Amazon Cognito renvoie un défi MFA TOTP, mettez à jour les préférences MFA de votre utilisateur avec MFAPreference. SetUser](#)
3. Dans votre application, indiquez à votre utilisateur qu'il a désactivé la MFA et demandez-lui de se reconnecter.

Configuration de votre ACL AWS WAF Web pour l'interface utilisateur hébergée (TOTP MFA)

Lorsqu'une ACL AWS WAF Web est associée à un groupe d'utilisateurs et qu'une règle de votre ACL Web présente un CAPTCHA, cela peut provoquer une erreur irrécupérable lors de l'enregistrement TOTP de l'interface utilisateur hébergée. AWS WAF Les règles CAPTCHA affectent uniquement le MFA TOTP dans l'interface utilisateur hébergée de cette manière. L'authentification MAF par SMS n'est pas affectée.

Amazon Cognito affiche l'erreur suivante lorsque votre règle CAPTCHA ne permet pas à un utilisateur de terminer la configuration MFA par TOTP.

La demande n'est pas autorisée en raison du captcha WAF.

Cette erreur se produit lorsque vous AWS WAF êtes invité à saisir un CAPTCHA en réponse à [AssociateSoftwareToken](#) des demandes d'[VerifySoftwareToken](#) API effectuées par votre groupe d'utilisateurs en arrière-plan. Pour créer une règle qui comporte une action CAPTCHA et qui n'affecte

pas le TOTP de l'interface utilisateur hébergée, excluez les valeurs d'en-tête `x-amzn-cognito-operation-name` égales à `AssociateSoftwareToken` et à `VerifySoftwareToken` à partir de l'action CAPTCHA de votre règle.

La capture d'écran suivante montre un exemple de AWS WAF règle qui applique une action CAPTCHA à toutes les demandes dont la valeur d'`x-amzn-cognito-operation-nameAssociateSoftwareToken` en-tête n'est pas `VerifySoftwareToken`

If a request matches all the statements (AND)

NOT Statement 1

Field to match

Single header (x-amzn-cognito-operation-name)

Positional constraint

Exactly matches string

Search string

AssociateSoftwareToken

Text transformations

- None (Priority 0)

AND

NOT Statement 2

Field to match

Single header (x-amzn-cognito-operation-name)

Positional constraint

Exactly matches string

Search string

VerifySoftwareToken

Text transformations

- None (Priority 0)

Then

Action

The action to take when a web request matches the rule statement.

Pour plus d'informations sur les ACL AWS WAF Web et Amazon Cognito, consultez. [Associer une ACL AWS WAF Web à un groupe d'utilisateurs](#)

Ajout de la sécurité avancée à un groupe d'utilisateurs

Une fois que vous avez créé votre groupe d'utilisateurs, vous avez accès à l'option Advanced security (Sécurité avancée) dans la barre de navigation de la console Amazon Cognito. Vous pouvez activer les fonctions de sécurité avancée des groupes d'utilisateurs et personnaliser les actions exécutées en réponse à différents risques. Ou vous pouvez utiliser le mode audit pour collecter des métriques sur les risques détectés sans qu'une atténuation de sécurité ne soit déclenchée. En mode audit, les fonctionnalités de sécurité avancées publient des métriques sur Amazon CloudWatch. Vous pouvez consulter des mesures de sécurité avancées une fois qu'Amazon Cognito a généré son premier événement de sécurité avancée. veuillez consulter [Affichage des métriques de sécurité avancée](#).

Les fonctions de sécurité avancée incluent la détection des informations d'identification compromises et l'authentification adaptative.

Informations d'identification compromises

Les utilisateurs réutilisent les mots de passe de plusieurs comptes utilisateurs. La fonctionnalité des informations d'identification compromises d'Amazon Cognito compile les données issues de fuites publiques de noms d'utilisateur et de mots de passe et compare les informations d'identification de vos utilisateurs à des listes d'informations d'identification divulguées. La détection des informations d'identification compromises permet également de vérifier les mots de passe les plus courants.

Vous pouvez choisir les actions de l'utilisateur qui invitent à vérifier la présence d'informations d'identification compromises et l'action que vous souhaitez qu'Amazon Cognito entreprenne en réponse. Pour les événements de connexion, d'inscription et de changement de mot de passe, Amazon Cognito peut Bloquer la connexion ou Autoriser la connexion. Dans les deux cas, Amazon Cognito génère un journal d'activité utilisateur dans lequel vous pouvez trouver plus d'informations sur l'événement.

Authentification adaptative

Amazon Cognito peut consulter les informations de localisation et d'appareil issues des demandes de connexion de vos utilisateurs et appliquer une réponse automatique pour protéger les comptes utilisateurs de votre groupe d'utilisateurs contre toute activité suspecte.

Lorsque vous activez la sécurité avancée, Amazon Cognito attribue un score de risque aux activités utilisateur. Vous pouvez attribuer une réponse automatique aux activités suspectes :

vous pouvez Demander l'authentification MFA, Bloquer la connexion ou simplement journaliser les détails de l'activité et le score de risque. Vous pouvez également envoyer automatiquement des e-mails pour informer votre utilisateur de l'activité suspecte afin qu'il puisse réinitialiser son mot de passe ou prendre d'autres mesures autonomes.

Personnalisation du jeton d'accès

Lorsque vous activez les fonctionnalités de sécurité avancées, vous pouvez configurer votre groupe d'utilisateurs pour qu'il accepte les réponses à un événement déclencheur Lambda version 2. Avec la version 2, vous pouvez personnaliser les portées et autres demandes des jetons d'accès. Cela augmente votre capacité à créer des résultats d'autorisation flexibles lorsque vos utilisateurs s'authentifient. Pour plus d'informations, consultez [Personnalisation du jeton d'accès](#).

Rubriques

- [Considérations et restrictions](#)
- [Prérequis](#)
- [Configuration des fonctions de sécurité avancée](#)
- [Vérification des informations d'identification compromises](#)
- [Utilisation de l'authentification adaptative](#)
- [Affichage des métriques de sécurité avancée](#)
- [Activation de la sécurité avancée des groupes d'utilisateurs à partir de votre application](#)

Considérations et restrictions

- Les fonctions de sécurité avancée d'Amazon Cognito entraînent un coût supplémentaire. Consultez la [page des tarifs d'Amazon Cognito](#).
- Amazon Cognito prend en charge l'authentification adaptative et la détection des informations d'identification compromises avec les flux d'authentification standard suivants :USER_PASSWORD_AUTH, et. ADMIN_USER_PASSWORD_AUTH USER_SRP_AUTH Vous ne pouvez pas utiliser la sécurité avancée avec un CUSTOM_AUTH flux et [Déclencheurs Lambda création d'une stimulation d'authentification personnalisée](#), ou avec une connexion fédérée.
- Avec les fonctionnalités de sécurité avancées d'Amazon Cognito en mode Fonction complet, vous pouvez créer des exceptions Toujours bloquer et Toujours autoriser pour l'adresse IP. Une session à partir d'une adresse IP figurant dans la liste d'exceptions Always block (Toujours bloquer) ne se

voit pas attribuer de niveau de risque par authentification adaptative et ne peut pas se connecter à votre groupe d'utilisateurs.

- Les demandes bloquées à partir des adresses IP figurant dans une liste d'exceptions Always block (Toujours bloquer) dans votre groupe d'utilisateurs contribuent aux [quotas de taux de demande](#) pour vos groupes d'utilisateurs. Les fonctions de sécurité avancées d'Amazon Cognito n'empêchent pas les attaques par déni de service distribué (DDoS). Pour mettre en œuvre des défenses contre les attaques volumétriques dans vos groupes d'utilisateurs, ajoutez des AWS WAF ACL Web. Pour plus d'informations, consultez [Associer une ACL AWS WAF Web à un groupe d'utilisateurs](#).
- Les autorisations d'identification des clients sont destinées à une autorisation machine-to-machine (M2M) sans connexion aux comptes utilisateurs. Des fonctions de sécurité avancée surveillent uniquement les comptes utilisateurs et les mots de passe de votre groupe d'utilisateurs. Pour implémenter des fonctionnalités de sécurité dans le cadre de votre activité M2M, considérez les capacités de surveillance AWS WAF des taux de demandes et du contenu. Pour plus d'informations, consultez [Associer une ACL AWS WAF Web à un groupe d'utilisateurs](#).

Prérequis

Avant de commencer, vous avez besoin de ce qui suit :

- Un groupe d'utilisateurs avec un client d'appli. Pour plus d'informations, consultez [Démarrage avec les groupes d'utilisateurs](#).
- Définissez l'authentification multifacteur (MFA) comme Optional (Facultative) dans la console Amazon Cognito pour utiliser la fonction d'authentification adaptative basée sur le risque. Pour plus d'informations, consultez [Ajout de l'authentification MFA à un groupe d'utilisateurs](#).
- Si vous utilisez des e-mails pour les notifications, accédez à la [console Amazon SES](#) pour configurer et vérifier une adresse e-mail ou un domaine à utiliser avec vos e-mails de notification. Pour plus d'informations sur Amazon SES, consultez [Vérification des identités dans Amazon SES](#).

Configuration des fonctions de sécurité avancée

Vous pouvez configurer les fonctions de sécurité avancées d'Amazon Cognito dans la AWS Management Console.

Pour configurer la sécurité avancée pour un groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
4. Choisissez l'onglet App integration (Intégration d'applications). Retrouvez Sécurité avancée et choisissez Activer. Si vous avez activé la sécurité avancée plus tôt, choisissez Modifier.
5. Sélectionnez Full function (Fonction complète) pour configurer les réponses de sécurité avancées sur les informations d'identification compromises et l'authentification adaptative. Sélectionnez Audit uniquement pour recueillir des informations et envoyer les données du groupe d'utilisateurs à CloudWatch. La tarification de la sécurité avancée s'applique dans les deux modes Audit uniquement et Fonction complète. Pour plus d'informations, consultez [Tarification d'Amazon Cognito](#).

Nous vous recommandons de conserver les fonctions de sécurité avancée en mode Audit pendant deux semaines avant d'activer des actions. Pendant ce temps, Amazon Cognito peut découvrir les modèles d'utilisation des utilisateurs de votre appli.

6. Si vous avez sélectionné Audit uniquement, choisissez Enregistrer les modifications. Si vous avez sélectionné Fonction complète :
 - a. Indiquez que vous allez exécuter une action personnalisée ou utiliser Paramètres Cognito par défaut pour traiter les informations d'identification compromises. Paramètres Cognito par défaut :
 - i. Détecter les informations d'identification compromises au moment de la connexion, de l'inscription et de la modification du mot de passe.
 - ii. Traiter les informations d'identification compromises avec l'action Bloquer la connexion.
 - b. Si vous avez sélectionné des actions personnalisées pour Compromised credentials (Informations d'identification compromises), choisissez les actions du groupe d'utilisateurs qu'Amazon Cognito utilisera pour la détection d'événements et les réponses aux informations d'identification compromises que vous voulez qu'Amazon Cognito apporte. Vous pouvez bloquer la connexion ou autoriser la connexion avec des informations d'identification compromises.
 - c. Choisissez comment répondre aux tentatives de connexion malveillantes sous Authentification adaptative. Indiquez si vous exécuterez une action personnalisée ou

utiliserez Paramètres Cognito par défaut pour traiter une activité malveillante présumée. Lorsque vous sélectionnez Paramètres Cognito par défaut, Amazon Cognito bloque la connexion à tous les niveaux de risque et ne prévient pas l'utilisateur.

- d. Si vous avez sélectionné des actions Personnalisées pour Authentification adaptative, choisissez les actions de Traitement automatique des risques d'Amazon Cognito en réponse aux risques détectés en fonction du niveau de sévérité. Lorsque vous attribuez une réponse à un niveau de risque, vous ne pouvez pas attribuer une réponse moins restrictive à un niveau de risque plus élevé. Vous pouvez attribuer les réponses suivantes aux niveaux de risque :
 - i. Autoriser la connexion - aucune action préventive n'est exécutée.
 - ii. Authentification MFA facultative - si l'utilisateur a configuré MFA, Amazon Cognito demandera toujours à l'utilisateur de fournir un SMS supplémentaire ou un mot de passe unique à durée limitée (TOTP) lorsqu'il se connecte. Si l'utilisateur n'a pas configuré MFA, il peut continuer à se connecter normalement.
 - iii. Demander l'authentification MFA - si l'utilisateur a configuré MFA, Amazon Cognito demandera toujours à l'utilisateur de fournir un SMS supplémentaire ou un mot de passe unique à durée limitée (TOTP). Si l'utilisateur n'a pas configuré MFA, Amazon Cognito l'invite à configurer MFA. Avant d'exiger automatiquement l'authentification MFA pour vos utilisateurs, configurez un mécanisme dans votre appli pour capturer les numéros de téléphone pour l'authentification MFA par SMS, ou pour enregistrer les applications d'authentification pour l'authentification MFA par mot de passe unique à durée limitée (TOTP).
 - iv. Bloquer la connexion : empêche l'utilisateur de se connecter.
 - v. Avertir l'utilisateur : envoie un courriel à l'utilisateur contenant des informations sur le risque détecté par Amazon Cognito et la réponse que vous avez choisie. Vous pouvez personnaliser des modèles d'e-mail pour les messages que vous envoyez.
7. Si vous avez choisi Notify user (Avertir l'utilisateur) à l'étape précédente, vous pouvez personnaliser vos paramètres de distribution d'e-mails et vos modèles d'e-mail pour l'authentification adaptative.
 - a. Sous Email configuration (Configuration d'e-mail), choisissez les paramètres SES Region (Région SES), FROM email address (Adresse e-mail d'envoi), FROM sender name (Nom de l'expéditeur) et REPLY-TO email address (Adresse e-mail de réponse) que vous souhaitez utiliser avec l'authentification adaptative. Pour plus d'informations sur l'intégration des

messages électroniques de votre groupe d'utilisateurs à Amazon Simple Email Service, consultez [Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito](#).

Adaptive authentication messages

Customize the messages sent to users when adaptive authentication triggers a notification. Adaptive authentication messages use [Amazon SES](#).

Email configuration

Configure the [Amazon SES](#) verified identity used to send adaptive authentication messages. [Learn more](#)

SES Region [Info](#)
Choose an AWS Region to use with SES in this user pool. For best performance, you should configure SES and your user pool in the same Region.

US East (N. Virginia) ▼

FROM email address [Info](#)
Choose an email address that you have verified with Amazon SES.

▼

FROM sender name - optional [Info](#)
Enter a friendly name for the email sender in the format "John Stiles <johnstiles@example.com>."

▼

REPLY-TO email address - optional [Info](#)
If you set an invalid reply-to address, sending restrictions may be imposed on your account.

▼

▼ **Email templates**

Risk detected, sign-in allowed

Email subject [Reset to default](#)

New sign-in attempt

Email message - Text [Reset to default](#) **Email message - HTML** [Reset to default](#)

We observed an unrecognized sign-in to your <!DOCTYPE html>

- b. Développez Email templates (Modèles d'e-mail) pour personnaliser les notifications d'authentification adaptative avec les versions HTML et en texte brut des e-mails. Pour en savoir plus sur les modèles d'e-mail, consultez [Modèles de messages](#).
8. Développer Exceptions d'adresses IP pour créer une liste Toujours autoriser ou Toujours bloquer de plages d'adresses IPv4 ou IPv6 qui seront toujours autorisées ou bloquées, quelle que soit l'évaluation des risques de la fonction de sécurité avancée. Indiquez les plages d'adresses IP en [notation CIDR](#) (par exemple, 192.168.100.0/24).

9. Sélectionnez Enregistrer les modifications.

Vérification des informations d'identification compromises

Amazon Cognito peut détecter si le nom d'utilisateur et le mot de passe d'un utilisateur ont été compromis ailleurs. Cela peut se produire lorsque des utilisateurs réutilisent des informations d'identification sur plusieurs sites, ou quand ils utilisent des mots de passe non sécurisés. Amazon Cognito vérifie les utilisateurs locaux qui se connectent avec un nom d'utilisateur et un mot de passe, dans l'interface utilisateur hébergée et avec l'API Amazon Cognito. Un utilisateur local existe exclusivement dans l'annuaire de votre groupe d'utilisateurs sans fédération via un fournisseur d'identité externe.

Dans Advanced security (Sécurité avancée), dans l'onglet App integration (Intégration d'applications) de la console Amazon Cognito, vous pouvez configurer Compromised credentials (Informations d'identification compromises). Configurez Event detection (Détection d'événements) pour choisir les événements utilisateur que vous souhaitez surveiller à la recherche d'informations d'identification compromises. Configurez Compromised credentials responses (Réponses d'informations d'identification compromises) pour autoriser ou bloquer l'utilisateur si des informations d'identification compromises sont détectées. Amazon Cognito peut vérifier les informations d'identification compromises lors des connexions, des inscriptions et des modifications de mot de passe.

Lorsque vous choisissez Autoriser la connexion, vous pouvez consulter Amazon CloudWatch Logs pour suivre les évaluations effectuées par Amazon Cognito sur les événements des utilisateurs. Pour plus d'informations, consultez [Affichage des métriques de sécurité avancée](#). Lorsque vous choisissez Block sign-in (Bloquer la connexion), Amazon Cognito empêche la connexion des utilisateurs qui utilisent des informations d'identification compromises. Quand Amazon Cognito bloque la connexion d'un utilisateur, il définit le paramètre `UserStatus` de l'utilisateur sur `RESET_REQUIRED`. Un utilisateur doté du statut `RESET_REQUIRED` doit modifier son mot de passe avant de pouvoir se reconnecter.

Note

Actuellement, Amazon Cognito ne vérifie pas les informations d'identification compromises pour les opérations de connexion avec un flux SRP (Secure Remote Password). SRP envoie une preuve de mot de passe hachée lors de la connexion. Amazon Cognito n'a pas accès aux mots de passe en interne. Il peut donc uniquement évaluer un mot de passe que votre client lui transmet en texte clair.

Amazon Cognito vérifie les connexions qui utilisent l'[AdminInitiateAuth](#) API avec ADMIN_USER_PASSWORD_AUTH flow, et l'API avec flow, pour détecter les informations [InitiateAuth](#) d'identification USER_PASSWORD_AUTH compromises.

Pour ajouter la protection contre les informations d'identification compromises à votre groupe d'utilisateurs, consultez [Ajout de la sécurité avancée à un groupe d'utilisateurs](#).

Utilisation de l'authentification adaptative

Avec l'authentification adaptative, vous pouvez configurer votre groupe d'utilisateurs pour bloquer les connexions suspectes ou ajouter un deuxième facteur d'authentification en réponse à un niveau de risque accru. Pour chaque tentative de connexion, Amazon Cognito génère un score de risque mesurant la probabilité que la demande de connexion provienne d'une source compromise. Ce score de risque est basé sur des facteurs qui incluent les informations relatives à l'appareil et à l'utilisateur. L'authentification adaptative peut activer ou exiger l'authentification multifactorielle (MFA) pour un utilisateur de votre groupe d'utilisateurs quand Amazon Cognito détecte un risque dans la session d'un utilisateur et que l'utilisateur n'a pas encore choisi de méthode MFA. Lorsque vous activez l'authentification MFA pour un utilisateur, celui-ci est toujours invité à fournir ou à configurer un second facteur lors de l'authentification, quelle que soit la manière dont vous avez configuré l'authentification adaptative. Du point de vue de l'utilisateur, votre application lui propose de l'aider à configurer l'authentification MFA et Amazon Cognito l'empêche éventuellement de se reconnecter tant qu'il n'a pas configuré un facteur supplémentaire.

Amazon Cognito publie les tentatives de connexion, leur niveau de risque et les échecs adressés à Amazon. CloudWatch Pour plus d'informations, consultez [Affichage des métriques de sécurité avancée](#).

Pour ajouter l'authentification adaptative à votre groupe d'utilisateurs, consultez [Ajout de la sécurité avancée à un groupe d'utilisateurs](#).

Rubriques

- [Présentation de l'authentification adaptative](#)
- [Ajout de données de session et de périphérique utilisateur aux demandes d'API](#)
- [Affichage de l'historique des événements de l'utilisateur](#)
- [Fourniture de commentaires sur des événements](#)
- [Envoi de messages de notification](#)

Présentation de l'authentification adaptative

Dans Sécurité avancée de l'onglet Intégration d'application de la console Amazon Cognito, vous pouvez choisir les paramètres pour l'authentification adaptative, notamment les actions à exécuter à différents niveaux de risque et la personnalisation des messages de notification envoyés aux utilisateurs. Vous pouvez attribuer une configuration de sécurité avancée globale à tous vos clients d'application, mais appliquez une configuration au niveau du client aux clients d'application individuels.


L'authentification adaptative Amazon Cognito attribue l'un des niveaux de risque suivants à chaque session utilisateur : élevé, moyen, faible ou aucun risque.

Examinez attentivement les options qui s'offrent à vous lorsque vous modifiez Enforcement method (Méthode d'application) en remplaçant Audit-only (Audit uniquement) par Full-function (Fonction complète). Les réponses automatiques que vous appliquez aux niveaux de risque influent sur le niveau de risque qu'Amazon Cognito affecte aux sessions d'utilisateur suivantes présentant les mêmes caractéristiques. Par exemple, une fois que vous avez choisi de ne réaliser aucune action ou de Allow (Autoriser) les sessions d'utilisateur qu'Amazon Cognito évalue initialement comme présentant un risque élevé, Amazon Cognito considère que les sessions similaires présentent un risque moindre.

Pour chaque niveau de risque, vous pouvez choisir parmi les options suivantes :

Option	Action
Autorisation	Les utilisateurs peuvent se connecter sans facteur supplémentaire.
Authentification MFA facultative	Les utilisateurs qui disposent d'un second facteur configuré doivent répondre à la demande de vérification du second facteur pour se connecter. Un numéro de téléphone pour SMS et un jeton logiciel TOTP sont les seconds facteurs disponibles. Les utilisateurs sans deuxième facteur configuré peuvent se connecter avec un seul ensemble d'informations d'identification.
Demander l'authentification MFA	Les utilisateurs qui disposent d'un second facteur configuré doivent répondre à la

Option	Action
	demande de vérification du second facteur pour se connecter. Amazon Cognito bloque la connexion des utilisateurs qui ne disposent pas d'un second facteur configuré.
Bloc	Amazon Cognito bloque toutes les tentatives de connexion au niveau de risque désigné.

 Note

Vous n'êtes pas tenu de vérifier les numéros de téléphone à utiliser pour l'envoi de SMS comme second facteur d'authentification.

Ajout de données de session et de périphérique utilisateur aux demandes d'API

Lorsque vous utilisez l'API pour inscrire un utilisateur, le connecter et réinitialiser son mot de passe, vous pouvez collecter les informations sur la session de cet utilisateur et les transmettre à la sécurité avancée d'Amazon Cognito. Ces informations incluent l'adresse IP de votre utilisateur et un identifiant de périphérique unique.

Vous disposez peut-être d'un périphérique réseau intermédiaire entre vos utilisateurs et Amazon Cognito, comme un service proxy ou un serveur d'applications. Vous pouvez collecter les données contextuelles des utilisateurs et les transmettre à Amazon Cognito, afin que l'authentification adaptative calcule votre risque en fonction des caractéristiques du point de terminaison utilisateur, au lieu de votre serveur ou de votre proxy. Si votre application côté client appelle directement les opérations de l'API Amazon Cognito, l'authentification adaptative enregistre automatiquement l'adresse IP source. Toutefois, elle n'enregistre pas d'autres informations de périphérique, telles que l'`user-agent`, sauf si vous collectez également une empreinte digitale de périphérique.

Générez ces données à l'aide de la bibliothèque de collecte de données contextuelles Amazon Cognito et soumettez-les à la sécurité avancée d'Amazon Cognito avec [ContextData](#) les paramètres et [UserContextData](#). La bibliothèque de collecte de données contextuelles est incluse dans les AWS SDK. Pour plus d'informations, consultez [Intégration d'Amazon Cognito aux applications Web et mobiles](#). Vous pouvez soumettre ContextData si vous avez activé des fonctionnalités de sécurité

avancée dans votre groupe d'utilisateurs. Pour plus d'informations, consultez [Configuration des fonctions de sécurité avancée](#).

Lorsque vous appelez les opérations d'API authentifiées Amazon Cognito suivantes depuis votre serveur d'applications, transmettez l'adresse IP du périphérique utilisateur dans le paramètre `ContextData`. En outre, transmettez le nom de votre serveur, le chemin d'accès du serveur et les données d'empreinte digitale du périphérique.

- [AdminInitiateAuth](#)
- [AdminRespondToAuthChallenge](#)

Lorsque vous appelez des opérations d'API non authentifiées Amazon Cognito, vous pouvez soumettre `UserContextData` aux fonctions de sécurité avancée d'Amazon Cognito. Ces données incluent une empreinte digitale de périphérique dans le paramètre `EncodedData`. Vous pouvez également soumettre un paramètre `IpAddress` dans `UserContextData` si les conditions suivantes sont remplies :

- Vous avez activé des fonctions de sécurité avancée dans votre groupe d'utilisateurs. Pour plus d'informations, consultez [Configuration des fonctions de sécurité avancée](#).
- Votre client d'application possède un secret client. Pour plus d'informations, consultez [Configuration d'un client d'application pour un groupe d'utilisateurs](#).
- Vous avez activé `Accept additional user context data` (Accepter des données contextuelles utilisateur supplémentaires) dans votre client d'application. Pour plus d'informations, consultez [Acceptation de données contextuelles utilisateur supplémentaires \(AWS Management Console\)](#).

Votre application peut renseigner le paramètre `UserContextData` avec les données d'empreinte digitale de périphérique codées et l'adresse IP du périphérique de l'utilisateur dans les opérations d'API non authentifiées Amazon Cognito suivantes.

- [InitiateAuth](#)
- [RespondToAuthChallenge](#)
- [SignUp](#)
- [ConfirmSignUp](#)
- [ForgotPassword](#)
- [ConfirmForgotPassword](#)

- [ResendConfirmationCode](#)

Acceptation de données contextuelles utilisateur supplémentaires (AWS Management Console)

Votre groupe d'utilisateurs accepte une adresse IP dans un paramètre `UserContextData`, une fois que vous avez activé la fonction `Accept additional user context data` (Accepter des données contextuelles utilisateur supplémentaires). Vous n'avez pas besoin d'activer cette fonctionnalité si :

- Vos utilisateurs ne se connectent qu'à l'aide d'opérations d'API authentifiées [AdminInitiateAuth](#), telles que, et vous utilisez le `ContextData` paramètre.
- vous souhaitez uniquement que vos opérations d'API non authentifiées envoient une empreinte digitale de périphérique, mais pas une adresse IP, aux fonctions de sécurité avancée d'Amazon Cognito.

Mettez à jour votre client d'application comme suit dans la console Amazon Cognito pour ajouter la prise en charge des données contextuelles utilisateur supplémentaires.

1. Connectez-vous à la [console Amazon Cognito](#).
2. Dans le volet de navigation, choisissez `Manage your User groupes` (Gérer vos groupes d'utilisateurs), puis choisissez le groupe d'utilisateurs que vous souhaitez modifier.
3. Choisissez l'onglet `App integration` (Intégration d'applications).
4. Sous `App clients and analytics` (Clients d'application et analytique), choisissez ou créez un client d'application. Pour plus d'informations, consultez [Configuration d'un client d'application pour groupe d'utilisateurs](#).
5. Choisissez `Edit (Modifier)` à partir du conteneur `App client information` (Informations sur le client d'application).
6. Dans `Advanced authentication settings` (Paramètres d'authentification avancée) pour votre client d'application, choisissez `Accept additional user context data` (Accepter des données contextuelles utilisateur supplémentaires).
7. Sélectionnez `Enregistrer les modifications`.

Pour configurer le client de votre application afin qu'il accepte les données contextuelles utilisateur dans l'API Amazon Cognito, définissez ce paramètre sur `EnablePropagateAdditionalUserContextData true` dans une demande [CreateUserPoolClient](#) or [UpdateUserPoolClient](#). Pour obtenir des informations sur la façon d'activer

la sécurité avancée à partir de votre application web ou mobile, consultez [Activation de la sécurité avancée des groupes d'utilisateurs à partir de votre application](#). Quand votre appli appelle Amazon Cognito à partir de votre serveur, collectez les données contextuelles utilisateur côté client. Voici un exemple d'utilisation de la méthode JavaScript `getData` SDK.

```
var encodedData =  
  AmazonCognitoAdvancedSecurityData.getData(username, userPoolId, clientId);
```

Lorsque vous concevez votre application pour utiliser l'authentification adaptative, nous vous recommandons d'intégrer le dernier kit SDK Amazon Cognito dans votre application. La dernière version du kit SDK collecte les informations d'empreinte digitale de l'appareil, telles que l'ID, le modèle et le fuseau horaire de celui-ci. Pour plus d'informations sur les kits SDK Amazon Cognito, consultez [Installer un kit SDK de groupe d'utilisateurs](#). La sécurité avancée d'Amazon Cognito enregistre et attribue uniquement un score de risque aux événements soumis par votre application dans le format approprié. Si Amazon Cognito renvoie une réponse d'erreur, vérifiez que votre demande contient un hachage secret valide et que le paramètre `IPAddress` est une adresse IPv4 ou IPv6 valide.

Ressources `ContextData` et `UserContextData`

- AWS Amplify SDK pour Android : [GetUserContextData](#)
- AWS Amplify SDK pour iOS : [userContextData](#)
- JavaScript: [amazon-cognito-advanced-security-data.min.js](#)

Affichage de l'historique des événements de l'utilisateur

Note

Dans la nouvelle console Amazon Cognito, vous pouvez afficher l'historique des événements de l'utilisateur dans l'onglet Utilisateurs.

Pour afficher l'historique de connexion d'un utilisateur, vous pouvez choisir l'utilisateur dans la console Amazon Cognito, dans l'onglet Users (Utilisateurs). Amazon Cognito conserve l'historique des événements utilisateur pendant deux ans.

Date (UTC)	Event	Result	Risk level	Risk decision	Challenge	IP	Device	Location	Event feedback
Jan 23, 2018 11:43:05 PM	Sign In	Pass	-	No Risk	Password:Success	52.94.36.11	Chrome, Windows 10	London	-
Jan 23, 2018 11:42:14 PM	Sign In	Pass	-	No Risk	Password:Success	52.94.36.11	Chrome, Windows 10	London	-
Jan 18, 2018 9:21:21 PM	Sign In	Fail	High	Account Takeover	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	-
Jan 18, 2018 9:20:28 PM	Sign In	In Progress	High	Account Takeover	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	-
Jan 18, 2018 9:18:18 PM	Sign In	Pass	-	No Risk	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	Invalid

5 per page < 1 2 3 >

Chaque événement de connexion possède un ID d'événement. L'événement a également des données contextuelles correspondantes, telles que l'emplacement, les détails de l'appareil et les résultats de détection des risques. [Vous pouvez consulter l'historique des événements des utilisateurs à l'aide de l'opération API Amazon Cognito AdminListUserAuthEvents ou à l'aide de AWS Command Line Interface \(AWS CLI\) with admin-list-user-auth -events.](#)

Vous pouvez également corréler l'ID de l'événement avec le jeton émis par Amazon Cognito au moment de l'enregistrement de l'événement. L'ID et les jetons d'accès incluent cet ID d'événement dans leur charge utile. Amazon Cognito établit également une corrélation entre l'utilisation du jeton d'actualisation et l'ID d'événement d'origine. Vous pouvez tracer l'ID d'événement d'origine en remontant jusqu'à l'ID de l'événement de connexion ayant conduit à l'émission des jetons Amazon Cognito. Vous pouvez tracer l'utilisation des jetons au sein de votre système jusqu'à un événement d'authentification particulier. Pour plus d'informations, consultez [Utilisation des jetons avec des groupes d'utilisateurs](#).

Fourniture de commentaires sur des événements

Les commentaires sur les événements ont un impact sur l'évaluation des risques en temps réel et améliorent l'algorithme d'évaluation des risques au fil du temps. Vous pouvez formuler des commentaires sur la validité des tentatives de connexion via la console Amazon Cognito et des opérations d'API.

 Note

Vos commentaires sur les événements influent sur le niveau de risque qu'Amazon Cognito affecte aux sessions d'utilisateur suivantes présentant les mêmes caractéristiques.

Dans la console Amazon Cognito, choisissez un utilisateur dans l'onglet Users (Utilisateurs) et sélectionnez Provide event feedback (Fournir des commentaires sur les événements). Vous pouvez passer en revue les détails de l'événement et sélectionner Set as valid (Définir comme valide) ou Set as invalid (Définir comme non valide).

La console répertorie l'historique des connexions dans l'onglet Users and groups (Utilisateurs et groupes). Si vous sélectionnez une entrée, vous pouvez marquer l'événement comme étant valide ou non valide. Vous pouvez également fournir des commentaires via l'opération [AdminUpdateAuthEventFeedbackAPI](#) du pool d'utilisateurs et via la AWS CLI commande [admin-update-auth-event-feedback](#).

Lorsque vous sélectionnez Set as valid (Définir comme valide) dans la console Amazon Cognito ou que vous fournissez une valeur `valid` pour `FeedbackValue` dans l'API, vous indiquez à Amazon Cognito que vous faites confiance à une session d'utilisateur dans laquelle Amazon Cognito a évalué un certain niveau de risque. Lorsque vous sélectionnez Set as invalid (Définir comme non valide) dans la console Amazon Cognito ou que vous fournissez une valeur `invalid` pour `FeedbackValue` dans l'API, vous indiquez à Amazon Cognito que vous ne faites pas confiance à une session d'utilisateur, ou que vous ne pensez pas qu'Amazon Cognito a évalué un niveau de risque suffisamment haut.

Envoi de messages de notification

Avec des protections de sécurité avancées, Amazon Cognito peut avertir les utilisateurs des tentatives de connexion risquées. Amazon Cognito peut également demander aux utilisateurs de sélectionner des liens pour indiquer si la connexion était valide ou non valide. Amazon Cognito utilise ces commentaires pour améliorer la précision de la détection des risques pour votre groupe d'utilisateurs.

Dans la section Automatic risk response (Réponse automatique aux risques), choisissez Notify Users (Avertir les utilisateurs) pour les situations à risque faible, moyen ou élevé.

Automatic risk response Info					
Risk level	Allow sign-in	Optional MFA	Require MFA	Block sign-in	Notify user
Low risk	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>
Medium risk	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>
High risk	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>

Amazon Cognito envoie des notifications par e-mail à vos utilisateurs, qu'ils aient confirmé ou non leur adresse e-mail.

Vous pouvez personnaliser les e-mails de notification, et fournir à la fois les versions en texte brut et HTML de ces messages. Pour personnaliser vos notifications par e-mail, ouvrez Email templates (Modèles d'e-mail) à partir de Adaptive authentication messages (Messages d'authentification adaptative) dans votre configuration de sécurité avancée. Pour en savoir plus sur les modèles de courriel, consultez [Modèles de messages](#).

Affichage des métriques de sécurité avancée

Amazon Cognito publie des statistiques relatives aux fonctionnalités de sécurité avancées de votre compte sur Amazon. CloudWatch Amazon Cognito regroupe les métriques de sécurité avancée par niveau de risque ainsi que par niveau de demande.

Pour afficher les métriques dans la CloudWatch console

- Ouvrez la CloudWatch console à l'[adresse https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/).
- Dans le panneau de navigation, sélectionnez Métriques.
- Choisissez Amazon Cognito.
- Choisissez un groupe de métriques regroupées, par exemple, By Risk Classification (Par classification de risque).
- L'onglet All metrics (Toutes les métriques) affiche toutes les métriques de votre choix. Vous pouvez effectuer les actions suivantes :
 - Pour trier le tableau, utilisez l'en-tête de colonne.
 - Pour représenter graphiquement une métrique, cochez la case en regard de la métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d'en-tête du tableau.

- Pour filtrer par ressource, sélectionnez l’ID de ressource, puis Add to search (Ajouter à la recherche).
- Pour filtrer par métrique, choisissez le nom de la métrique, puis Ajouter à la recherche.

Métrique	Description	Dimensions métriques
CompromisedCredentialRisk	Demands dans lesquelles Amazon Cognito a détecté des informations d’identification compromises.	Opération : les types d’opérations PasswordChange , SignIn ou SignUp sont les seules dimensions. UserPoolId: identifiant du groupe d'utilisateurs. RiskLevel: élevé (par défaut), moyen ou faible.
AccountTakeoverRisk	Demands dans lesquelles Amazon Cognito a détecté un risque de prise de contrôle de compte.	Opération : les types d’opérations PasswordChange , SignIn ou SignUp sont les seules dimensions. UserPoolId: identifiant du groupe d'utilisateurs. RiskLevel: élevé, moyen ou faible.
OverrideBlock	Demands bloquées par Amazon Cognito en raison de la configuration fournie par le développeur.	Opération : les types d’opérations PasswordChange , SignIn ou SignUp sont les seules dimensions. UserPoolId: identifiant du groupe d'utilisateurs. RiskLevel: élevé, moyen ou faible.

Métrique	Description	Dimensions métriques
Risque	Demands qu'Amazon Cognito a marquées comme risquées.	Opération : type d'opération, comme PasswordChange , SignIn ou SignUp. UserPoolId: identifiant du groupe d'utilisateurs.
NoRisk	Demands dans lesquelles Amazon Cognito n'a identifié aucun risque.	Opération : type d'opération, comme PasswordChange , SignIn ou SignUp. UserPoolId: identifiant du groupe d'utilisateurs.

Amazon Cognito vous propose deux groupes prédéfinis de métriques pour une analyse facile. CloudWatch By Risk Classification (Classification par risque) identifie la granularité du niveau de risque pour les demandes identifiées par Amazon Cognito comme risquées. By Request Classification (Classification par demande) reflète les métriques agrégées par niveau de demande.

Groupe de métriques regroupées	Description
By Risk Classification	Demands qu'Amazon Cognito identifie comme risquées.
By Request Classification	Métriques regroupées par demande.

Activation de la sécurité avancée des groupes d'utilisateurs à partir de votre application

Une fois que vous avez configuré les fonctions de sécurité avancée pour votre groupe d'utilisateurs, vous devez les activer dans votre application web ou mobile.

Utilisation d'une sécurité avancée avec JavaScript

1. Ajoutez le [SDK Amazon Cognito Identity pour JavaScript](#) à votre application.

2. Dans `CognitoUserPool.js`, définissez `AdvancedSecurityDataCollectionFlag` à `true`. Définissez `UserPoolId` sur l'ID de votre groupe d'utilisateurs.
3. Ajoutez cette référence de source au JavaScript fichier de votre application. Remplacez `<region>` par un Région AWS dans la liste suivante : `us-east-1`, `us-east-2`, `us-west-2`, `eu-west-1`, `eu-west-2`, `eu-central-1`.

```
<script src="https://amazon-cognito-assets.<region>.amazoncognito.com/amazon-cognito-advanced-security-data.min.js"></script>
```

Utilisation de la sécurité avancée avec Android

1. Créez votre application avec AWS Amplify pour Android. Pour plus d'informations, consultez [Configuration du projet](#) (langue française non garantie) sur le site AWS Amplify Dev Center.
2. Avec `userContextDataProvider`, incluez des informations sur l'utilisateur et l'appareil dans vos demandes d'authentification.

Pour plus d'informations sur l'ajout de données contextuelles utilisateur dans l'[ancien kit SDK Android](#), consultez [aws-android-sdk-cognitoidentityprovider-asf](#).

Utilisation de la sécurité avancée avec iOS

1. Créez votre application avec AWS Amplify for Swift ou Flutter. Pour plus d'informations, consultez [Configuration du projet](#) Swift et [Configuration du projet](#) Flutter (langue française non garantie) sur le site AWS Amplify Dev Center.
2. Incluez des informations sur l'utilisateur et l'appareil dans vos demandes d'authentification. Pour un exemple à utiliser avec l'opération d'`InitiateAuth` API, voir `userContextData` [InitiateAuthInput +Amplify.SWIFT](#) on. GitHub

Pour plus d'informations sur l'ajout de données contextuelles utilisateur dans l'[ancien kit SDK iOS](#), consultez [AWSCognitoIdentityProviderASF](#).

Associer une ACL AWS WAF Web à un groupe d'utilisateurs

AWS WAF est un pare-feu d'applications Web. Avec une liste de contrôle d'accès AWS WAF Web (ACL Web), vous pouvez protéger votre groupe d'utilisateurs contre les demandes indésirables adressées à votre interface utilisateur hébergée et aux points de terminaison du service d'API

Amazon Cognito. Une ACL web vous permet de contrôler avec précision toutes les requêtes web HTTPS auxquelles votre groupe d'utilisateurs répond. Pour plus d'informations sur les ACL AWS WAF Web, consultez [la section Gestion et utilisation d'une liste de contrôle d'accès Web \(ACL Web\)](#) dans le manuel du AWS WAF développeur.

Lorsqu'une ACL AWS WAF Web est associée à un groupe d'utilisateurs, Amazon Cognito transmet certains en-têtes non confidentiels et le contenu des demandes de vos utilisateurs à AWS WAF. AWS WAF inspecte le contenu de la demande, le compare aux règles que vous avez spécifiées dans votre ACL Web et renvoie une réponse à Amazon Cognito.

Ce qu'il faut savoir sur les ACL AWS WAF Web et Amazon Cognito

- Les demandes bloquées par AWS WAF ne sont pas prises en compte dans le quota de taux de demandes, quel que soit le type de demande. Le AWS WAF gestionnaire est appelé avant les gestionnaires de régulation au niveau de l'API.
- Lorsque vous créez une liste ACL web, peu de temps s'écoule avant qu'elle ne soit entièrement propagée et ne soit disponible pour Amazon Cognito. Le temps de propagation peut aller de quelques secondes à plusieurs minutes. AWS WAF renvoie un [WAFUnavailableEntityException](#) lorsque vous tentez d'associer une ACL Web avant qu'elle ne soit complètement propagée.
- Vous pouvez associer une liste ACL web à un groupe d'utilisateurs.
- Votre demande peut entraîner une charge utile supérieure aux limites de ce que AWS WAF peut inspecter. Consultez la section [Gestion des composants de demandes surdimensionnés](#) dans le guide du AWS WAF développeur pour savoir comment configurer le mode de gestion des demandes AWS WAF surdimensionnées provenant d'Amazon Cognito.
- Vous ne pouvez pas associer une ACL Web qui utilise AWS WAF [la prévention de la prise de contrôle des fraudes \(ATP\)](#) à un groupe d'utilisateurs Amazon Cognito. Vous implémentez la fonctionnalité ATP lorsque vous ajoutez le groupe de règles gérées par AWS-`AWSMANAGEDRULES`ATPRuleSet. Avant de l'associer à un groupe d'utilisateurs, assurez-vous que votre ACL web n'utilise pas ce groupe de règles gérées.
- Lorsqu'une ACL AWS WAF Web est associée à un groupe d'utilisateurs et qu'une règle de votre ACL Web présente un CAPTCHA, cela peut provoquer une erreur irrécupérable lors de l'enregistrement TOTP de l'interface utilisateur hébergée. Pour créer une règle qui comporte une action CAPTCHA et qui n'affecte pas l'enregistrement TOTP de l'interface utilisateur hébergée, veuillez consulter [Configuration de votre ACL AWS WAF Web pour l'interface utilisateur hébergée \(TOTP MFA\)](#).

AWS WAF inspecte les demandes adressées aux points de terminaison suivants.

Interface utilisateur hébergée

Demandes adressées à tous les points de terminaison de [Référence de l'interface utilisateur hébergée et des points de terminaison de la fédération des groupes d'utilisateurs](#).

Opérations d'API publiques

Demandes de votre application adressées à l'API Amazon Cognito qui n'utilisent pas AWS d'informations d'identification pour les autoriser. Cela inclut les opérations d'API telles que [InitiateAuthRespondToAuthChallenge](#), et [GetUser](#). Les opérations d'API concernées ne nécessitent AWS WAF pas d'authentification avec des informations d'AWS identification. Elles ne sont pas authentifiées ni autorisées par une chaîne de session ou un jeton d'accès. Pour plus d'informations, consultez [Opérations d'API authentifiées et non authentifiées des groupes d'utilisateurs Amazon Cognito](#).

Vous pouvez configurer les règles dans votre ACL web avec des actions de règles qui effectuent les opérations Count, Allow ou Block, ou qui présentent un CAPTCHA en réponse à une demande qui correspond à une règle. Pour plus d'informations, veuillez consulter la rubrique [Règles AWS WAF](#) dans le Manuel du développeur AWS WAF . En fonction de l'action de la règle, vous pouvez personnaliser la réponse qu'Amazon Cognito renvoie à vos utilisateurs.

Important

Les options qui s'offrent à vous pour personnaliser la réponse aux erreurs dépendent de la manière dont vous envoyez une demande d'API.

- Vous pouvez personnaliser le code d'erreur et le corps de réponse des requêtes d'interface utilisateur hébergées. Vous ne pouvez présenter un CAPTCHA pour que votre utilisateur puisse le résoudre dans l'interface utilisateur hébergée.
- Pour les demandes que vous faites avec l'[API de groupes d'utilisateurs](#) Amazon Cognito, vous pouvez personnaliser le corps de la réponse d'une demande qui reçoit une réponse Bloquer. Vous pouvez également spécifier un code d'erreur personnalisé compris entre 400 et 499.
- Le AWS Command Line Interface (AWS CLI) et les AWS SDK renvoient une `ForbiddenException` erreur aux demandes qui produisent une réponse Block ou CAPTCHA.

Associer une ACL web à votre groupe d'utilisateurs

Pour utiliser une ACL Web dans votre groupe d'utilisateurs, votre principal AWS Identity and Access Management (IAM) doit disposer des autorisations Amazon Cognito suivantes. Pour plus d'informations sur AWS WAF les autorisations, consultez la section [Autorisations d'AWS WAF API](#) dans le Guide du AWS WAF développeur.

- `cognito-idp:AssociateWebACL`
- `cognito-idp:DisassociateWebACL`
- `cognito-idp:GetWebACLForResource`
- `cognito-idp:ListResourcesForWebACL`

Bien que vous deviez accorder des autorisations IAM, les actions répertoriées sont uniquement des autorisations et ne correspondent à aucune [opération d'API](#).

Pour l'activer AWS WAF pour votre groupe d'utilisateurs et associer une ACL Web

1. Connectez-vous à la [console Amazon Cognito](#).
2. Dans le volet de navigation, choisissez Groupes d'utilisateurs, puis choisissez le groupe d'utilisateurs que vous souhaitez modifier.
3. Choisissez l'onglet User pool properties (Propriétés du groupe d'utilisateurs).
4. Choisissez Edit (Modifier) en regard de AWS WAF.
5. Sous AWS WAF, sélectionnez Utiliser AWS WAF avec votre groupe d'utilisateurs.

AWS WAF

Use AWS WAF web ACLs to monitor requests to your user pool.

AWS WAF

Use AWS WAF with your user pool - Recommended
Activate support for AWS WAF web ACLs in this user pool. AWS WAF can add cost to your bill. [Learn more about AWS WAF pricing](#)

AWS WAF Web ACL

Choose a web access control list (web ACL) that you want to associate with your user pool.

demo-webacl ▼ ↻ 🔗 View Web ACL

🔗 Create Web ACL in AWS WAF

6. Choisissez une ACL AWS WAF Web que vous avez déjà créée, ou choisissez Create Web ACL in AWS WAF pour en créer une dans une nouvelle AWS WAF session dans le AWS Management Console.
7. Sélectionnez Enregistrer les modifications.

Pour associer par programmation une ACL Web à votre groupe d'utilisateurs dans le SDK AWS Command Line Interface ou dans un SDK, utilisez l'[AssociateWebACL](#) depuis l'API. AWS WAF Amazon Cognito ne dispose pas d'une opération d'API distincte qui associe une ACL Web.

Test et journalisation des AWS WAF ACL Web

Lorsque vous définissez une action de règle sur Count dans votre ACL Web AWS WAF , vous ajoutez la demande au nombre de demandes correspondant à la règle. Pour tester une ACL web avec votre groupe d'utilisateurs, définissez les actions des règles sur Count et examinez le volume de demandes correspondant à chaque règle. Par exemple, si une règle que vous souhaitez définir comme action Block correspond à un grand nombre de demandes que vous considérez comme relevant du trafic utilisateur normal. Vous devrez peut-être reconfigurer votre règle. Pour plus d'informations, consultez la section [Tester et régler vos AWS WAF protections](#) dans le Guide du AWS WAF développeur.

Vous pouvez également configurer AWS WAF pour consigner les en-têtes des demandes dans un groupe de CloudWatch journaux Amazon Logs, un bucket Amazon Simple Storage Service (Amazon S3) ou un Amazon Data Firehose. Vous pouvez identifier les demandes Amazon Cognito que vous faites à l'aide de l'API des groupes d'utilisateurs grâce à x-amzn-cognito-client-id et x-amzn-cognito-operation-name. Les requêtes d'interface utilisateur hébergée incluent uniquement l'en-tête x-amzn-cognito-client-id. Pour plus d'informations, consultez [Journalisation du trafic ACL web](#) dans le Guide du développeur AWS WAF .

AWS WAF Les ACL Web ne sont pas soumises à la [tarification des fonctionnalités de sécurité avancées d'Amazon Cognito](#). Les fonctionnalités de sécurité AWS WAF complètent les fonctionnalités de sécurité avancées d'Amazon Cognito. Vous pouvez activer les deux fonctionnalités dans un groupe d'utilisateurs. AWS WAF facture séparément pour l'inspection des demandes du groupe d'utilisateurs. Pour plus d'informations, consultez [Tarification d'AWS WAF](#).

Les données de AWS WAF demande de journalisation sont soumises à une facturation supplémentaire par le service sur lequel vous ciblez vos logs. Pour plus d'informations, veuillez consulter la rubrique [Tarification pour la journalisation des informations sur le trafic ACL web](#) dans le Manuel du développeur AWS WAF .

Sensibilité à la casse du groupe d'utilisateurs

Par défaut, les groupes d'utilisateurs Amazon Cognito que vous créez dans le groupe AWS Management Console ne distinguent pas les majuscules et minuscules. Lorsqu'un groupe d'utilisateurs n'est pas sensible à la casse, `utilisateur@exemple.com` et `User@exemple.com` reportez-vous au même utilisateur. Lorsque les noms d'utilisateur d'un groupe d'utilisateurs ne sont pas sensibles à la casse, il en va de même pour les attributs `preferred_username` et `email`.

Pour prendre en compte les paramètres de sensibilité à la casse du groupe d'utilisateurs, identifiez les utilisateurs dans le code de votre application en fonction d'un autre attribut d'utilisateur. Comme la casse d'un nom d'utilisateur, d'un nom d'utilisateur préféré ou d'un attribut d'adresse e-mail peut varier dans différents profils d'utilisateur, reportez-vous à la place à l'attribut `sub`. Vous pouvez également créer un attribut personnalisé inaltérable dans votre groupe d'utilisateurs et attribuer votre propre valeur d'identificateur unique à l'attribut de chaque nouveau profil d'utilisateur. Quand vous créez un utilisateur pour la première fois, vous pouvez écrire une valeur dans l'attribut personnalisé inaltérable que vous avez créé.

Note

Quels que soient les paramètres de sensibilité à la casse de votre groupe d'utilisateurs, Amazon Cognito exige qu'un utilisateur fédéré d'un fournisseur d'identité (IdP) SAML ou OIDC transmette une revendication `NameId` ou `sub` unique et sensible à la casse. Pour plus d'informations sur la distinction majuscules/minuscules des identificateurs uniques et sur le protocole SAML IdPs, consultez [Utilisation de la connexion SAML initiée par le SP](#).

Création d'un groupe d'utilisateurs sensible à la casse

Si vous créez des ressources avec le AWS Command Line Interface (AWS CLI) et des opérations d'API telles que [CreateUserPool](#), vous devez définir le `CaseSensitive` paramètre booléen sur `false`. Ce paramètre crée un groupe d'utilisateurs insensible à la casse. Si vous ne spécifiez aucune valeur, le `CaseSensitive` utilise la valeur `true` par défaut. Ce comportement par défaut est à l'opposé du comportement par défaut des groupes d'utilisateurs que vous créez dans AWS Management Console. Avant le 12 février 2020, les groupes d'utilisateurs étaient sensibles à la casse, quelle que soit la plate-forme.

Vous pouvez utiliser l'onglet Expérience de connexion AWS Management Console ou l'[DescribeUserPool](#) API pour vérifier les paramètres de distinction majuscules/minuscules pour chaque groupe d'utilisateurs de votre compte.

Migration vers un nouveau groupe d'utilisateurs

En raison des conflits potentiels entre les profils utilisateur, vous ne pouvez pas rendre insensible à la casse un groupe d'utilisateurs Amazon Cognito sensible à la casse. Au lieu de cela, migrez vos utilisateurs vers un nouveau groupe d'utilisateurs. Vous devez créer un code de migration pour résoudre les conflits liés à la casse. Ce code doit renvoyer un nouvel utilisateur unique ou rejeter la tentative de connexion s'il détecte un conflit. Dans un nouveau groupe d'utilisateurs insensible à la casse, attribuez un [Déclencheur Lambda de migration d'utilisateur](#). La AWS Lambda fonction peut créer des utilisateurs dans le nouveau groupe d'utilisateurs qui ne fait pas la distinction majuscules/minuscules. Quand l'utilisateur ne parvient pas à se connecter avec le groupe d'utilisateurs insensible à la casse, la fonction Lambda détecte et duplique l'utilisateur à partir du groupe d'utilisateurs sensible à la casse. Vous pouvez également activer un déclencheur Lambda de migration utilisateur lors [ForgotPassword](#) d'événements. Amazon Cognito transmet les informations utilisateur et les métadonnées d'événements de l'action de connexion ou de récupération de mot de passe à votre fonction Lambda. Vous pouvez utiliser les données d'événement pour gérer les conflits entre les noms d'utilisateur et les adresses e-mail lorsque votre fonction crée le nouvel utilisateur dans votre groupe d'utilisateurs insensible à la casse. Ces conflits interviennent entre des noms d'utilisateur et des adresses e-mail qui seraient uniques dans un groupe d'utilisateurs insensible à la casse, mais identiques dans un groupe d'utilisateurs sensible à la casse.


Pour plus d'informations sur l'utilisation d'un déclencheur Lambda de migration d'utilisateurs entre des groupes d'utilisateurs Amazon Cognito, [consultez la section Migration d'utilisateurs vers des groupes d'utilisateurs Amazon Cognito sur](#) le blog. AWS

Protection contre la suppression du groupe d'utilisateurs

Pour éviter que vos administrateurs ne suppriment accidentellement votre groupe d'utilisateurs, activez la protection contre la suppression. Lorsque la protection contre la suppression est active, vous devez confirmer que vous souhaitez supprimer votre groupe d'utilisateurs avant de le supprimer. Lorsque vous supprimez un groupe d'utilisateurs dans la AWS Management Console, vous pouvez désactiver la protection contre la suppression au même moment. Lorsque vous acceptez l'invite de désactivation de la protection contre la suppression et que vous confirmez la suppression, comme illustré dans l'image suivante, Amazon Cognito supprime votre groupe d'utilisateurs.

Delete user pool [redacted] ? ✕

Before you delete this user pool, first make sure no services or apps rely on it.

 If you delete this user pool, and your app still relies on it, any sign-in and sign-up attempts will fail.

- To delete this user pool, permit Amazon Cognito to also take the following prerequisite actions.
 - Deactivate deletion protection**
- To confirm deletion, enter testUserPool in the field.

testUserPool

Cancel Delete

Pour supprimer un groupe d'utilisateurs à l'aide d'une demande d'API Amazon Cognito, vous devez d'abord affecter à `DeletionProtection` la valeur `Inactive` dans une demande [UpdateUserPool](#). Si vous ne désactivez pas la protection contre la suppression, Amazon Cognito renvoie une erreur `InvalidParameterException`. Après avoir désactivé la protection contre la suppression, vous pouvez supprimer le groupe d'utilisateurs dans une demande [DeleteUserPool](#).

Amazon Cognito active par défaut `Deletion protection` (Protection contre la suppression) lorsque vous créez un nouveau groupe d'utilisateurs dans la AWS Management Console. Lorsque vous créez un groupe d'utilisateurs à l'aide de l'API `CreateUserPool`, la protection contre la suppression est inactive par défaut. Pour utiliser cette fonctionnalité dans les groupes d'utilisateurs que vous créez à l'aide de l'AWS CLI ou d'un kit SDK AWS, définissez le paramètre `DeletionProtection` sur `True`.

Vous pouvez activer ou désactiver le statut de protection contre la suppression dans le conteneur `Deletion protection` (Protection contre la suppression) de l'onglet `User pool settings` (Paramètres du groupe d'utilisateurs) de la console Amazon Cognito.

Pour configurer la protection contre la suppression

- Accédez à la [console Amazon Cognito](#). Il se peut que vous soyez invité à saisir vos informations d'identification AWS.

2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#).
4. Choisissez l'onglet User pool settings (Paramètres du groupe d'utilisateurs). Localisez Deletion Protection (Protection contre la suppression) et sélectionnez Activate (Activer) ou Deactivate (Désactiver).
5. Confirmez votre choix dans la boîte de dialogue suivante.

Gestion des réponses aux erreurs liées à l'existence des utilisateurs

Amazon Cognito prend en charge la personnalisation des réponses d'erreur renvoyées par les groupes d'utilisateurs. Des réponses d'erreur personnalisées sont disponibles pour les opérations de création et d'authentification, de récupération de mot de passe et de confirmation d'utilisateurs.

Utilisez le paramètre `PreventUserExistenceErrors` d'un client d'application de groupe d'utilisateurs pour activer ou désactiver les erreurs liées à l'existence de l'utilisateur. Lorsque vous créez un nouveau client d'application avec l'API des groupes d'utilisateurs Amazon Cognito, elle `PreventUserExistenceErrors` est `LEGACY` ou est désactivée par défaut. Dans la console Amazon Cognito, l'option Empêcher les erreurs liées à l'existence des utilisateurs (paramètre `ENABLED` pour `PreventUserExistenceErrors`) est sélectionnée par défaut. Pour mettre à jour votre `PreventUserExistenceErrors` configuration, effectuez l'une des opérations suivantes :

- Modifiez la valeur comprise `PreventUserExistenceErrors` entre `ENABLED` et `LEGACY` dans une demande d'[UpdateUserPoolClientAPI](#).
- Modifiez le client de votre application dans la console Amazon Cognito et modifiez l'état de Empêcher les erreurs d'existence des utilisateurs entre sélectionné (`ENABLED`) et désélectionné (`LEGACY`).

Lorsque cette propriété a une valeur égale à `LEGACY`, le client de votre application renvoie une réponse `UserNotFoundException` d'erreur lorsqu'un utilisateur tente de se connecter avec un nom d'utilisateur qui n'existe pas dans votre groupe d'utilisateurs.

Lorsque cette propriété a une valeur de `ENABLED`, le client de votre application ne divulgue pas l'inexistence d'un compte utilisateur dans votre groupe d'utilisateurs avec une `UserNotFoundException` erreur. Une `PreventUserExistenceErrors` configuration de `ENABLED` produit les effets suivants :

- Amazon Cognito fournit des informations non spécifiques aux demandes d'API lorsque sa réponse pourrait sinon révéler l'existence d'un utilisateur valide.
- Les API de connexion et de mot de passe oublié d'Amazon Cognito renvoient une réponse générique en cas d'échec d'authentification. La réponse d'erreur vous indique que le nom d'utilisateur ou le mot de passe est incorrect.
- Les API de confirmation de compte et de récupération de mot de passe Amazon Cognito renvoient une réponse indiquant qu'un code a été envoyé à un support de diffusion simulé, au lieu d'une représentation partielle des coordonnées d'un utilisateur.

Les informations suivantes détaillent les comportements des opérations du groupe d'utilisateurs lorsque `PreventUserExistenceErrors` ce paramètre est défini sur `ENABLED`.

Opérations d'authentification et de création d'utilisateurs

Vous pouvez configurer les réponses aux erreurs dans le cadre de l'authentification par nom d'utilisateur-mot de passe et par mot de passe distant sécurisé (SRP). Vous pouvez également personnaliser les erreurs que vous renvoyez avec une authentification personnalisée. Les API suivantes exécutent ces opérations d'authentification :

- `AdminInitiateAuth`
- `AdminRespondToAuthChallenge`
- `InitiateAuth`
- `RespondToAuthChallenge`

La liste suivante montre comment personnaliser les réponses d'erreur dans les opérations d'authentification des utilisateurs.

Authentification par nom d'utilisateur et mot de passe

Pour connecter un utilisateur avec `ADMIN_USER_PASSWORD_AUTH` et `USER_PASSWORD_AUTH`, incluez le nom d'utilisateur et le mot de passe dans une demande d'API `AdminInitiateAuth` ou `InitiateAuth`. Amazon Cognito renvoie une erreur `NotAuthorizedException` générique lorsque le nom d'utilisateur ou le mot de passe sont incorrects.

Authentification basée sur un mot de passe distant sécurisé (Secure Remote Password, SRP)

Pour connecter un utilisateur avec `USER_SRP_AUTH`, incluez un nom d'utilisateur et un paramètre `SRP_A` dans une demande d'API `AdminInitiateAuth` ou `InitiateAuth`. En réponse,

Amazon Cognito renvoie du sel SRP_B à l'utilisateur. Si un utilisateur n'est pas trouvé, Amazon Cognito renvoie une réponse simulée lors de la première étape, comme décrit dans [RFC 5054](#). Amazon Cognito renvoie la même valeur salt et un ID d'utilisateur interne au format [UUID \(identifiant universel unique\)](#) pour la même combinaison de nom d'utilisateur et de groupe d'utilisateurs. Lorsque vous envoyez une demande d'API RespondToAuthChallenge avec preuve de mot de passe, Amazon Cognito renvoie une erreur `NotAuthorizedException` générique lorsque le nom d'utilisateur ou le mot de passe sont incorrects.

Note

Vous pouvez simuler une réponse générique avec l'authentification par nom d'utilisateur et mot de passe si vous utilisez des attributs d'alias basés sur la vérification, et que le nom d'utilisateur immuable ne se présente pas sous la forme d'un UUID.

Déclencheur Lambda de stimulation d'authentification personnalisée

Si vous utilisez le [déclencheur Lambda de stimulation d'authentification personnalisée](#) et que vous activez les réponses d'erreur, `LambdaChallenge` renvoie un paramètre booléen nommé `UserNotFound`. Il est ensuite passé dans la demande de déclencheurs Lambda `DefineAuthChallenge`, `VerifyAuthChallenge` et `CreateAuthChallenge`. Vous pouvez utiliser ce déclencheur pour simuler des stimulations d'autorisation personnalisée pour un utilisateur inexistant. Si vous appelez le déclencheur Lambda d'authentification préalable pour un utilisateur inexistant, Amazon Cognito renvoie `UserNotFound`.

La liste suivante montre comment personnaliser les réponses d'erreur lors des opérations de création d'utilisateurs.

SignUp

L'opération `SignUp` revient toujours `UsernameExistsException` lorsqu'un nom d'utilisateur est déjà utilisé. Si vous ne voulez pas qu'Amazon Cognito renvoie une erreur `UsernameExistsException` pour les adresses e-mail et les numéros de téléphone au moment où vous inscrivez des utilisateurs dans votre application, utilisez des attributs d'alias basés sur la vérification. Pour en savoir plus sur les alias, consultez la section [Personnalisation des attributs de connexion](#).

Pour voir un exemple de la façon dont Amazon Cognito peut empêcher l'utilisation de demandes d'API `SignUp` pour découvrir les utilisateurs de votre groupe d'utilisateurs, consultez [Prévention des erreurs `UsernameExistsException` pour les adresses e-mail et les numéros de téléphone au moment de l'inscription](#).

Utilisateurs importés

Si l'option `PreventUserExistenceErrors` est activée, lors de l'authentification d'utilisateurs importés, une erreur `NotAuthorizedException` générique est renvoyée, indiquant que le nom d'utilisateur ou le mot de passe étaient incorrects au lieu de renvoyer `PasswordResetRequiredException`. Pour plus d'informations, consultez [Obligation pour les utilisateurs importés de réinitialiser leur mot de passe](#).

Déclencheur Lambda de migration d'utilisateur

Amazon Cognito renvoie une réponse simulée pour des utilisateurs inexistantes quand une réponse vide a été définie dans le contexte d'événement d'origine par le déclencheur Lambda. Pour plus d'informations, consultez [Déclencheur Lambda de migration d'utilisateur](#).

Prévention des erreurs **`UsernameExistsException`** pour les adresses e-mail et les numéros de téléphone au moment de l'inscription

L'exemple suivant montre comment, au moment de configurer des attributs d'alias dans votre groupe d'utilisateurs, vous pouvez empêcher que des adresses e-mail et des numéros de téléphone en double ne génèrent des erreurs `UsernameExistsException` en réponse à des demandes d'API `SignUp`. Vous devez avoir créé votre groupe d'utilisateurs avec l'adresse e-mail ou le numéro de téléphone en tant qu'attribut d'alias. Pour en savoir plus, consultez la section Personnalisation des attributs de connexion du document [Attributs de groupe d'utilisateurs](#).

1. Jie s'inscrit pour obtenir un nouveau nom d'utilisateur et fournit également l'adresse e-mail `jie@example.com`. Amazon Cognito envoie un code à son adresse e-mail.

Exemple de AWS CLI commande

```
aws cognito-idp sign-up --client-id 1234567890abcdef0 --username jie --password  
PASSWORD --user-attributes Name="email",Value="jie@example.com"
```

Exemple de réponse

```
{
```

```
"UserConfirmed": false,
"UserSub": "<subId>",
"CodeDeliveryDetails": {
  "AttributeName": "email",
  "Destination": "j****@e****",
  "DeliveryMedium": "EMAIL"
}
}
```

2. Jie fournit le code qui lui a été envoyé pour confirmer que l'adresse e-mail lui appartient. Cela termine son inscription en tant qu'utilisateur.

Exemple de AWS CLI commande

```
aws cognito-idp confirm-sign-up --client-id 1234567890abcdef0 --username=jie --
confirmation-code xxxxxx
```

3. Shirley inscrit un nouveau compte d'utilisateur et fournit l'adresse e-mail `jie@example.com`. Amazon Cognito ne renvoie pas d'erreur `UsernameExistsException` et envoie un code de confirmation à l'adresse e-mail de Jie.

Exemple de AWS CLI commande

```
aws cognito-idp sign-up --client-id 1234567890abcdef0 --username shirley --password
PASSWORD --user-attributes Name="email",Value="jie@example.com"
```

Exemple de réponse

```
{
  "UserConfirmed": false,
  "UserSub": "<new subId>",
  "CodeDeliveryDetails": {
    "AttributeName": "email",
    "Destination": "j****@e****",
    "DeliveryMedium": "EMAIL"
  }
}
```

4. Dans un autre scénario, Shirley est propriétaire de `jie@example.com`. Shirley récupère le code qu'Amazon Cognito a envoyé à l'adresse e-mail de Jie et tente de confirmer le compte.

Exemple de AWS CLI commande

```
aws cognito-idp confirm-sign-up --client-id 1234567890abcdef0 --username=shirley --
confirmation-code xxxxxx
```

Exemple de réponse

```
An error occurred (AliasExistsException) when calling the ConfirmSignUp operation: An
account with the email already exists.
```

Amazon Cognito ne renvoie pas d'erreur à la demande `aws cognito-idp sign-up` de Shirley, bien que `jie@example.com` soit attribué à un utilisateur existant. Shirley doit prouver qu'elle est propriétaire de l'adresse e-mail avant qu'Amazon Cognito ne renvoie une réponse d'erreur. Dans un groupe d'utilisateurs doté d'attributs d'alias, ce comportement empêche l'utilisation de l'API `SignUp` publique pour vérifier l'existence d'un utilisateur à partir d'une adresse e-mail ou d'un numéro de téléphone donnés.

Ce comportement est différent de la réponse qu'Amazon Cognito renvoie à une demande `SignUp` associée à un nom d'utilisateur existant, comme le montre l'exemple suivant. Bien que cette réponse fasse découvrir à Shirley qu'il existe déjà un utilisateur doté du nom d'utilisateur `jie`, elle ne lui apprend rien concernant les adresses e-mail et les numéros de téléphone associés à cet utilisateur.

Exemple de commande d'interface de ligne de commande

```
aws cognito-idp sign-up --client-id 1example23456789 --username jie --password PASSWORD
--user-attributes Name="email",Value="shirley@example.com"
```

Exemple de réponse

```
An error occurred (UsernameExistsException) when calling the SignUp operation: User
already exists
```

Opérations de réinitialisation de mot de passe

Amazon Cognito renvoie les réponses suivantes aux opérations de réinitialisation du mot de passe utilisateur lorsque vous empêchez les erreurs liées à l'existence d'un utilisateur.

ForgotPassword

Quand un utilisateur n'est pas trouvé, est désactivé ou ne dispose d'aucun mécanisme de communication vérifié pour récupérer son mot de passe, Amazon Cognito renvoie `CodeDeliveryDetails` avec un mode de communication simulé pour un utilisateur. Le mode de communication simulé est déterminé par le format du nom d'utilisateur d'entrée et les paramètres de vérification du pool d'utilisateurs.

ConfirmForgotPassword

Amazon Cognito renvoie l'erreur `CodeMismatchException` pour des utilisateurs inexistantes ou désactivés. Si aucun code n'est demandé lors de l'utilisation de `ForgotPassword`, Amazon Cognito renvoie l'erreur `ExpiredCodeException`.

Opérations de confirmation

Amazon Cognito renvoie les réponses suivantes aux opérations de confirmation et de vérification de l'utilisateur lorsque vous empêchez les erreurs liées à l'existence d'un utilisateur.

ResendConfirmationCode

Amazon Cognito renvoie `CodeDeliveryDetails` pour un utilisateur désactivé ou inexistant. Amazon Cognito envoie un code de confirmation au courriel ou au numéro de téléphone de l'utilisateur existant.

ConfirmSignUp

`ExpiredCodeException` renvoie si un code a expiré. Amazon Cognito retourne `NotAuthorizedException` quand un utilisateur n'est pas autorisé. Si le code ne correspond pas à ce que le serveur attend, Amazon Cognito renvoie `CodeMismatchException`.

Groupes d'identités Amazon Cognito

Une réserve d'identités Amazon Cognito est un annuaire d'identités fédérées que vous pouvez échanger contre des informations d'identification AWS. Les pools d'identités génèrent des AWS informations d'identification temporaires pour les utilisateurs de votre application, qu'ils soient connectés ou que vous ne les ayez pas encore identifiés. Avec les rôles et les politiques AWS Identity and Access Management (IAM), vous pouvez choisir le niveau d'autorisation que vous souhaitez accorder à vos utilisateurs. Les utilisateurs peuvent commencer en tant qu'invités et récupérer les ressources que vous conservez dans les Services AWS. Ils peuvent ensuite se connecter auprès d'un fournisseur d'identité tiers pour débloquer l'accès aux ressources que vous mettez à la disposition des membres enregistrés. Le fournisseur d'identité tiers peut être un fournisseur OAuth 2.0 grand public (social) tel qu'Apple ou Google, un fournisseur d'identité SAML ou OIDC personnalisé, ou un schéma d'authentification personnalisé, également appelé fournisseur de développement, de votre propre conception.

Fonctionnalités des réserves d'identités Amazon Cognito

Signer des demandes pour Services AWS

[Signez des demandes d'API](#) Services AWS comme Amazon Simple Storage Service (Amazon S3) et Amazon DynamoDB. Analysez l'activité des utilisateurs avec des services tels qu'Amazon Pinpoint et Amazon CloudWatch

Filtrer les demandes avec des politiques basées sur les ressources

Exercez un contrôle précis sur l'accès des utilisateurs à vos ressources. Transformez les champs standard utilisateur en [balises de session IAM](#) et créez des politiques IAM qui accordent l'accès aux ressources à divers sous-ensembles de vos utilisateurs.

Attribuer un accès invité

Pour vos utilisateurs qui ne se sont pas encore connectés, configurez votre réserve d'identités pour générer des informations d'identification AWS avec une étendue d'accès restreinte. Authentifiez les utilisateurs via un fournisseur d'authentification unique pour améliorer leur accès.

Attribuer des rôles IAM en fonction des caractéristiques de l'utilisateur

Attribuez un rôle IAM unique à tous vos utilisateurs authentifiés ou choisissez le rôle en fonction des champs standard de chaque utilisateur.

Accepter divers fournisseurs d'identité

Échangez un identifiant ou un jeton d'accès, un jeton de groupe d'utilisateurs, une assertion SAML ou un jeton OAuth du fournisseur social contre des informations d'identification. AWS

Valider vos propres identités

Procédez à votre propre validation utilisateur et utilisez vos AWS informations d'identification de développeur pour délivrer des informations d'identification à vos utilisateurs.

Vous disposez peut-être déjà d'un groupe d'utilisateurs Amazon Cognito qui fournit des services d'authentification et d'autorisation à votre application. Vous pouvez configurer votre groupe d'utilisateurs en tant que fournisseur d'identité (IdP) pour votre réserve d'identités. Lorsque vous le faites, vos utilisateurs peuvent s'authentifier via votre groupe d'utilisateurs IdPs, regrouper leurs demandes dans un jeton d'identité OIDC commun et échanger ce jeton contre des informations d'AWS identification. Votre utilisateur peut ensuite présenter ses informations d'identification dans une demande signée adressée à vos Services AWS.

Vous pouvez également présenter des champs standard authentifiés provenant de l'un de vos fournisseurs d'identité directement dans votre réserve d'identités. Amazon Cognito personnalise les demandes des utilisateurs émanant des fournisseurs SAML, OAuth et OIDC sous forme de demande d'API pour des informations d'identification à court terme. [AssumeRoleWithWebIdentity](#)

Les groupes d'utilisateurs Amazon Cognito sont comme des fournisseurs d'identité OIDC pour vos applications compatibles SSO. Les réserves d'identités font office de fournisseur d'identité AWS pour toute application dont les dépendances de ressources fonctionnent le mieux avec une autorisation IAM.

Les groupes d'identités Amazon Cognito prennent en charge les fournisseurs d'identité suivants :

- Fournisseurs publics : [Configuration de Login with Amazon en tant qu'IdP de pool d'identités](#), [Configuration de Facebook en tant qu'IdP de pool d'identités](#), [Configuration de Google en tant qu'IdP de pool d'identités](#), [Configuration de la connexion avec Apple en tant qu'IdP du pool d'identités](#), Twitter.
- [Groupes d'utilisateurs Amazon Cognito](#)
- [Configuration d'un fournisseur OIDC en tant qu'IdP de pool d'identités](#)
- [Configuration d'un fournisseur SAML en tant qu'IdP de pool d'identités](#)
- [Identités authentifiées par le développeur \(réserves d'identités\)](#)

Pour plus d'informations sur la disponibilité régionale des groupes d'identités Amazon Cognito, consultez [Disponibilité des services AWS par région](#).

Pour plus d'informations sur les groupes d'identités Amazon Cognito, consultez les rubriques suivantes.

Rubriques

- [Utilisation des groupes d'identités \(identités fédérées\)](#)
- [Concepts de groupes d'identités](#)
- [Bonnes pratiques de sécurité pour les groupes d'identités Amazon Cognito](#)
- [Utilisation d'attributs pour le contrôle d'accès](#)
- [Utilisation du contrôle d'accès basé sur les rôles](#)
- [Obtention des informations d'identification](#)
- [Accès aux AWS services](#)
- [Fournisseurs d'identité externes aux groupes d'identités](#)
- [Identités authentifiées par le développeur \(réserves d'identités\)](#)
- [Basculement d'utilisateurs non authentifiés à des utilisateurs authentifiés \(groupes d'identités\)](#)

Utilisation des groupes d'identités (identités fédérées)

Les pools d'identités Amazon Cognito fournissent des AWS informations d'identification temporaires aux utilisateurs invités (non authentifiés) et aux utilisateurs authentifiés et ayant reçu un jeton. Un groupe d'identités stocke les données d'identité utilisateur spécifiques à votre compte.

Pour créer un groupe d'identités dans la console

1. Connectez-vous à la [console Amazon Cognito](#) et sélectionnez Groupes d'identités.
2. Choisissez Créer un groupe d'identités.
3. Dans Configurer l'approbation du groupe d'identités, choisissez de configurer votre réserve d'identités en sélectionnant Accès authentifié, Accès invité ou les deux.
 - Si vous avez choisi Accès authentifié, sélectionnez un ou plusieurs types d'identité que vous souhaitez définir comme source des identités authentifiées dans votre réserve d'identités. Si vous configurez un fournisseur du développeur personnalisé, vous ne pouvez ni le modifier ni le supprimer après avoir créé votre réserve d'identités.

4. Dans Configurer les autorisations, choisissez un rôle IAM par défaut pour les utilisateurs authentifiés ou invités dans votre réserve d'identités.
 - a. Choisissez Créer un nouveau rôle IAM si vous souhaitez qu'Amazon Cognito crée un nouveau rôle pour vous avec des autorisations de base et une relation d'approbation avec votre réserve d'identités. Saisissez le nom du rôle IAM pour identifier votre nouveau rôle, par exemple `myidentitypool1_authenticatedrole`. Sélectionnez Afficher le document de stratégie pour passer en revue les autorisations qu'Amazon Cognito attribuera à votre nouveau rôle IAM.
 - b. Vous pouvez choisir d'utiliser un rôle IAM existant si vous avez déjà un rôle Compte AWS que vous souhaitez utiliser dans le vôtre. Vous devez configurer votre politique d'approbation de rôle IAM de manière à inclure `cognito-identity.amazonaws.com`. Configurez votre politique d'approbation de rôle pour autoriser Amazon Cognito à endosser le rôle uniquement quand il présente une preuve que la demande provient d'un utilisateur authentifié dans votre réserve d'identités spécifique. Pour plus d'informations, consultez [Autorisations et approbation de rôle](#).
5. Dans Connect identity providers, entrez les détails des fournisseurs d'identité (IdPs) que vous avez choisis dans Configurer la confiance du pool d'identités. Vous pouvez être invité à fournir des informations sur le client d'application OAuth, à choisir un groupe d'utilisateurs Amazon Cognito, à choisir un fournisseur d'identité IAM ou à saisir un identifiant personnalisé pour un fournisseur de développement.
 - a. Choisissez les paramètres de rôle pour chaque fournisseur d'identité. Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles. Avec un fournisseur d'identité de groupe d'utilisateurs Amazon Cognito, vous pouvez également sélectionner Choisir le rôle avec `preferred_role` dans les jetons. Pour plus d'informations sur le champ standard `cognito:preferred_role`, consultez [Affectation de valeurs de priorité à des groupes](#).
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.

- ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
 - b. Configurez Attributs de contrôle d'accès pour chaque fournisseur d'identité. L'option Attributs de contrôle d'accès mappe les champs standard utilisateur sur les [balises de principal](#) qu'Amazon Cognito applique à la session temporaire. Vous pouvez générer des politiques IAM pour filtrer l'accès des utilisateurs en fonction des balises que vous appliquez à leur session.
 - i. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - ii. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - iii. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
6. Dans Configurer les propriétés, saisissez un nom sous Nom du groupe d'identités.
7. Sous Authentification de base (classique), choisissez si vous souhaitez activer le flux de base. Lorsque le flux de base est actif, vous pouvez ignorer les sélections de rôles que vous avez effectuées pour vos IdPs et appeler [AssumeRoleWithWebIdentity](#) directement. Pour plus d'informations, consultez [Flux d'authentification de groupes d'identités \(identités fédérées\)](#).
8. Sous Balises, choisissez Ajouter une balise si vous souhaitez appliquer des [balises](#) à votre réserve d'identités.
9. Dans Vérifier et créer, confirmez les sélections que vous avez effectuées pour votre nouvelle réserve d'identités. Sélectionnez Modifier pour revenir dans l'assistant et modifier des paramètres. Lorsque vous avez terminé, sélectionnez Créer un groupe d'identités.

Rôles IAM d'utilisateur

Un rôle IAM définit les autorisations permettant à vos utilisateurs d'accéder à AWS des ressources, telles que [Amazon Cognito Sync](#). Les utilisateurs de votre application assument les rôles que vous créez. Vous pouvez spécifier différents rôles pour les utilisateurs authentifiés et ceux qui ne le sont pas. Pour plus d'informations sur les rôles IAM, consultez la section [Rôles IAM](#).

Identités authentifiées et non authentifiées

Les groupes d'identités Amazon Cognito prennent en charge les identités authentifiées et non authentifiées. Les identités authentifiées appartiennent aux utilisateurs authentifiés par tout fournisseur d'identité pris en charge. Les identités non authentifiées appartiennent généralement aux utilisateurs invités.

- Pour configurer les identités authentifiées avec un fournisseur de connexion public, consultez la section [Fournisseurs d'identité externes aux groupes d'identités](#).
- Pour configurer votre propre processus d'authentification backend, consultez la section [Identités authentifiées par le développeur \(réserves d'identités\)](#).

Activation ou désactivation de l'accès invité

L'accès invité aux pools d'identités Amazon Cognito (identités non authentifiées) fournit un identifiant et des informations d'AWS identification uniques aux utilisateurs qui ne s'authentifient pas auprès d'un fournisseur d'identité. Si votre application accepte les utilisateurs qui ne se connectent pas, vous pouvez activer l'accès pour les identités non authentifiées. Pour en savoir plus, veuillez consulter la section [Commencer à utiliser les pools d'identités Amazon Cognito](#).

Pour mettre à jour l'accès invité dans une réserve d'identités

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Localisez Accès Invité. Dans une réserve d'identités qui ne prend actuellement pas en charge l'accès invité, Statut a pour valeur Inactif.
 - a. Si Accès invité a pour valeur Actif et que vous souhaitez désactiver l'accès invité, sélectionnez Désactiver.
 - b. Si Accès Invité a pour valeur Inactif et que vous souhaitez activer l'accès invité, sélectionnez Modifier.
 - Choisissez un rôle IAM par défaut pour les utilisateurs invités dans votre réserve d'identités.
 - A. Choisissez Créer un nouveau rôle IAM si vous souhaitez qu'Amazon Cognito crée un nouveau rôle pour vous avec des autorisations de

base et une relation d'approbation avec votre réserve d'identités.

Saisissez le nom du rôle IAM pour identifier votre nouveau rôle, par exemple `myidentitypool_authenticatedrole`. Sélectionnez **Afficher le document de stratégie** pour passer en revue les autorisations qu'Amazon Cognito attribuera à votre nouveau rôle IAM.

- B. Vous pouvez choisir d'utiliser un rôle IAM existant si vous avez déjà un rôle Compte AWS que vous souhaitez utiliser dans le vôtre. Vous devez configurer votre politique d'approbation de rôle IAM de manière à inclure `cognito-identity.amazonaws.com`. Configurez votre politique d'approbation de rôle pour autoriser Amazon Cognito à endosser le rôle uniquement quand il présente une preuve que la demande provient d'un utilisateur authentifié dans votre réserve d'identités spécifique. Pour plus d'informations, consultez [Autorisations et approbation de rôle](#).
- C. Sélectionnez **Enregistrer les modifications**.
- D. Pour activer l'accès invité, sélectionnez **Activer** dans l'onglet **Accès utilisateur**.

Modification du rôle associé à un type d'identité

Chaque identité de votre groupe est authentifiée ou non authentifiée. Les identités authentifiées appartiennent aux utilisateurs authentifiés par un fournisseur de connexion public (groupes d'utilisateurs Amazon Cognito, Login with Amazon, Se connecter avec Apple, Facebook, Google, SAML ou fournisseurs OpenID Connect) ou par un fournisseur de développement (votre propre processus d'authentification backend). Les identités non authentifiées appartiennent généralement aux utilisateurs invités.

Un rôle est attribué à chaque type d'identité. Ce rôle est associé à une politique qui indique à Services AWS qui il peut accéder. Quand Amazon Cognito reçoit une requête, le service détermine le type d'identité, identifie le rôle qui lui est assigné et utilise la politique associée à ce rôle pour répondre. En modifiant une politique ou en attribuant un rôle différent à un type d'identité, vous pouvez contrôler le type d'identité auquel Services AWS un type d'identité peut accéder. Pour afficher ou modifier les politiques associées aux rôles de votre groupe d'identités, consultez la [console IAM AWS](#).

Pour modifier le rôle authentifié ou non authentifié par défaut de la réserve d'identités

1. Choisissez **Groupes d'identités** dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.

2. Choisissez l'onglet Accès utilisateur.
3. Localisez Accès invité ou Accès authentifié. Dans une réserve d'identités qui n'est pas actuellement configurée pour ce type d'accès, Statut a pour valeur Inactif. Tâche de sélection Modifier.
4. Choisissez un rôle IAM par défaut pour les utilisateurs invités ou authentifiés dans votre réserve d'identités.
 - a. Choisissez Créer un nouveau rôle IAM si vous souhaitez qu'Amazon Cognito crée un nouveau rôle pour vous avec des autorisations de base et une relation d'approbation avec votre réserve d'identités. Saisissez le nom du rôle IAM pour identifier votre nouveau rôle, par exemple `myidentitypool1_authenticatedrole`. Sélectionnez Afficher le document de stratégie pour passer en revue les autorisations qu'Amazon Cognito attribuera à votre nouveau rôle IAM.
 - b. Vous pouvez choisir d'utiliser un rôle IAM existant si vous avez déjà un rôle Compte AWS que vous souhaitez utiliser dans le vôtre. Vous devez configurer votre politique d'approbation de rôle IAM de manière à inclure `cognito-identity.amazonaws.com`. Configurez votre politique d'approbation de rôle pour autoriser Amazon Cognito à endosser le rôle uniquement quand il présente une preuve que la demande provient d'un utilisateur authentifié dans votre réserve d'identités spécifique. Pour plus d'informations, consultez [Autorisations et approbation de rôle](#).
5. Sélectionnez Enregistrer les modifications.

Modification des fournisseurs d'identité

Si vous autorisez vos utilisateurs à s'authentifier à l'aide de fournisseurs d'identité grand public (par exemple, les groupes d'utilisateurs Amazon Cognito, Login with Amazon, Se connecter avec Apple, Facebook ou Google), vous pouvez spécifier vos identifiants d'application dans la console des réserves d'identités Amazon Cognito (identités fédérées). Cette approche associe l'ID de l'application (fourni par le fournisseur de connexion public) à votre groupe d'identités.

Vous pouvez également configurer des règles d'authentification pour chaque fournisseur sur cette page. Chaque fournisseur autorisé jusqu'à 25 règles. Ces règles sont appliquées dans l'ordre d'enregistrement pour chaque fournisseur. Pour plus d'informations, consultez [Utilisation du contrôle d'accès basé sur les rôles](#).

⚠ Warning

Le remplacement de l'ID d'application du fournisseur d'identité lié dans votre réserve d'identités empêche les utilisateurs existants de s'authentifier auprès de cette réserve d'identités. Pour plus d'informations, consultez [Fournisseurs d'identité externes aux groupes d'identités](#).

Pour mettre à jour un fournisseur d'identité (IdP)

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Localisez Fournisseurs d'identité. Choisissez le fournisseur d'identité que vous souhaitez modifier. Si vous souhaitez ajouter un nouveau fournisseur d'identité, sélectionnez Ajouter un fournisseur d'identité.
 - Si vous avez choisi Ajouter un fournisseur d'identité, choisissez l'un des types d'identité que vous souhaitez ajouter.
4. Pour modifier l'ID d'application, choisissez Modifier dans Informations sur le fournisseur d'identité.
5. Pour modifier le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, choisissez Modifier dans Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles. Avec un fournisseur d'identité de groupe d'utilisateurs Amazon Cognito, vous pouvez également sélectionner Choisir le rôle avec preferred_role dans les jetons. Pour plus d'informations sur le champ standard cognito:preferred_role, consultez [Affectation de valeurs de priorité à des groupes](#).
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond.

- Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
- ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
6. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, choisissez Modifier dans Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
 7. Sélectionnez Enregistrer les modifications.

Supprimer un groupe d'identités

Vous ne pouvez pas annuler la suppression d'une réserve d'identités. Une fois que vous avez supprimé une réserve d'identités, toutes les applications et tous les utilisateurs qui en dépendent cessent de fonctionner.

Pour supprimer un groupe d'identités

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Activez la case d'option en regard de la réserve d'identités que vous souhaitez supprimer.
2. Sélectionnez Delete (Supprimer).
3. Saisissez ou collez le nom de votre réserve d'identités et sélectionnez Supprimer.

Warning

Lorsque vous sélectionnez le bouton Delete (Supprimer), vous supprimez définitivement votre groupe d'identités et toutes les données utilisateur qu'il contient. La suppression d'une

réserve d'identités entraîne l'arrêt du fonctionnement des applications et des autres services qui utilisaient cette réserve d'identités.

Supprimer une identité d'un groupe d'identités

Lorsque vous supprimez une identité d'une réserve d'identités, vous supprimez les informations d'identification qu'Amazon Cognito a stockées pour cet utilisateur fédéré. Lorsque votre utilisateur demande à nouveau des informations d'identification, il reçoit un nouvel ID d'identité si votre réserve d'identités fait toujours confiance à son fournisseur d'identité. Vous ne pouvez pas annuler cette opération.

Pour supprimer une identité

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Navigateur d'identité.
3. Cochez les cases en regard des identités que vous souhaitez supprimer et choisissez Supprimer. Confirmez que vous voulez supprimer ces identités et choisissez Supprimer.

Utilisation d'Amazon Cognito Sync avec des groupes d'identités

Amazon Cognito Sync est une Service AWS bibliothèque cliente qui permet de synchroniser les données utilisateur relatives aux applications sur tous les appareils. Amazon Cognito Sync peut synchroniser les données de profil utilisateur entre différents appareils mobiles et le web sans utiliser votre propre backend. Les bibliothèques client mettent en cache les données localement pour que votre application puisse lire et écrire des données quel que soit l'état de connectivité de l'appareil. Quand l'appareil est en ligne, vous pouvez synchroniser les données. Si vous configurez la synchronisation en mode push, vous pouvez avertir immédiatement les autres appareils qu'une mise à jour est disponible.

Gestion des jeux de données

Si vous avez implémenté la fonctionnalité Amazon Cognito Sync dans votre application, la console de groupes d'identités Amazon Cognito vous permet de créer et de supprimer manuellement des jeux de données et des enregistrements pour les identités individuelles. Toute modification apportée au

jeu de données ou aux enregistrements d'une identité dans la console de groupes d'identités Amazon Cognito n'est pas enregistrée tant que vous n'avez pas sélectionné Synchronize (Synchroniser) sur la console. La modification n'est pas visible par l'utilisateur final tant que l'identité n'a pas appelé Synchronize (Synchroniser). Les données en cours de synchronisation provenant d'autres appareils pour des identités individuelles sont visibles lorsque vous actualisez la page de liste de jeux de données d'une identité donnée.

Créer un jeu de données pour une identité

La synchronisation Amazon Cognito associe un jeu de données à une identité. Vous pouvez remplir votre jeu de données avec des informations d'identification au sujet de l'utilisateur que l'identité représente, puis synchroniser ces informations sur tous les appareils de l'utilisateur.

Pour ajouter un jeu de données et des enregistrements de jeu de données à une identité

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Navigateur d'identité.
3. Sélectionnez l'identité que vous souhaitez modifier.
4. Dans Jeux de données, choisissez Créer un jeu de données.
5. Saisissez le nom du jeu de données et sélectionnez Créer un jeu de données.
6. Si vous souhaitez ajouter des enregistrements à votre jeu de données, choisissez votre jeu de données dans les détails d'identité. Dans Enregistrements, sélectionnez Créer un enregistrement.
7. Saisissez une clé et une valeur pour votre enregistrement. Choisissez Confirmer. Répétez l'opération pour ajouter d'autres enregistrements.

Supprimer un jeu de données de données associé à une identité

Pour supprimer un jeu de données et ses enregistrements d'une identité

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Navigateur d'identité.
3. Sélectionnez l'identité qui contient le jeu de données que vous souhaitez supprimer.
4. Dans Jeux de données, choisissez la case d'option en regard du jeu de données que vous voulez supprimer.

5. Sélectionnez Delete (Supprimer). Passez en revue votre choix et sélectionnez à nouveau Supprimer.

Publier des données en bloc

La publication en bloc peut être utilisée pour exporter les données déjà stockées dans votre magasin Amazon Cognito Sync vers un flux Amazon Kinesis. Pour obtenir des instructions sur la publication en bloc de tous les flux, consultez la section [Flux Amazon Cognito](#).

Activation de la synchronisation push

Amazon Cognito suit automatiquement l'association entre l'identité et les appareils. La fonctionnalité de synchronisation push vous permet de veiller à ce que chaque instance d'une identité donnée soit informée en cas de modification des données d'identité. Chaque fois que le jeu de données d'une identité change, tous les dispositifs associés à cette identité reçoivent une notification push silencieuse les informant de ce changement.

Vous pouvez activer la synchronisation push dans la console Amazon Cognito.

Pour activer la synchronisation push

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Propriétés du groupe d'identités.
3. Dans Synchronisation push, sélectionnez Modifier
4. Sélectionnez Activer la synchronisation push avec votre réserve d'identités.
5. Choisissez l'une des applications de plateforme Amazon Simple Notification Service (Amazon SNS) que vous avez créées dans la Région AWS actuelle. Amazon Cognito publie des notifications push sur votre application de plateforme. Sélectionnez Créer une application de plateforme pour accéder à la console Amazon SNS et en créer une nouvelle.
6. Pour publier sur votre application de plateforme, Amazon Cognito endosse un rôle IAM dans votre Compte AWS. Choisissez Créer un nouveau rôle IAM si vous souhaitez qu'Amazon Cognito crée un nouveau rôle pour vous avec des autorisations de base et une relation d'approbation avec votre réserve d'identités. Saisissez le nom du rôle IAM pour identifier votre nouveau rôle, par exemple `myidentitypool_authenticatedrole`. Sélectionnez Afficher le document de stratégie pour passer en revue les autorisations qu'Amazon Cognito attribuera à votre nouveau rôle IAM.

7. Vous pouvez choisir d'utiliser un rôle IAM existant si vous avez déjà un rôle Compte AWS que vous souhaitez utiliser dans le vôtre. Vous devez configurer votre politique d'approbation de rôle IAM de manière à inclure `cognito-identity.amazonaws.com`. Configurez votre politique d'approbation de rôle pour autoriser Amazon Cognito à endosser le rôle uniquement quand il présente une preuve que la demande provient d'un utilisateur authentifié dans votre réserve d'identités spécifique. Pour plus d'informations, consultez [Autorisations et approbation de rôle](#).
8. Sélectionnez Enregistrer les modifications.

Configurer les flux Amazon Cognito

Les flux Amazon Cognito permettent aux développeurs de voir et de contrôler les données stockées dans Amazon Cognito. Les développeurs peuvent désormais configurer un flux Kinesis pour recevoir les événements de données. Amazon Cognito peut transmettre en temps réel une modification de jeu de données à un flux Kinesis que vous possédez. Pour obtenir des instructions sur la configuration des flux Amazon Cognito dans la console Amazon Cognito, consultez la section [Flux Amazon Cognito](#).

Configurer les événements Amazon Cognito

Amazon Cognito Events vous permet d'exécuter une AWS Lambda fonction en réponse à des événements importants dans Amazon Cognito Sync. Amazon Cognito Sync déclenche l'événement Sync Trigger lors de la synchronisation d'un jeu de données. Vous pouvez utiliser cet événement pour effectuer une action lorsqu'un utilisateur met à jour des données. Pour plus d'informations sur la configuration d'événements Amazon Cognito à partir de la console, consultez [Événements Amazon Cognito](#).

Pour en savoir plus AWS Lambda, consultez [AWS Lambda](#).

Concepts de groupes d'identités

Vous pouvez utiliser les groupes d'identités Amazon Cognito pour créer des identités uniques pour vos utilisateurs, et les authentifier avec des fournisseurs d'identité. Avec une identité, vous pouvez obtenir des informations d'AWS identification temporaires à privilèges limités pour accéder à d'autres. Services AWS Les groupes d'identités Amazon Cognito prennent en charge les fournisseurs d'identité publics, tels qu'Amazon, Apple, Facebook et Google, ainsi que les identités non authentifiées. Il prend également en charge les identités authentifiées par les développeurs, qui vous permettent d'inscrire et d'authentifier les utilisateurs par l'intermédiaire de votre processus d'authentification backend.

Pour plus d'informations sur la disponibilité régionale des groupes d'identités Amazon Cognito, consultez [Disponibilité des services AWS par région](#). Pour plus d'informations sur les concepts de groupes d'identités Amazon Cognito, consultez les rubriques suivantes.

Rubriques

- [Flux d'authentification de groupes d'identités \(identités fédérées\)](#)
- [Rôles IAM](#)
- [Autorisations et approbation de rôle](#)

Flux d'authentification de groupes d'identités (identités fédérées)

Amazon Cognito vous aide à créer des identifiants uniques que vos utilisateurs finaux peuvent utiliser sur divers appareils et plateformes. Amazon Cognito fournit également des informations d'identification temporaires à privilèges limités à votre application pour accéder aux ressources. AWS Cette page présente les principes de base de l'authentification dans Amazon Cognito, et explique le cycle de vie d'une identité au sein d'un groupe d'identités.

Flux d'authentification avec fournisseurs externes

Un utilisateur s'authentifiant avec Amazon Cognito suit un processus en plusieurs étapes pour amorcer ses informations d'identification. Amazon Cognito propose deux flux d'authentification distincts auprès de fournisseurs publics : le flux de base et le flux amélioré.

Une fois que vous avez terminé l'un de ces flux, vous pouvez accéder Services AWS aux autres conformément aux politiques d'accès de votre rôle. Par défaut, la [console Amazon Cognito](#) crée des rôles ayant accès au magasin Amazon Cognito Sync et à Amazon Mobile Analytics. Pour plus d'informations sur la façon d'octroyer des accès supplémentaires, consultez [Rôles IAM](#).

Les pools d'identités acceptent les artefacts suivants provenant des fournisseurs :

Fournisseur	Artefact d'authentification
Groupe d'utilisateurs Amazon Cognito	Jeton d'identification
OpenID Connect (OIDC)	Jeton d'identification
SAML 2.0	Assertion SAML

Fournisseur	Artefact d'authentification
Prestataire social	Jeton d'accès

Flux d'authentification amélioré (simplifié)

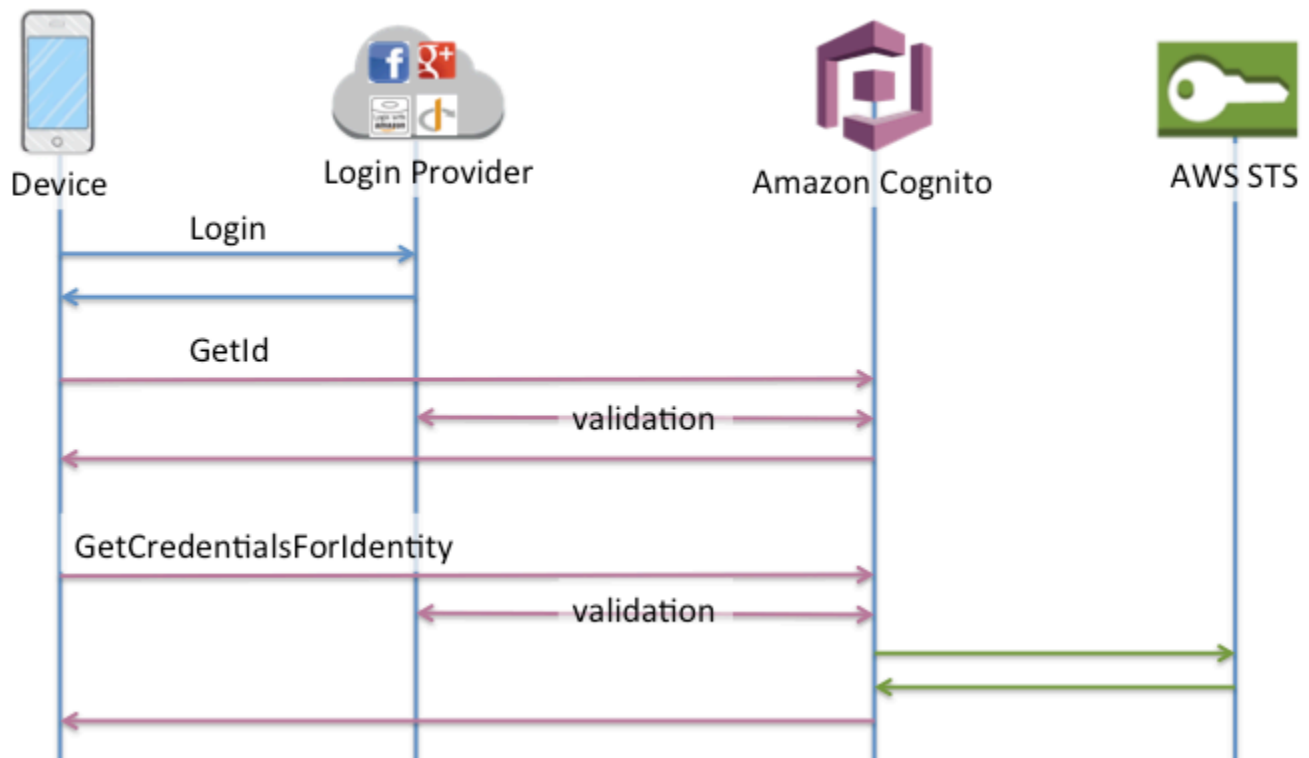
Lorsque vous utilisez le flux d'authentification amélioré, votre application présente d'abord une preuve d'authentification provenant d'un groupe d'utilisateurs Amazon Cognito autorisé ou d'un fournisseur d'identité tiers dans [GetId](#) une demande.

1. [Votre application présente une preuve d'authentification \(jeton Web JSON ou assertion SAML\) provenant d'un groupe d'utilisateurs Amazon Cognito autorisé ou d'un fournisseur d'identité tiers dans une demande GetID.](#)
2. Votre pool d'identités renvoie un identifiant d'identité.
3. Votre application associe l'identifiant d'identité à la même preuve d'authentification dans une [GetCredentialsForIdentity](#) demande.
4. Votre pool d'identités renvoie les AWS informations d'identification.
5. Votre application signe les demandes AWS d'API avec les informations d'identification temporaires.

L'authentification améliorée gère la logique de sélection des rôles IAM et de récupération des informations d'identification dans la configuration de votre pool d'identités. Vous pouvez configurer votre pool d'identités pour sélectionner un rôle par défaut, pour appliquer les principes du contrôle d'accès basé sur les attributs (ABAC) ou du contrôle d'accès basé sur les rôles (RBAC) à la sélection des rôles. Les AWS informations d'identification issues de l'authentification améliorée sont valides pendant une heure.

Ordre des opérations dans l'authentification améliorée

1. `GetId`
2. `GetCredentialsForIdentity`



Flux d'authentification basique (classique)

Lorsque vous utilisez le flux d'authentification de base,

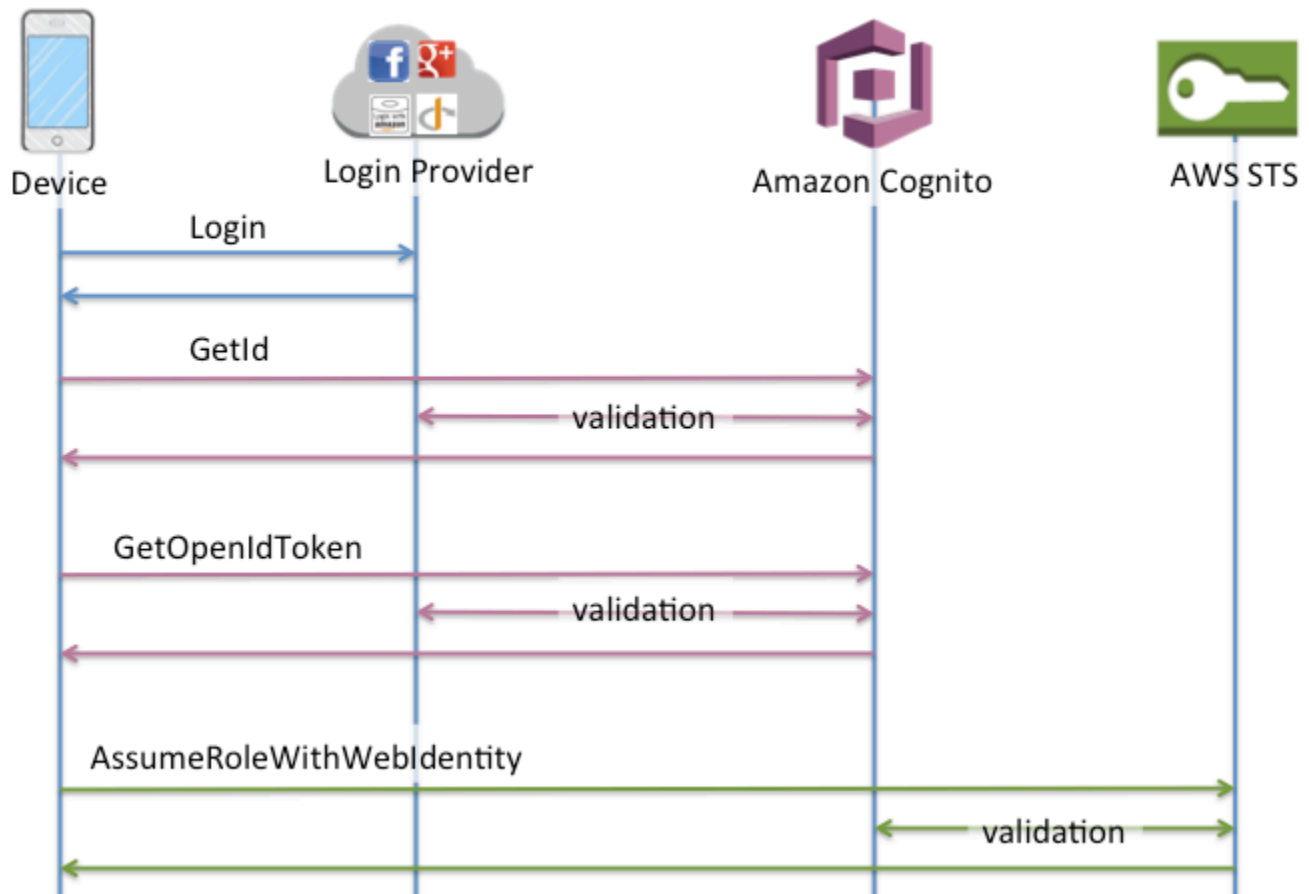
1. [Votre application présente une preuve d'authentification \(jeton Web JSON ou assertion SAML\) provenant d'un groupe d'utilisateurs Amazon Cognito autorisé ou d'un fournisseur d'identité tiers dans une demande GetID.](#)
2. Votre pool d'identités renvoie un identifiant d'identité.
3. Votre application associe l'identifiant d'identité à la même preuve d'authentification dans une [GetOpenIdToken](#) demande.
4. [GetOpenIdToken](#) renvoie un nouveau jeton OAuth 2.0 émis par votre pool d'identités.
5. Votre application présente le nouveau jeton dans une [AssumeRoleWithWebIdentity](#) demande.
6. AWS Security Token Service (AWS STS) renvoie les AWS informations d'identification.
7. Votre application signe les demandes AWS d'API avec les informations d'identification temporaires.

Le flux de travail de base vous offre un contrôle plus précis sur les informations d'identification que vous distribuez à vos utilisateurs. La demande `GetCredentialsForIdentity` du

flux d'authentification amélioré demande un rôle basé sur le contenu d'un jeton d'accès. La `AssumeRoleWithWebIdentity` demande dans le flux de travail classique donne à votre application une plus grande capacité à demander des informations d'identification pour tout AWS Identity and Access Management rôle que vous avez configuré avec une politique de confiance suffisante. Vous pouvez également demander une durée de session de rôle personnalisée.

Ordre des opérations dans l'authentification de base

1. `GetId`
2. `GetOpenIdToken`
3. `AssumeRoleWithWebIdentity`



Flux d'authentification des identités par les développeurs

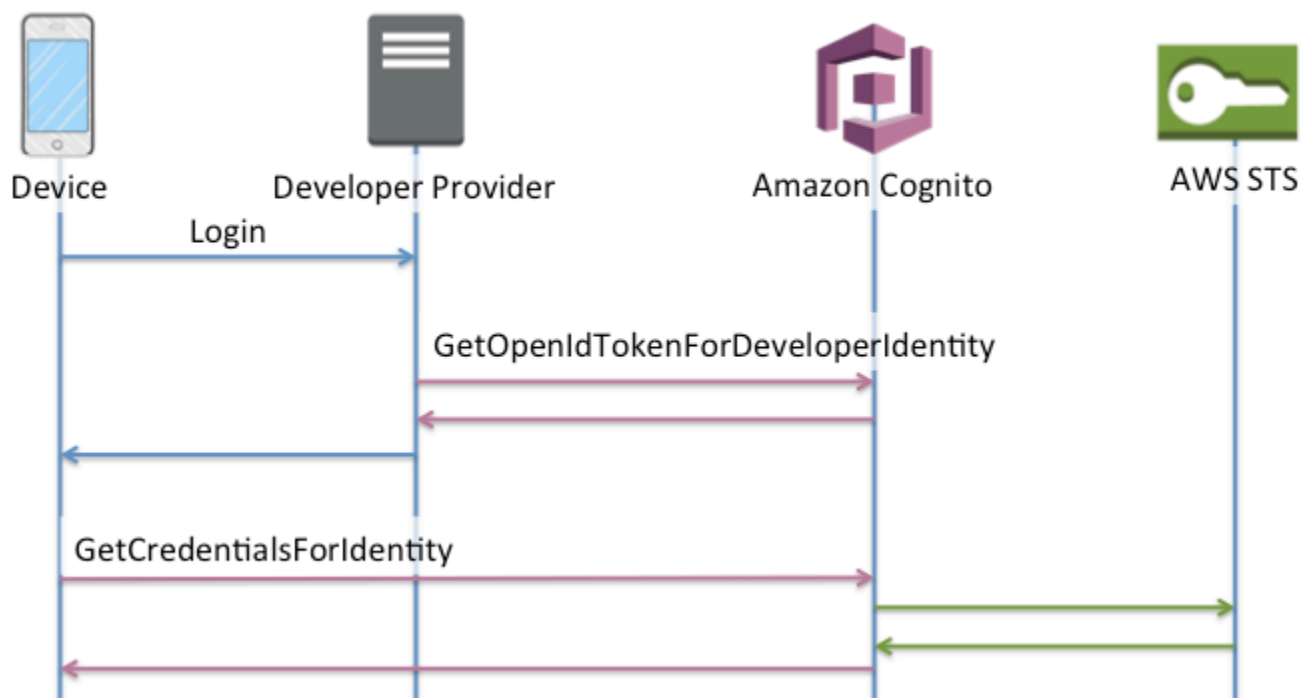
Lors de l'utilisation de [Identités authentifiées par le développeur \(réserves d'identités\)](#), le client utilise un autre flux d'authentification incluant du code en dehors d'Amazon Cognito afin de valider

l'utilisateur dans votre propre système d'authentification. Le code en dehors d'Amazon Cognito est indiqué tel quel.

Flux d'authentification amélioré

Ordre des opérations dans l'authentification améliorée auprès d'un fournisseur de développement

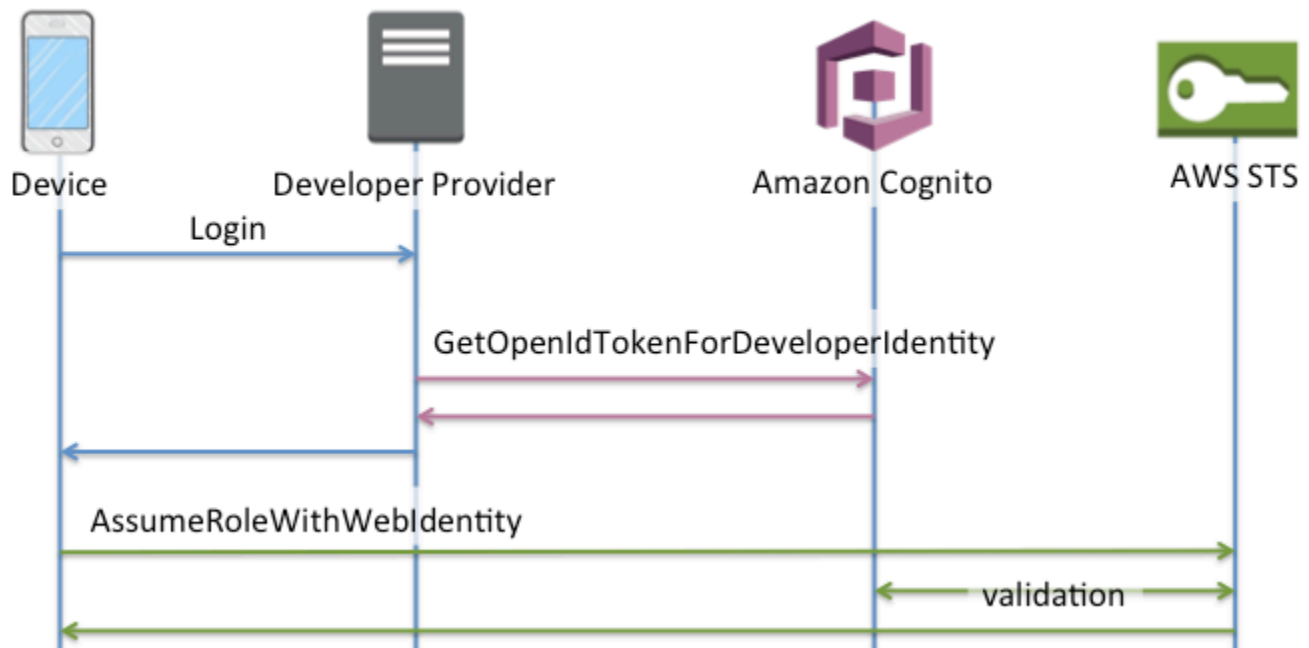
1. Se connecter via un fournisseur de développement (code en dehors d'Amazon Cognito)
2. Valider la connexion de l'utilisateur (code en dehors d'Amazon Cognito)
3. [GetOpenIdTokenForDeveloperIdentity](#)
4. [GetCredentialsForIdentity](#)



Ordre des opérations dans l'authentification de base auprès d'un fournisseur de développement

1. Implémentez une logique en dehors du pool d'identités pour vous connecter et générer un identifiant développeur-fournisseur.
2. Récupérez les informations d'identification stockées côté serveur. AWS
3. Envoyez l'identifiant du fournisseur de développement dans une demande d'[GetOpenIdTokenForDeveloperIdentity](#) API signée avec des AWS informations d'identification autorisées.

4. Demandez les informations d'identification de l'application auprès de [AssumeRoleWithWebIdentity](#).



Quel flux d'authentification utiliser ?

Le flux amélioré est le choix le plus sûr qui demande le moins d'efforts aux développeurs :

- Le flux amélioré réduit la complexité, la taille et le taux des demandes d'API.
- Votre application n'a pas besoin de faire de demandes d'API supplémentaires à AWS STS.
- Votre pool d'identités évalue vos utilisateurs en fonction des informations d'identification du rôle IAM qu'ils devraient recevoir. Vous n'avez pas besoin d'intégrer la logique de sélection des rôles dans votre client.

⚠ Important

Lorsque vous créez un nouveau pool d'identités, il est recommandé de ne pas activer l'authentification de base (classique) par défaut. Pour implémenter l'authentification de base, évaluez d'abord les relations de confiance de vos rôles IAM pour les identités Web. Intégrez ensuite la logique de sélection des rôles à votre client et protégez le client contre toute modification par les utilisateurs.

Le flux d'authentification de base délègue la logique de sélection des rôles IAM à votre application. Dans ce flux, Amazon Cognito valide la session authentifiée ou non authentifiée de votre utilisateur et émet un jeton que vous pouvez échanger contre des informations d'identification. AWS STS Les utilisateurs peuvent échanger les jetons issus de l'authentification de base contre tous les rôles IAM qui font confiance à votre pool d'identités et/ou à votre état amx authentifié/non authentifié.

De même, sachez que l'authentification des développeurs est un raccourci pour valider l'authentification du fournisseur d'identité. Amazon Cognito fait confiance aux AWS informations d'identification qui autorisent une [GetOpenIdTokenForDeveloperIdentity](#) demande sans validation supplémentaire du contenu de la demande. Sécurisez les secrets qui autorisent les développeurs à s'authentifier pour empêcher les utilisateurs d'y accéder.

Récapitulatif API

GetId

L'appel [GetId](#) d'API est le premier appel nécessaire pour établir une nouvelle identité dans Amazon Cognito.

Accès non authentifié

Amazon Cognito peut accorder un accès invité non authentifié dans vos applications. Si cette fonctionnalité est activée dans votre groupe d'identités, les utilisateurs peuvent à tout moment demander un nouvel ID d'identité via l'API Get Id. L'application est censée mettre en cache cet ID d'identité pour effectuer les appels suivants à Amazon Cognito. Les SDK AWS mobiles et le AWS SDK pour JavaScript in the Browser disposent de fournisseurs d'informations d'identification qui gèrent cette mise en cache pour vous.

Accès authentifié

Lorsque vous avez configuré votre application avec la prise en charge pour un fournisseur de connexion public (Facebook, Google+, Login with Amazon ou Se connecter avec Apple), les utilisateurs sont également en mesure de fournir des jetons (OAuth ou OpenID Connect) qui les identifient auprès de ces fournisseurs. Utilisé dans un appel à Get Id, Amazon Cognito crée une nouvelle identité authentifiée ou renvoie l'identité déjà associée à cette connexion spécifique. Amazon Cognito fait cela en validant le jeton auprès du fournisseur et en s'assurant que :

- Le jeton est valide et provient du fournisseur configuré.
- Le jeton n'a pas expiré.

- Le jeton correspond à l'identificateur d'application créé avec ce fournisseur (par exemple, ID d'application Facebook).
- Le jeton correspond à l'identifiant de l'utilisateur.

GetCredentialsForIdentity

L'[GetCredentialsForIdentity](#) API peut être appelée une fois que vous avez établi un identifiant d'identité. Cette opération est donc [AssumeRoleWithWebIdentity](#) fonctionnellement équivalente à un appel [GetOpenIdToken](#).

Pour qu'Amazon Cognito appelle `AssumeRoleWithWebIdentity` en votre nom, des rôles IAM doivent être associés à votre groupe d'identités. Vous pouvez le faire via la console Amazon Cognito ou manuellement via l'[SetIdentityPoolRoles](#) opération.

GetOpenIdToken

Effectuez une demande d'[GetOpenIdToken](#) API après avoir établi un identifiant d'identité. Mettez en cache les identifiants d'identité après votre première demande et démarrez les sessions de base (classiques) suivantes pour cette identité avec `GetOpenIdToken`.

La réponse à une demande d'API `GetOpenIdToken` est un jeton généré par Amazon Cognito. Vous pouvez soumettre ce jeton en tant que `WebIdentityToken` paramètre d'une [AssumeRoleWithWebIdentity](#) demande.

Avant de soumettre le jeton OpenID, vérifiez-le dans votre application. Vous pouvez utiliser les bibliothèques OIDC de votre kit SDK ou une bibliothèque comme [aws-jwt-verify](#) pour confirmer qu'Amazon Cognito a émis le jeton. L'ID de clé de signature, ou `kid`, du jeton OpenID est l'un de ceux répertoriés dans le [document jwks_uri](#) d'identité Amazon Cognito†. Ces clés sont susceptibles d'être modifiées. Votre fonction qui vérifie les jetons d'identité Amazon Cognito doit régulièrement mettre à jour sa liste de clés à partir du document `jwks_uri`. Amazon Cognito définit la durée d'actualisation dans l'en-tête de réponse `cache-control` de `jwks_uri`, actuellement définie avec un paramètre `max-age` de 30 jours.

Accès non authentifié

Pour obtenir un jeton pour une identité non authentifiée, vous avez uniquement besoin de l'ID d'identité. Il n'est pas possible d'obtenir un jeton non authentifié pour les identités authentifiées ou les identités que vous avez désactivées.

Accès authentifié

Si vous avez une identité authentifiée, vous devez transmettre au moins un jeton valide pour une connexion déjà associée à cette identité. Tous les jetons transmis au cours de l'appel

`GetOpenIdToken` doivent passer par le processus de validation mentionné ci-dessus. Si un jeton échoue, l'appel n'aboutit pas. La réponse à l'appel `GetOpenIdToken` inclut également l'ID d'identité, car l'ID d'identité que vous fournissez n'est pas toujours celui qui est renvoyé.

Liaison de connexions

Si vous soumettez un jeton pour une connexion qui n'est pas déjà associée à une identité, la connexion est considéré comme étant « liée » à l'identité associée. Vous ne pouvez lier qu'une connexion par fournisseur public. Toute tentative de liaison de plusieurs connexions avec un fournisseur public entraîne une réponse d'erreur `ResourceConflictException`. Si une connexion est simplement liée à une identité existante, l'ID d'identité renvoyé à partir de `GetOpenIdToken` est la même que celle que vous avez transmise.

Fusion d'identités

Si vous transmettez un jeton pour une connexion qui n'est pas liée à l'identité donnée, mais qui est liée à une autre identité, les deux identités sont fusionnées. Puis, une identité devient le parent/propriétaire de toutes les connexions associées, tandis que l'autre est désactivée. Dans ce cas, l'ID d'identité du propriétaire/parent est renvoyée. Vous devez mettre à jour votre cache local si cette valeur diffère. Les fournisseurs des SDK AWS mobiles ou du AWS SDK du navigateur exécutent cette opération pour vous. JavaScript

`GetOpenIdTokenForDeveloperIdentity`

L'[GetOpenIdTokenForDeveloperIdentity](#) opération remplace l'utilisation de [GetIdet](#) [GetOpenIdToken](#) depuis l'appareil lors de l'utilisation d'identités authentifiées par le développeur. Dans la mesure où votre application signe les demandes adressées à cette opération d'API avec des AWS informations d'identification, Amazon Cognito est sûr que l'identifiant utilisateur fourni dans la demande est valide. L'authentification du développeur remplace la validation par jeton effectuée par Amazon Cognito auprès de fournisseurs externes.

La charge utile de cette API inclut une `logins` carte. Cette carte doit contenir la clé de votre fournisseur de développement et une valeur servant d'identifiant pour l'utilisateur de votre système. Si l'identifiant de l'utilisateur n'est pas encore lié à une identité existante, Amazon Cognito crée une nouvelle identité et renvoie le nouvel ID d'identité ainsi qu'un jeton `OpenID Connect` pour cette identité. Si l'identifiant de l'utilisateur est déjà lié, Amazon Cognito renvoie l'ID d'identité préexistant et un jeton `OpenID Connect`. Mettez en cache les identifiants d'identité de développeur après votre première demande et démarrez les sessions de base (classiques) suivantes pour cette identité avec `GetOpenIdTokenForDeveloperIdentity`.

La réponse à une demande d'API `GetOpenIdTokenForDeveloperIdentity` est un jeton généré par Amazon Cognito. Vous pouvez soumettre ce jeton en tant que paramètre `WebIdentityToken` dans une demande `AssumeRoleWithWebIdentity`.

Avant de soumettre le jeton OpenID Connect, vérifiez-le dans votre application. Vous pouvez utiliser les bibliothèques OIDC de votre kit SDK ou une bibliothèque comme [aws-jwt-verify](#) pour confirmer qu'Amazon Cognito a émis le jeton. L'ID de clé de signature, ou `kid`, du jeton OpenID Connect est l'un de ceux répertoriés dans le [document `jwtks_uri`](#) d'identité Amazon Cognito. Ces clés sont susceptibles d'être modifiées. Votre fonction qui vérifie les jetons d'identité Amazon Cognito doit régulièrement mettre à jour sa liste de clés à partir du document `jwtks_uri`. Amazon Cognito définit la durée d'actualisation dans l'en-tête de réponse `cache-control` de `jwtks_uri`, actuellement définie avec un paramètre `max-age` de 30 jours.

Liaison de connexions

Comme avec les fournisseurs externes, fournir des connexions supplémentaires qui ne sont pas encore associées à une identité entraîne implicitement la liaison de ces connexions à cette identité. Si vous liez une connexion de fournisseur externe à une identité, l'utilisateur peut utiliser le flux d'authentification du fournisseur externe avec ce fournisseur. Toutefois, il ne peut pas utiliser le nom de votre fournisseur de développement dans la carte des connexions lorsqu'il appelle `GetId` ou `GetOpenIdToken`.

Fusion d'identités

Grâce aux identités authentifiées par les développeurs, Amazon Cognito prend en charge à la fois la fusion implicite et la fusion explicite via l'API. [MergeDeveloperIdentities](#) La fusion explicite vous permet de marquer deux identités avec des identifiants utilisateur dans votre système comme une seule identité. Si vous fournissez les identifiants utilisateur source et de destination, Amazon Cognito les fusionne. La prochaine fois que vous demandez un jeton OpenID Connect pour l'un de ces identifiants utilisateur, le même ID d'identité est renvoyé.

AssumeRoleWithWebIdentity

Une fois que vous avez un jeton OpenID Connect, vous pouvez l'échanger contre des informations d'AWS identification temporaires via la demande d'[AssumeRoleWithWebIdentity](#) API adressée à AWS Security Token Service (AWS STS).

Comme il n'y a aucune restriction sur le nombre d'identités que vous pouvez créer, il est important de comprendre les autorisations que vous accordez à vos utilisateurs. Configurez différents rôles IAM pour votre application : un pour les utilisateurs non authentifiés et un pour les utilisateurs authentifiés. La console Amazon Cognito peut créer des rôles par défaut lorsque vous configurez

vos pools d'identités pour la première fois. Aucune autorisation n'est effectivement accordée à ces rôles. Modifiez-les en fonction de vos besoins.

En savoir plus sur [Autorisations et approbation de rôle](#).

† Le document [jwks_uri](#) d'identité Amazon Cognito par défaut contient des informations sur les clés utilisées pour signer les jetons pour les groupes d'identités dans la plupart des Régions AWS. Les régions suivantes ont des documents `jwks_uri` différents.

Amazon Cognito Identity JSON web key URIs in other Régions AWS

Région AWS	Chemin d'accès au document <code>jwks_uri</code>
AWS GovCloud (US-Ouest)	<code>https://cognito-identity.us-gov-west-1.amazonaws.com/.well-known/jwks_uri</code>
Chine (Beijing)	<code>https://cognito-identity.cn-north-1.amazonaws.com.cn/.well-known/jwks_uri</code>
Régions optionnelles comme l'Europe (Milan) et l'Afrique (Le Cap)	<code>https://cognito-identity.<i>Region</i>.amazonaws.com/.well-known/jwks_uri</code>

Vous pouvez également extrapoler le document `jwks_uri` provenant de l'émetteur, ou `iss`, que vous recevez dans le jeton OpenID provenant d'Amazon Cognito. Le point de terminaison de découverte standard OIDC `<issuer>/.well-known/openid-configuration` répertorie un chemin d'accès vers le document `jwks_uri` pour votre jeton.

Rôles IAM

Lors de la création d'un groupe d'identités, vous êtes invité à mettre à jour les rôles IAM qu'assument vos utilisateurs. Les rôles IAM fonctionnent comme suit : lorsqu'un utilisateur se connecte à votre application, Amazon Cognito génère des informations d'identification AWS temporaires pour l'utilisateur. Ces informations d'identification temporaires sont associées à un rôle IAM spécifique. Avec le rôle IAM, vous pouvez définir un ensemble d'autorisations pour accéder à vos AWS ressources.

Vous pouvez spécifier des rôles IAM par défaut pour les utilisateurs authentifiés et ceux qui ne le sont pas. Vous pouvez également définir des règles pour choisir le rôle pour chaque utilisateur en fonction de demandes dans le jeton d'identification de l'utilisateur. Pour plus d'informations, consultez [Utilisation du contrôle d'accès basé sur les rôles](#).

Par défaut, la console Amazon Cognito crée des rôles IAM donnant accès à Amazon Mobile Analytics et à Amazon Cognito Sync. Vous pouvez également choisir d'utiliser des rôles IAM existants.

Modifiez les rôles IAM pour autoriser ou restreindre l'accès à d'autres services. Pour cela, [connectez-vous à la console IAM](#). Sélectionnez ensuite Roles (Rôles), puis sélectionnez un rôle. Les politiques associées au rôle sélectionné sont répertoriées dans l'onglet Permissions. Vous pouvez personnaliser une stratégie d'accès en sélectionnant le lien correspondant Manage Policy (Gérer la politique). Pour plus d'informations sur l'utilisation et la définition des politiques, consultez la section [Présentation des politiques IAM](#).

Note

La bonne pratique consiste à définir des politiques qui suivent le principe du moindre privilège. En d'autres termes, les politiques incluent uniquement les autorisations dont les utilisateurs ont besoin pour effectuer leurs tâches. Pour plus d'informations, consultez [Accorder le privilège le plus faible](#) dans le Guide de l'utilisateur IAM.

N'oubliez pas que les identités non authentifiées sont assumées par des utilisateurs qui ne se connectent pas à votre application. En général, les autorisations que vous attribuez pour les identités non authentifiées doivent être plus restrictives que celles créées pour les identités authentifiées.

Rubriques

- [Configurer une politique d'approbation](#)
- [politiques d'accès](#)

Configurer une politique d'approbation

Amazon Cognito utilise des rôles IAM pour générer des informations d'identification temporaires pour les utilisateurs de votre application. L'accès aux autorisations est contrôlé par les relations d'approbation d'un rôle. En savoir plus sur [Autorisations et approbation de rôle](#).

Le jeton présenté AWS STS est généré par un pool d'identités, qui traduit un pool d'utilisateurs, un jeton social ou un jeton de fournisseur OIDC, ou une assertion SAML, en son propre jeton. Le jeton de la réserve d'identités contient une réclamation aud qui est l'ID de la réserve d'identités.

L'exemple de politique de confiance des rôles suivant permet au principal du service fédéré `cognito-identity.amazonaws.com` d'appeler l' AWS STS API `AssumeRoleWithWebIdentity`. La demande n'aboutira que si le jeton de la réserve d'identités figurant dans la demande d'API comporte les demandes suivantes.

1. Une demande aud concernant la région `us-west-2:abcdefg-1234-5678-910a-0e8443553f95` de l'ID de la réserve d'identités.
2. Une demande `amr` de `authenticated` qui est ajoutée lorsque l'utilisateur s'est connecté et n'est pas un utilisateur invité.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-
west-2:abcdefg-1234-5678-910a-0e8443553f95"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
```

Politiques de confiance pour les rôles IAM dans l'authentification de base (classique)

Vous devez appliquer au moins une condition qui limite les politiques de confiance pour les rôles que vous utilisez avec les groupes d'identités. Lorsque vous créez ou mettez à jour des politiques

d'approbation des rôles pour les pools d'identités, IAM renvoie un message d'erreur si vous essayez d'enregistrer vos modifications sans au moins une clé de condition limitant les identités sources. AWS STS n'autorise pas les [AssumeRoleWithWebIdentity](#) opérations entre comptes, qu'il s'agisse de pools d'identités ou de rôles IAM dépourvus de ce type de condition.

Cette rubrique inclut plusieurs conditions qui limitent les identités sources pour les pools d'identités. Pour une liste complète, voir [Clés disponibles pour la fédération d'identité AWS Web](#).

Dans le cadre de l'authentification de base ou classique avec un pool d'identités, vous pouvez assumer n'importe quel rôle IAM AWS STS s'il dispose de la bonne politique de confiance. Les rôles IAM pour les pools d'identités Amazon Cognito font confiance au `cognito-identity.amazonaws.com` principal du service pour assumer ce rôle. Cette configuration n'est pas suffisante pour protéger vos rôles IAM contre tout accès involontaire aux ressources. Les rôles de ce type doivent appliquer une condition supplémentaire à la politique de confiance des rôles. Vous ne pouvez pas créer ou modifier des rôles pour des pools d'identités sans au moins l'une des conditions suivantes.

`cognito-identity.amazonaws.com:aud`

Limite le rôle aux opérations provenant d'un ou de plusieurs pools d'identités. Amazon Cognito indique le pool d'identités source dans la `aud` réclamation dans le jeton du pool d'identités.

`cognito-identity.amazonaws.com:amr`

Limite le rôle aux utilisateurs `authenticated` ou aux utilisateurs `unauthenticated` (invités). Amazon Cognito indique l'état d'authentification de la `amr` réclamation dans le jeton du pool d'identités.

`cognito-identity.amazonaws.com:sub`

Limite le rôle à un ou plusieurs utilisateurs par UUID. Cet UUID est l'ID d'identité de l'utilisateur dans le pool d'identités. Cette valeur n'est pas celle `sub` du fournisseur d'identité d'origine de l'utilisateur. Amazon Cognito indique cet UUID dans la `sub` réclamation figurant dans le jeton du pool d'identités.

L'authentification à flux amélioré nécessite que le rôle IAM soit identique à celui Compte AWS du pool d'identités, mais ce n'est pas le cas dans le cas de l'authentification de base.

Des considérations supplémentaires s'appliquent aux pools d'identités Amazon Cognito qui assument des rôles IAM [entre comptes](#). Les politiques de confiance de ces rôles doivent accepter le principe du `cognito-identity.amazonaws.com` service et contenir la `cognito-`

`identity.amazonaws.com:aud` condition spécifique. Pour empêcher tout accès involontaire à vos AWS ressources, la clé de `aud` condition limite le rôle aux utilisateurs des pools d'identités figurant dans la valeur de la condition.

Le jeton émis par un pool d'identités pour une identité contient des informations sur l'origine Compte AWS du pool d'identités. Lorsque vous présentez un jeton de pool d'identités dans une AWS STS demande d'[AssumeRoleWithWebIdentity](#) API, vérifiez si le pool d'identités d'origine est Compte AWS identique au rôle IAM. S'il AWS STS détermine que la demande est intercomptes, il vérifie si la politique de confiance des rôles comporte une `aud` condition. L'appel `assume-rôle` échoue si aucune condition de ce type n'est présente dans la politique d'approbation des rôles. Si la demande n'est pas intercomptes, cette restriction AWS STS n'est pas appliquée. La meilleure pratique consiste à toujours appliquer une condition de ce type aux politiques de confiance des rôles de votre pool d'identités.

Conditions supplémentaires de la politique de confiance

Réutilisation des rôles entre les groupes d'identités

Pour réutiliser un rôle dans plusieurs groupes d'identités qui partagent un jeu d'autorisations commun, procédez comme suit afin d'inclure les groupes d'identités requis :

```
"StringEquals": {
  "cognito-identity.amazonaws.com:aud": [
    "us-east-1:12345678-abcd-abcd-abcd-123456790ab",
    "us-east-1:98765432-dcba-dcba-dcba-123456790ab"
  ]
}
```

Restriction de l'accès à des identités spécifiques

Pour créer une politique limitée à un ensemble spécifique d'utilisateurs de l'application, vérifiez la valeur `:cognito-identity.amazonaws.com:sub`

```
"StringEquals": {
  "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-abcd-abcd-
abcd-123456790ab",
  "cognito-identity.amazonaws.com:sub": [
    "us-east-1:12345678-1234-1234-1234-123456790ab",
    "us-east-1:98765432-1234-1234-1243-123456790ab"
  ]
}
```

Restriction de l'accès à des fournisseurs spécifiques

Pour créer une politique limitée aux utilisateurs qui se sont connectés avec un fournisseur spécifique (votre propre fournisseur de connexion, par exemple), vérifiez la valeur `:cognito-identity.amazonaws.com:amr`

```
"ForAnyValue:StringLike": {  
  "cognito-identity.amazonaws.com:amr": "login.myprovider.myapp"  
}
```

Par exemple, une application qui n'autorise que Facebook présente la clause AMR suivante :

```
"ForAnyValue:StringLike": {  
  "cognito-identity.amazonaws.com:amr": "graph.facebook.com"  
}
```

politiques d'accès

Les autorisations que vous attachez à un rôle s'appliquent à tous les utilisateurs qui assument ce rôle. Pour partitionner l'accès de vos utilisateurs, utilisez des conditions et des variables de politique. Pour en savoir plus, consultez [Éléments des politiques IAM : variables et balises](#). Vous pouvez utiliser la condition `sub` pour limiter les actions aux ID d'identité Amazon Cognito dans vos stratégies d'accès. Utilisez cette option avec prudence, surtout avec les identités non authentifiées, qui ne disposent pas d'un ID d'utilisateur constant. Pour plus d'informations sur les variables de politique IAM pour la fédération Web avec Amazon Cognito, [consultez la section Clés contextuelles IAM AWS STS et conditionnelles](#) dans AWS Identity and Access Management le Guide de l'utilisateur.

Pour une sécurité accrue, Amazon Cognito applique une politique de limitation de portée aux informations d'identification que vous attribuez à vos utilisateurs non authentifiés dans le [flux amélioré](#), à l'aide de `GetCredentialsForIdentity`. La politique de limitation de portée ajoute une [Politique de session en ligne](#) et une [AWS politique de session gérée](#) aux politiques IAM que vous appliquez à votre rôle non authentifié. Puisque vous devez accorder l'accès à la fois dans les politiques IAM de votre rôle et dans les politiques de session, la politique de limitation de portée limite l'accès des utilisateurs aux services autres que ceux figurant dans la liste suivante.

Note

Dans le flux de base (classique), vous créez votre propre demande d'API [AssumeRoleWithWebIdentity](#), et vous pouvez appliquer ces restrictions à la demande. Pour

être en phase avec les bonnes pratiques de sécurité, évitez d'attribuer à des utilisateurs non authentifiés des autorisations supérieures à celles de cette politique de limitation de portée.

Amazon Cognito empêche également les utilisateurs authentifiés et non authentifiés d'envoyer des demandes d'API aux réserves d'identités Amazon Cognito et à Amazon Cognito Sync. D'autres Services AWS peuvent imposer des restrictions sur l'accès aux services à partir d'identités Web.

Dans le cas d'une demande réussie avec le flux amélioré, Amazon Cognito effectue une demande d'API `AssumeRoleWithWebIdentity` en arrière-plan. Parmi les paramètres de cette demande, voici ceux qu'Amazon Cognito inclut.

1. L'ID d'identité de votre utilisateur.
2. L'ARN du rôle IAM que votre utilisateur souhaite assumer.
3. Un paramètre `policy` qui ajoute une politique de session en ligne.
4. `PolicyArns.member.N` Paramètre dont la valeur est une politique AWS gérée qui accorde des autorisations supplémentaires sur Amazon CloudWatch.

Services auxquels les utilisateurs non authentifiés peuvent accéder

Lorsque vous utilisez le flux amélioré, les politiques de limitation d'étendue qu'Amazon Cognito applique à la session de votre utilisateur l'empêchent d'utiliser d'autres services que ceux répertoriés dans le tableau suivant. Pour un sous-ensemble de services, seules des actions spécifiques sont autorisées.

Catégorie	Service
Analyse	Amazon Data Firehose
	Service géré Amazon pour Apache Flink
Intégration des applications	Amazon Simple Queue Service
AR et VR	Amazon Sumerian ¹
Applications métier	Amazon Mobile Analytics
	Amazon Simple Email Service

Catégorie	Service
Calcul	AWS Lambda
Chiffrement et PKI	AWS Key Management Service ¹
Base de données	Amazon DynamoDB Amazon SimpleDB
Web et mobile front-end	AWS AppSync Amazon Location Service Amazon Simple Notification Service Amazon Pinpoint
Développement de jeux	Amazon GameLift
Internet des objets (IoT)	AWS IoT
Machine Learning (apprentissage automatique)	Amazon CodeWhisperer Amazon Comprehend Amazon Lex Amazon Machine Learning Amazon Personalize Amazon Polly Amazon Rekognition Amazon SageMaker ¹ Amazon Textract ¹ Amazon Transcribe Amazon Translate

Catégorie	Service
Gestion et gouvernance	Amazon CloudWatch Amazon CloudWatch Logs
Mise en réseau et diffusion de contenu	Amazon API Gateway
Sécurité, identité et conformité	Groupes d'utilisateurs Amazon Cognito
Stockage	Amazon Simple Storage Service

¹ Services AWS Dans le tableau suivant, la politique intégrée autorise un sous-ensemble d'actions. Le tableau affiche les actions disponibles pour chaque politique.

Service AWS	Autorisations maximales pour les utilisateurs du flux amélioré non authentifiés
AWS Key Management Service	Encrypt Decrypt ReEncrypt GenerateDataKey
Amazon SageMaker	InvokeEndpoint
Amazon Textract	DetectDocumentText AnalyzeDocument
Amazon Sumerian	View*

Pour autoriser l'accès Services AWS au-delà de cette liste, activez le flux d'authentification de base (classique) dans votre pool d'identités. Si vos utilisateurs obtiennent des erreurs `NotAuthorizedException` de Services AWS autorisés par les politiques affectées au rôle IAM pour les utilisateurs non authentifiés, déterminez si vous pouvez éliminer ces services de votre cas d'utilisation. Si ce n'est pas possible, basculez sur le flux de base.

La politique de session en ligne

La politique de session intégrée empêche les autorisations effectives de votre utilisateur d'inclure l'accès à des autorisations Services AWS extérieures à celles figurant dans la liste suivante. Vous devez également leur accorder des autorisations Services AWS dans les politiques que vous appliquez au rôle IAM de l'utilisateur. Les autorisations effectives d'un utilisateur pour une session avec rôle assumé se trouvent à l'intersection des politiques attribuées à son rôle et de sa politique de session. Pour en savoir plus, consultez [Politiques de session](#) dans le Guide de l'utilisateur AWS Identity and Access Management .

Amazon Cognito ajoute la politique en ligne suivante aux sessions de vos utilisateurs dans les Régions AWS qui sont activées par défaut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:*",
        "logs:*",
        "dynamodb:*",
        "kinesis:*",
        "mobileanalytics:*",
        "s3:*",
        "ses:*",
        "sns:*",
        "sqs:*",
        "lambda:*",
        "machinelearning:*",
        "execute-api:*",
        "iot:*",
        "gamelift:*",
        "scs:*",
        "cognito-identity:*",
        "cognito-idp:*",
        "lex:*",
        "polly:*",
        "comprehend:*",
        "translate:*",
        "transcribe:*",
        "rekognition:*
```

```

        "mobiletargeting:*",
        "firehose:*",
        "appsync:*",
        "personalize:*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "sagemaker:InvokeEndpoint",
        "cognito-sync:*",
        "sumerian:View*",
        "codewhisperer:*",
        "textract:DetectDocumentText",
        "textract:AnalyzeDocument",
        "sdb:*"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

Pour toutes les autres régions, la politique de limitation d'étendue intégrée inclut tout ce qui est répertorié dans les régions par défaut, à l'exception des déclarations Action suivantes.

```

    "cognito-sync:*",
    "sumerian:View*",
    "codewhisperer:*",
    "textract:DetectDocumentText",
    "textract:AnalyzeDocument",
    "sdb:*"

```

La politique de session AWS gérée

Amazon Cognito limite également la portée des autorisations des utilisateurs non authentifiés grâce à la politique gérée AWS `AmazonCognitoUnAuthedIdentitiesSessionPolicy` à vos utilisateurs non authentifiés dans le flux amélioré. Vous devez également accorder cette autorisation dans les politiques que vous attachez à votre rôle IAM non authentifié.

La politique gérée `AmazonCognitoUnAuthedIdentitiesSessionPolicy` contient les autorisations suivantes.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "rum:PutRumEvents",
      "polly:*",
      "comprehend:*",
      "translate:*",
      "transcribe:*",
      "rekognition:*",
      "mobiletargeting:*",
      "firehose:*",
      "personalize:*",
      "sagemaker:InvokeEndpoint"
    ],
    "Resource": "*"
  }]
}
```

Exemples de politiques d'accès

Dans cette section, vous trouverez des exemples de stratégies d'accès Amazon Cognito qui accordent aux utilisateurs les autorisations minimales nécessaires pour exécuter une opération spécifique. Vous pouvez limiter davantage les autorisations pour un ID d'identité donné en utilisant des variables de politique dans la mesure du possible. Par exemple, avec `${cognito-identity.amazonaws.com:sub}`. Pour plus d'informations, consultez [Présentation de l'authentification Amazon Cognito 3e partie : Rôles et politiques](#) sur le blog AWS Mobile.

Note

La bonne pratique en matière de sécurité consiste à inclure uniquement les autorisations dont les utilisateurs ont besoin pour exécuter leurs tâches. Cela signifie que vous devez toujours essayer de définir l'accès à une identité individuelle pour les objets dans la mesure du possible.

Octroi à une identité d'un accès en lecture à un objet individuel dans Amazon S3

La stratégie d'accès suivante accorde des autorisations de lecture à une identité pour récupérer un seul objet à partir d'un compartiment S3 donné.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket/assets/my_picture.jpg"]
    }
  ]
}
```

Octroi à une identité d'un accès en lecture et en écriture à des chemins spécifiques à l'identité dans Amazon S3

La stratégie d'accès suivante accorde des autorisations de lecture et d'écriture pour accéder à un préfixe spécifique « folder » d'un compartiment S3 en mappant le préfixe à la variable `${cognito-identity.amazonaws.com:sub}`.

Avec cette stratégie, une identité telle que `us-east-1:12345678-1234-1234-1234-123456790ab` insérée via `${cognito-identity.amazonaws.com:sub}` est en mesure d'obtenir, de placer et de répertorier des objets dans `arn:aws:s3:::mybucket/us-east-1:12345678-1234-1234-1234-123456790ab`. Toutefois, l'identité ne se verra pas accorder l'accès à d'autres objets dans `arn:aws:s3:::mybucket`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket"],
      "Condition": {"StringLike": {"s3:prefix": ["${cognito-identity.amazonaws.com:sub}/*"]}}
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ]
    }
  ]
}
```

```
    ],
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::mybucket/${cognito-identity.amazonaws.com:sub}/*"]
  }
]
}
```

Attribution à des identités d'un accès précis à Amazon DynamoDB

La stratégie d'accès suivante fournit un contrôle précis des accès aux ressources DynamoDB à l'aide de variables d'environnement Amazon Cognito. Ces variables accordent l'accès aux éléments dans DynamoDB par ID d'identité. Pour de plus amples informations, veuillez consulter [Utilisation des conditions de politique IAM pour un contrôle précis des accès](#) dans le Manuel du développeur Amazon DynamoDB.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb>DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
      "Resource": [
        "arn:aws:dynamodb:us-west-2:123456789012:table/MyTable"
      ],
      "Condition": {
        "ForAllValues:StringEquals": {
          "dynamodb:LeadingKeys": ["${cognito-identity.amazonaws.com:sub}"]
        }
      }
    }
  ]
}
```

Octroi à une identité de l'autorisation d'invoquer une fonction Lambda

La stratégie d'accès suivante accorde à une identité l'autorisation d'invoquer une fonction Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": [
        "arn:aws:lambda:us-west-2:123456789012:function:MyFunction"
      ]
    }
  ]
}
```

Octroi à une identité de l'autorisation de publier des enregistrements dans Kinesis Data Streams

La stratégie d'accès suivante permet à une identité d'utiliser l'opération PutRecord avec n'importe lequel des Kinesis Data Streams. Elle peut être appliquée aux utilisateurs qui ont besoin d'ajouter des enregistrements de données à tous les flux d'un compte. Pour de plus amples informations, veuillez consulter [Contrôle de l'accès aux ressources Amazon Kinesis Data Streams à l'aide d'IAM](#) dans le Manuel du développeur Amazon Kinesis Data Streams.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": [
        "arn:aws:kinesis:us-east-1:111122223333:stream/stream1"
      ]
    }
  ]
}
```

Octroi à une identité de l'accès à ses données dans le magasin de synchronisation Amazon Cognito

La stratégie d'accès suivante accorde à une identité les autorisations d'accéder uniquement à ses propres données dans le magasin Amazon Cognito Sync.

```
{
```

```
"Version": "2012-10-17",
"Statement": [{
  "Effect": "Allow",
  "Action": "cognito-sync:*",
  "Resource": ["arn:aws:cognito-sync:us-east-1:123456789012:identitypool/${cognito-identity.amazonaws.com:aud}/identity/${cognito-identity.amazonaws.com:sub}/*"]
}]
}
```

Autorisations et approbation de rôle

Ces rôles diffèrent au niveau de leurs relations d'approbation. Voici un exemple de stratégie d'approbation pour un rôle non authentifié :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-corner-cafe-123456790ab"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "unauthenticated"
        }
      }
    }
  ]
}
```

Cette stratégie autorise les utilisateurs fédérés à partir de `cognito-identity.amazonaws.com` (émetteur du jeton OpenID Connect) à assumer ce rôle. En outre, la politique restreint le paramètre `aud` du jeton (dans ce cas, l'ID du groupe d'identités) pour qu'il corresponde au groupe d'identités. Enfin, la politique spécifie que l'un des membres du tableau de la revendication `amr` de valeur

multiple du jeton émis par l'opération API `GetOpenIdToken` Amazon Cognito possède la valeur `unauthenticated`.

Lorsque Amazon Cognito crée un jeton, le paramètre `amr` du jeton est défini comme `unauthenticated` ou `authenticated`. Si le paramètre `amr` est `authenticated`, le jeton inclut tous les fournisseurs utilisés au cours de l'authentification. Cela signifie que vous pouvez créer un rôle qui approuve uniquement les utilisateurs qui se sont connectés via Facebook en modifiant la condition `amr` comme indiqué :

```
"ForAnyValue:StringLike": {  
  "cognito-identity.amazonaws.com:amr": "graph.facebook.com"  
}
```

Soyez prudent lorsque vous modifiez les relations d'approbation de vos rôles ou lorsque vous essayez d'utiliser des rôles entre plusieurs groupes d'identités. Si vous ne configurez pas votre rôle correctement pour approuver votre groupe d'identités, il en résulte une exception STS similaire à la suivante :

```
AccessDenied -- Not authorized to perform sts:AssumeRoleWithWebIdentity
```

Si vous voyez ce message, vérifiez que votre groupe d'identités et votre type d'authentification disposent d'un rôle approprié.

Bonnes pratiques de sécurité pour les groupes d'identités Amazon Cognito

Les pools d'identités Amazon Cognito fournissent des AWS informations d'identification temporaires pour votre application. Comptes AWS contiennent souvent à la fois les ressources dont les utilisateurs de votre application ont besoin et des ressources dorsales privées. Les rôles et politiques IAM qui constituent les AWS informations d'identification peuvent accorder l'accès à n'importe laquelle de ces ressources.

La principale bonne pratique en matière de configuration du pool d'identités consiste à garantir que votre application peut effectuer le travail sans privilèges excessifs ou involontaires. Pour éviter toute mauvaise configuration de sécurité, consultez ces recommandations avant le lancement de chaque application que vous souhaitez mettre en production.

Rubriques

- [Bonnes pratiques en matière de configuration IAM](#)
- [Bonnes pratiques en matière de configuration du pool d'identités](#)

Bonnes pratiques en matière de configuration IAM

Lorsqu'un invité ou un utilisateur authentifié lance une session dans votre application qui nécessite des informations d'identification du pool d'identités, votre application récupère des informations d'AWS identification temporaires pour un rôle IAM. Les informations d'identification peuvent concerner un rôle par défaut, un rôle choisi par les règles de la configuration de votre pool d'identités ou un rôle personnalisé choisi par votre application. Les autorisations attribuées à chaque rôle permettent à votre utilisateur d'accéder à vos AWS ressources.

Pour plus d'informations sur les meilleures pratiques générales en matière d'IAM, consultez la section [Bonnes pratiques IAM](#) dans le guide de l'AWS Identity and Access Management utilisateur.

Utiliser des conditions de politique de confiance dans les rôles IAM

IAM exige que les rôles pour les pools d'identités soient soumis à au moins une condition de politique de confiance. Cette condition peut, par exemple, définir l'étendue du rôle pour les utilisateurs authentifiés uniquement. AWS STS exige également que les demandes d'authentification de base entre comptes soient soumises à deux conditions spécifiques : `cognito-identity.amazonaws.com:aud` et `cognito-identity.amazonaws.com:amr`. Il est recommandé d'appliquer ces deux conditions à tous les rôles IAM qui font confiance au principal `cognito-identity.amazonaws.com` du service des pools d'identités.

- `cognito-identity.amazonaws.com:aud`: La réclamation AUD dans le jeton du pool d'identités doit correspondre à un ID de pool d'identités fiable.
- `cognito-identity.amazonaws.com:amr`: La réclamation amr contenue dans le jeton du pool d'identités doit être authentifiée ou non authentifiée. Avec cette condition, vous pouvez réserver l'accès à un rôle uniquement à des invités non authentifiés, ou uniquement à des utilisateurs authentifiés. Vous pouvez affiner davantage la valeur de cette condition pour restreindre le rôle aux utilisateurs d'un fournisseur spécifique, par exemple `graph.facebook.com`.

L'exemple de politique de confiance des rôles suivant accorde l'accès à un rôle dans les conditions suivantes :

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
      "StringEquals": {
        "cognito-identity.amazonaws.com:aud": "us-east-1:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
      },
      "ForAnyValue:StringLike": {
        "cognito-identity.amazonaws.com:amr": "authenticated"
      }
    }
  }
]
```

Éléments relatifs aux pools d'identités

- "Federated": "cognito-identity.amazonaws.com": Les utilisateurs doivent provenir d'un pool d'identités.
- "cognito-identity.amazonaws.com:aud": "us-east-1:a1b2c3d4-5678-90ab-cdef-example11111": Les utilisateurs doivent provenir du pool d'identités spécifique us-east-1:a1b2c3d4-5678-90ab-cdef-example11111.
- "cognito-identity.amazonaws.com:amr": "authenticated": Les utilisateurs doivent être authentifiés. Les utilisateurs invités ne peuvent pas assumer ce rôle.

Appliquer les autorisations du moindre privilège

Lorsque vous définissez des autorisations à l'aide de politiques IAM pour un accès authentifié ou un accès invité, accordez uniquement les autorisations spécifiques requises pour effectuer des tâches spécifiques, ou les autorisations du moindre privilège. L'exemple de politique IAM suivant, lorsqu'il est appliqué à un rôle, accorde un accès en lecture seule à un seul fichier image dans un compartiment Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket/assets/my_picture.jpg"]
    }
  ]
}
```

Bonnes pratiques en matière de configuration du pool d'identités

Les pools d'identités proposent des options flexibles pour la génération d' AWS informations d'identification. N'utilisez pas de raccourcis de conception lorsque votre application peut fonctionner avec les méthodes les plus sécurisées.

Comprenez les effets de l'accès des invités

L'accès invité non authentifié permet aux utilisateurs de récupérer des données auprès de vous Compte AWS avant de se connecter. Toute personne connaissant l'ID de votre pool d'identités peut demander des informations d'identification non authentifiées. L'identifiant de votre pool d'identités n'est pas une information confidentielle. Lorsque vous activez l'accès invité, les AWS autorisations que vous accordez aux sessions non authentifiées sont accessibles à tous.

Il est recommandé de laisser l'accès invité désactivé et de récupérer les ressources requises uniquement après l'authentification des utilisateurs. Si votre application nécessite l'accès aux ressources avant de vous connecter, prenez les précautions suivantes.

- Familiarisez-vous avec les [limites automatiques imposées aux rôles non authentifiés](#).
- Surveillez et ajustez les autorisations de vos rôles IAM non authentifiés en fonction des besoins spécifiques de votre application.
- Accordez l'accès à des ressources spécifiques.
- Sécurisez la politique de confiance de votre rôle IAM non authentifié par défaut.
- Activez l'accès invité uniquement lorsque vous êtes sûr d'accorder les autorisations associées à votre rôle IAM à n'importe qui sur Internet.

Utiliser l'authentification améliorée par défaut

Avec l'authentification de base (classique), Amazon Cognito délègue la sélection du rôle IAM à votre application. En revanche, le flux amélioré utilise la logique centralisée de votre pool d'identités pour déterminer le rôle IAM. Il fournit également une sécurité supplémentaire pour les identités non authentifiées grâce à une [politique de délimitation](#) qui fixe une limite supérieure aux autorisations IAM. Le flux amélioré est le choix le plus sûr qui demande le moins d'efforts aux développeurs. Pour en savoir plus sur ces options, consultez [Flux d'authentification de groupes d'identités \(identités fédérées\)](#).

Le flux de base peut exposer la logique côté client qui sous-tend la sélection des rôles et l'assemblage de la demande d'informations d'identification de l'API AWS STS. Le flux amélioré masque à la fois la logique et la demande d'attribution de rôle qui sous-tendent l'automatisation du pool d'identités.

Lorsque vous configurez l'authentification de base, appliquez les [meilleures pratiques IAM](#) à vos rôles IAM et à leurs autorisations.

Utilisez les fournisseurs de développement en toute sécurité

Les identités authentifiées par les développeurs sont une fonctionnalité des pools d'identités pour les applications côté serveur. Les seules preuves d'authentification dont les pools d'identités ont besoin pour authentifier les développeurs sont les AWS informations d'identification d'un développeur de pool d'identités. Les pools d'identités n'imposent aucune restriction quant à la validité des identifiants développeur-fournisseur que vous présentez dans ce flux d'authentification.

Il est recommandé de n'implémenter des fournisseurs de développement que dans les conditions suivantes :

- Pour responsabiliser l'utilisation des informations d'identification authentifiées par le développeur, concevez le nom et les identifiants de votre fournisseur de développement de manière à indiquer la source d'authentification. Par exemple : "Logins" : {"MyCorp provider" : "[*provider application ID*]"}.
- Évitez les informations d'identification utilisateur de longue durée. [Configurez votre client côté serveur pour demander des identités avec des rôles liés à un service, tels que des profils d'instance EC2 et des rôles d'exécution Lambda.](#)
- Évitez de mélanger des sources de confiance internes et externes dans le même pool d'identités. Ajoutez votre fournisseur de développement et vos fournisseurs d'authentification unique (SSO) dans des pools d'identités distincts.

Utilisation d'attributs pour le contrôle d'accès

Les attributs de contrôle d'accès sont l'implémentation du contrôle d'accès par attributs (ABAC) dans les réserves d'identités Amazon Cognito. Vous pouvez utiliser des stratégies IAM pour contrôler l'accès aux ressources AWS via des groupes d'identités Amazon Cognito basés sur des attributs utilisateur. Ces attributs peuvent être tirés de fournisseurs d'identités sociale et d'entreprise. Vous pouvez mapper des attributs figurant dans des jetons d'accès et d'identification, ou dans des assertions SAML des fournisseurs, à des étiquettes qui peuvent être référencées dans les stratégies d'autorisations IAM.

Vous pouvez choisir des mappages par défaut ou créer vos propres mappages personnalisés dans des groupes d'identités Amazon Cognito. Les mappages par défaut vous permettent d'écrire des stratégies IAM basées sur un ensemble fixe d'attributs utilisateur. Les mappages personnalisés vous permettent de sélectionner un ensemble personnalisé d'attributs utilisateur référencés dans les stratégies d'autorisations IAM. Les noms d'attribut dans la console Amazon Cognito sont mappés à Tag key for principal (Clé d'étiquettes pour mandataire), qui sont les étiquettes référencées dans la stratégie d'autorisations IAM.

Par exemple, supposons que vous disposez d'un service de streaming multimédia avec des formules d'adhésion gratuite et payante. Vous stockez les fichiers multimédias dans Amazon S3 et les étiquetez avec des étiquettes Free ou Premium. Vous pouvez utiliser des attributs pour le contrôle d'accès afin d'autoriser l'accès au contenu gratuit et payant en fonction du niveau d'adhésion de l'utilisateur spécifié dans le profil de ce dernier. Vous pouvez mapper l'attribut d'adhésion à une clé d'étiquette pour mandataire à transmettre à la stratégie d'autorisations IAM. Vous pouvez ainsi créer une stratégie d'autorisations uniques et autoriser conditionnellement l'accès au contenu premium en fonction de la valeur du niveau d'adhésion et de l'étiquette sur les fichiers de contenu.

Rubriques

- [Utilisation d'attributs pour le contrôle d'accès avec des groupes d'identités Amazon Cognito](#)
- [Exemple d'utilisation d'attributs pour une stratégie de contrôle d'accès](#)
- [Désactiver les attributs pour le contrôle d'accès \(console\)](#)
- [Mappages de fournisseurs par défaut](#)

L'utilisation d'attributs pour contrôler l'accès présente plusieurs avantages :

- La gestion des autorisations est plus efficace lorsque vous utilisez des attributs pour le contrôle d'accès. Vous pouvez créer une stratégie d'autorisations de base qui utilise des attributs utilisateur au lieu de créer plusieurs stratégies pour différentes fonctions professionnelles.
- Vous n'avez pas besoin de mettre à jour vos stratégies chaque fois que vous ajoutez ou supprimez des ressources ou des utilisateurs pour votre application. La stratégie d'autorisations n'accorde l'accès qu'aux utilisateurs titulaires des attributs utilisateur appropriés. Par exemple, il se peut que vous deviez contrôler l'accès à certains compartiments S3 en fonction du titre professionnel des utilisateurs. Dans ce cas, vous pouvez créer une stratégie d'autorisations afin de n'autoriser l'accès à ces fichiers qu'aux utilisateurs titulaires du titre professionnel défini. Pour plus d'informations, consultez le [Didacticiel IAM : utilisation de balises de session SAML pour le contrôle ABAC](#).
- Les attributs peuvent être transmis en tant qu'étiquettes de mandataires à une stratégie qui autorise ou refuse des autorisations en fonction des valeurs de ces attributs.

Utilisation d'attributs pour le contrôle d'accès avec des groupes d'identités Amazon Cognito

Avant d'utiliser des attributs pour le contrôle d'accès, assurez-vous que les prérequis suivants sont réunis :

- [Un compte AWS](#)
- [Groupe d'utilisateurs](#)
- [Groupe d'identités](#)
- [Configuration d'un kit SDK](#)
- [Fournisseurs d'identité intégrés](#)
- [Informations d'identification](#)

Pour utiliser des attributs pour le contrôle d'accès, la demande que vous définissez comme source de données définit la valeur de la clé de balise que vous choisissez. Amazon Cognito applique la clé et la valeur de balise à la session de votre utilisateur. Vos politiques IAM peuvent évaluer l'accès de votre utilisateur à partir de la condition `{aws:PrincipalTag/tagkey}`. IAM évalue la valeur de la balise de votre utilisateur par rapport à la politique.

Vous devez préparer les rôles IAM dont vous souhaitez transmettre les informations d'identification à vos utilisateurs. La politique d'approbation de ces rôles doit autoriser Amazon Cognito à endosser le rôle pour votre utilisateur. Pour les attributs de contrôle d'accès, vous devez également autoriser

Amazon Cognito à appliquer des balises relatives au principal à la session temporaire de votre utilisateur. Accordez l'autorisation d'endosser le rôle avec l'action [AssumeRoleWithWebIdentity](#). Accordez l'autorisation afin de baliser les sessions des utilisateurs avec l'[action avec autorisation uniquement](#) `sts:TagSession`. Pour plus d'informations, consultez [Transmission des balises de session dans AWS Security Token Service](#), dans le Guide de l'utilisateur AWS Identity and Access Management. Pour obtenir un exemple de politique d'approbation qui accorde les autorisations `sts:AssumeRoleWithWebIdentity` et `sts:TagSession` au principal du service Amazon Cognito `cognito-identity.amazonaws.com`, consultez [Exemple d'utilisation d'attributs pour une stratégie de contrôle d'accès](#).

Pour configurer les attributs pour le contrôle d'accès dans la console

1. Connectez-vous à la [console Amazon Cognito](#) et sélectionnez Groupes d'identités. Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Localisez Fournisseurs d'identité. Choisissez le fournisseur d'identité que vous souhaitez modifier. Si vous souhaitez ajouter un nouveau fournisseur d'identité, sélectionnez Ajouter un fournisseur d'identité.
4. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, choisissez Modifier dans Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard `sub` et `aud`, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
5. Sélectionnez Enregistrer les modifications.

Exemple d'utilisation d'attributs pour une stratégie de contrôle d'accès

Découvrez un scénario dans lequel un employé du service juridique d'une société doit répertorier tous les fichiers figurant dans des compartiments appartenant à son département, qui sont classés

en fonction de leur niveau de sécurité. Supposons que le jeton que cet employé reçoit du fournisseur d'identité contient les revendications suivantes.

Revendications

```
{ .
  .
  "sub" : "57e7b692-4f66-480d-98b8-45a6729b4c88",
  "department" : "legal",
  "clearance" : "confidential",
  .
  .
}
```

Ces attributs peuvent être mappés à des étiquettes et référencés dans des stratégies d'autorisations IAM en tant qu'étiquettes de mandataires. Vous pouvez désormais gérer l'accès en modifiant le profil utilisateur du côté fournisseur d'identité. Vous pouvez également modifier les attributs du côté ressource en utilisant des noms ou des étiquettes sans modifier la stratégie elle-même.

La stratégie d'autorisations suivante fait deux choses :

- Elle permet de répertorier l'accès à tous les compartiments S3 qui se terminent par un préfixe correspondant au nom de service de l'utilisateur.
- Elle permet d'accéder en lecture aux fichiers figurant dans ces compartiments pour autant que l'étiquette d'autorisation sur le fichier corresponde à l'attribut d'autorisation de l'utilisateur.

Politique d'autorisations

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:List*",
      "Resource": "arn:aws:s3:::*-${aws:PrincipalTag/department}"
    },
    {
      "Effect": "Allow",
```



```

    "Action": "s3:GetObject*",
    "Resource": "arn:aws:s3:::*-${aws:PrincipalTag/department}/*",
    "Condition": {
      "StringEquals": {
        "s3:ExistingObjectTag/clearance": "${aws:PrincipalTag/clearance}"
      }
    }
  }
]
}

```

La stratégie d'approbation détermine qui peut assumer ce rôle. La stratégie de relation d'approbation permet d'utiliser `sts:AssumeRoleWithWebIdentity` et `sts:TagSession` pour autoriser l'accès. Elle ajoute des conditions pour limiter la politique à la réserve d'identités que vous avez créée, et elle garantit que la politique est destinée à un rôle authentifié.

Politique d'approbation

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRoleWithWebIdentity",
        "sts:TagSession"
      ],
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "IDENTITY-POOL-ID"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}

```

}

Désactiver les attributs pour le contrôle d'accès (console)

Suivez cette procédure pour désactiver les attributs pour le contrôle d'accès.

Pour désactiver les attributs pour le contrôle d'accès dans la console

1. Connectez-vous à la [console Amazon Cognito](#) et sélectionnez Groupes d'identités. Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Localisez Fournisseurs d'identité. Choisissez le fournisseur d'identité que vous souhaitez modifier.
4. Choisissez Modifier dans Attributs pour le contrôle d'accès.
5. Pour n'appliquer aucune balise de principal, choisissez Inactif.
6. Sélectionnez Enregistrer les modifications.

Mappages de fournisseurs par défaut

Le tableau suivant contient les informations de mappage par défaut pour les fournisseurs d'authentification qu'Amazon Cognito prend en charge.

Fournisseur	Type de jeton	Valeurs d'étiquette de mandataire	Exemple
Groupe d'utilisateurs Amazon Cognito	Jeton d'identification	aud(client ID) et sub(user ID)	"6jk8ltokc7ac9es6jrtg9q572f", "57e7b692-4f66-480d-98b8-45a6729b4c88"
Facebook	Jeton d'accès	aud(app_id), sub(user_id)	"492844718097981", "112177216992379"
Google	Jeton d'identification	aud(client ID) et sub(user ID)	"620493171733-eebk7c0hcp5lj3e1tlqp1gntt3k0rncv.apps.go

Fournisseur	Type de jeton	Valeurs d'étiquette de mandataire	Exemple
			ogleusercontent.com", "10922006345240474 6097"
SAML	Assertions	"http://schemas.xml Isoap.org/ws/2005/ 05/identity/claims /nameidentifier" , "http://schemas.xml Isoap.org/ws/2005/05/ identity/claims/name"	"auth0 5e28d196f8f 55a0eaaa95de3", "user123@gmail.com"
Apple	Jeton d'identification	aud(client ID) et sub (user ID)	"com.amazonaws.ec2 -54-80-172-243.com pute-1.client", "001968.a6ca34e9c1 e742458a26cf800585 4be9.0733"
Amazon	Jeton d'accès	aud (Client ID on Amzn Dev Ac), user_id(user ID)	"amzn1.application -oa2-client.9d70d9 382d34461 08aaee3dd763a0fa6", "amzn1.account.AGH NIFJQMFSB G3G6XCPVB 35ORQAA"
Fournisseurs d'OIDC standard	Jetons d'identification et d'accès	aud (as client_id), sub (as user ID)	"620493171733-eebk 7c0hcp5lj3e1tlqp1g ntt3k0rncv.apps.go ogleusercontent.com", "10922006345240474 6097"

Fournisseur	Type de jeton	Valeurs d'étiquette de mandataire	Exemple
Twitter	Jeton d'accès	aud (app ID; app Secret), sub (user ID)	"DfwifTtKEX1FiIBRn OTIR0CFK; Xgj5xb8xlrIVCPjXgL ldkW7fXmw cJJrFvnoK9gwZkLexo 1y5z1", "12690038 84292222976"
DevAuth	Map	Ne s'applique pas	"tag1", "tag2"

Note

L'option de mappages d'attributs par défaut est automatiquement renseignée pour les champs Tag Key for Principal (Clé d'étiquette pour mandataire) et Attribute Names (Noms d'attribut). Vous ne pouvez pas modifier les mappages par défaut.

Utilisation du contrôle d'accès basé sur les rôles

Les groupes d'identités Amazon Cognito attribuent à vos utilisateurs authentifiés un ensemble d'informations d'identification temporaires à privilèges limités pour accéder à vos ressources. AWS Les autorisations pour chaque utilisateur sont contrôlées via des [rôles IAM](#) que vous créez. Vous pouvez définir des règles pour choisir le rôle pour chaque utilisateur en fonction de demandes dans le jeton d'ID de l'utilisateur. Vous pouvez définir un rôle par défaut pour les utilisateurs authentifiés. Vous pouvez également définir un rôle IAM distinct avec des autorisations limitées pour les utilisateurs invités qui ne sont pas authentifiés.

Création de rôles pour le mappage de rôles

Il est important d'ajouter la politique d'approbation appropriée pour chaque rôle afin que celui-ci puisse uniquement être assumé par Amazon Cognito pour les utilisateurs authentifiés dans votre groupe d'identités. Voici un exemple d'une telle politique d'approbation :

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
      "StringEquals": {
        "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-corner-
cafe-123456790ab"
      },
      "ForAnyValue:StringLike": {
        "cognito-identity.amazonaws.com:amr": "authenticated"
      }
    }
  }
]
```

Cette politique autorise les utilisateurs fédérés à partir de `cognito-identity.amazonaws.com` (diffuseur du jeton OpenID Connect) à assumer ce rôle. En outre, la politique restreint le paramètre `aud` du jeton (dans ce cas, l'ID du groupe d'identités) pour qu'il corresponde au groupe d'identités. Enfin, la politique spécifie que l'un des membres du tableau de la revendication `amr` de valeur multiple du jeton émis par l'action API `GetOpenIdToken` Amazon Cognito possède la valeur `authenticated`.

Octroi d'une autorisation de transmission de rôle

Pour permettre à un utilisateur de définir des rôles avec des autorisations dépassant les autorisations existantes de l'utilisateur sur un groupe d'identités, accordez-leur l'autorisation `iam:PassRole` pour transmettre le rôle à l'API `set-identity-pool-roles`. Par exemple, si l'utilisateur ne peut pas écrire dans Amazon S3 mais que le rôle IAM que l'utilisateur définit sur le groupe d'identités accorde une autorisation en écriture dans Amazon S3, l'utilisateur peut définir ce rôle uniquement si une autorisation `iam:PassRole` est octroyée pour le rôle. L'exemple de politique suivant montre comment accorder l'autorisation `iam:PassRole`.

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

```
"Statement": [
  {
    "Sid": "Stmt1",
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::123456789012:role/myS3WriteAccessRole"
    ]
  }
]
```

Dans cet exemple de politique, l'autorisation `iam:PassRole` est accordée pour le rôle `myS3WriteAccessRole`. Le rôle a été spécifié à l'aide de l'Amazon Resource Name (ARN) du rôle. Vous devez également attacher cette politique à votre utilisateur. Pour plus d'informations, consultez [Utilisation des politiques gérées](#).

Note

Les fonctions Lambda utilisent des politiques basées sur des ressources, où la politique est attachée directement à la fonction Lambda. Lorsque vous créez une règle qui appelle une fonction Lambda, vous ne transmettez pas de rôle, et l'utilisateur qui crée la règle n'a donc pas besoin de l'autorisation `iam:PassRole`. Pour plus d'informations sur l'autorisation de fonctions Lambda, consultez [Gestion des autorisations : utilisation d'une politique de fonction Lambda](#).

Utilisation de jetons pour attribuer des rôles aux utilisateurs

Pour les utilisateurs qui se connectent via les groupes d'utilisateurs Amazon Cognito, des rôles peuvent être transmis dans le jeton d'identification qui a été affecté par le groupe d'utilisateurs. Les rôles apparaissent dans les demandes suivantes dans le jeton d'ID :

- La demande `cognito:preferred_role` est l'ARN du rôle.
- La demande `cognito:roles` est une chaîne séparée par des virgules contenant un ensemble d'ARN de rôles autorisés.

Les demandes sont définies comme suit :

- La demande `cognito:preferred_role` est définie sur le rôle du groupe avec la meilleure valeur pour `Precedence` (la valeur la plus faible). S'il n'existe qu'un seul rôle autorisé, `cognito:preferred_role` est défini sur ce rôle. S'il existe plusieurs rôles et qu'aucun rôle ne dispose de la meilleure priorité, cette demande n'est pas définie.
- La demande `cognito:roles` est définie, s'il existe au moins un rôle.

Lorsque des jetons sont utilisés pour attribuer des rôles, s'il existe plusieurs rôles pouvant être affectés à l'utilisateur, la fonctionnalité de Groupes d'identités Amazon Cognito (identités fédérées) choisit le rôle comme suit :

- Utilisez le [GetCredentialsForIdentityCustomRoleArn](#) paramètre s'il est défini et qu'il correspond à un rôle dans la `cognito:roles` réclamation. Si ce paramètre ne correspond pas à un rôle dans `cognito:roles`, l'accès est refusé.
- Si la demande `cognito:preferred_role` est définie, elle est utilisée.
- Si la `cognito:preferred_role` réclamation n'est pas définie, la `cognito:roles` réclamation est définie et `CustomRoleArn` n'est pas spécifiée dans l'appel à `GetCredentialsForIdentity`, puis le paramètre de résolution des rôles dans la console ou le `AmbiguousRoleResolution` champ (dans le `RoleMappings` paramètre de l'[SetIdentityPoolRoles](#) API) est utilisé pour déterminer le rôle à attribuer.

Utilisation du mappage basé sur des règles pour attribuer des rôles aux utilisateurs

Les règles vous permettent de mapper des revendications d'un jeton de fournisseur d'identité à des rôles IAM.

Chaque règle spécifie une demande de jeton (par exemple, un attribut utilisateur dans le jeton d'ID d'un groupe d'utilisateurs Amazon Cognito), un type de correspondance, une valeur et un rôle IAM. Le type de correspondance peut être `Equals`, `NotEqual`, `StartsWith` ou `Contains`. Si un utilisateur a une valeur correspondante pour la demande, il peut assumer ce rôle lorsqu'il obtient les informations d'identification. Par exemple, vous pouvez créer une règle qui accorde un rôle IAM spécifique pour les utilisateurs avec une valeur d'attribut personnalisé `custom:dept Sales`.

Note

Dans les paramètres de la règle, les attributs personnalisés ont besoin du préfixe `custom:` pour se distinguer des attributs standard.

Les règles sont évaluées dans l'ordre, et le rôle IAM pour la première règle correspondante est utilisé, sauf si `CustomRoleArn` a été spécifié pour modifier l'ordre. Pour plus d'informations sur les attributs utilisateur dans les groupes d'utilisateurs Amazon Cognito, consultez [Attributs de groupe d'utilisateurs](#).

Vous pouvez définir plusieurs règles pour un fournisseur d'authentification dans la console de groupe d'identités (Identités fédérées). Les règles sont appliquées dans l'ordre. Vous pouvez faire glisser les règles pour changer leur ordre. La première règle correspondante a la priorité. Si le type de correspondance est `NotEqual` et que la demande n'existe pas, la règle n'est pas évaluée. Si aucune règle ne correspond, le paramètre Résolution de rôle est appliqué à Utiliser le rôle authentifié par défaut ou Refuser la demande.

Dans l'API et la CLI, vous pouvez spécifier le rôle à attribuer lorsqu'aucune règle ne correspond dans le `AmbiguousRoleResolution` champ du [RoleMapping](#) type spécifié dans le `RoleMappings` paramètre de l'[SetIdentityPoolRolesAPI](#).

Vous pouvez configurer un mappage basé sur des règles pour les fournisseurs d'identité OpenID Connect (OIDC) et SAML dans l'API AWS CLI or avec le champ du type. `RulesConfiguration RoleMapping` Vous pouvez spécifier ce champ dans le `RoleMappings` paramètre de l'[SetIdentityPoolRolesAPI](#). AWS Management Console Actuellement, vous ne pouvez pas ajouter de règles pour les fournisseurs OIDC ou SAML.

Par exemple, la AWS CLI commande suivante ajoute une règle qui attribue le rôle `arn:aws:iam::123456789012:role/Sacramento_team_S3_admin` aux utilisateurs de votre site de Sacramento qui ont été authentifiés par OIDC IdP : `arn:aws:iam::123456789012:oidc-provider/myOIDCIdP`

```
aws cognito-identity set-identity-pool-roles --region us-east-1 --cli-input-json
file://role-mapping.json
```

Contenu de **role-mapping.json** :

```
{
```



```
"IdentityPoolId": "us-east-1:12345678-corner-cafe-123456790ab",
"Roles": {
  "authenticated": "arn:aws:iam::123456789012:role/myS3WriteAccessRole",
  "unauthenticated": "arn:aws:iam::123456789012:role/myS3ReadAccessRole"
},
"RoleMappings": {
  "arn:aws:iam::123456789012:oidc-provider/myOIDCIdP": {
    "Type": "Rules",
    "AmbiguousRoleResolution": "AuthenticatedRole",
    "RulesConfiguration": {
      "Rules": [
        {
          "Claim": "locale",
          "MatchType": "Equals",
          "Value": "Sacramento",
          "RoleARN": "arn:aws:iam::123456789012:role/
Sacramento_team_S3_admin"
        }
      ]
    }
  }
}
```

Pour chaque groupe d'utilisateurs ou autre fournisseur d'authentification que vous configurez pour un groupe d'identités, vous pouvez créer jusqu'à 25 règles. Cette limite n'est pas réglable. Pour plus d'informations, consultez [Quotas dans Amazon Cognito](#).

Demandes de jetons à utiliser dans le mappage basé sur des règles

Amazon Cognito

Un jeton d'identification Amazon Cognito est représenté comme un jeton web JSON (JSON Web Token, JWT). Il contient les demandes sur l'identité de l'utilisateur authentifié, par exemple `name`, `family_name` et `phone_number`. Pour plus d'informations sur les demandes standard, consultez la [spécification OpenID Connect](#). Hormis les revendications standard, les revendications suivantes sont spécifiques d'Amazon Cognito :

- `cognito:groups`
- `cognito:roles`
- `cognito:preferred_role`

Amazon

Les demandes suivantes, ainsi que les valeurs possibles pour ces demandes, peuvent être utilisées avec Login with Amazon :

- `iss` : `www.amazon.com`
- `aud` : ID d'application
- `sub` : sub du jeton Login with Amazon

Facebook

Les demandes suivantes, ainsi que les valeurs possibles pour ces demandes, peuvent être utilisées avec Facebook :

- `iss` : `graph.facebook.com`
- `aud` : ID d'application
- `sub` : sub du jeton Facebook

Google

Jeton Google contenant les demandes standard émanant de la [spécification OpenID Connect](#). Toutes les demandes contenues dans le jeton OpenID sont disponibles pour le mappage basé sur des règles. Consultez le site [OpenID Connect](#) de Google pour connaître les demandes disponibles grâce au jeton Google.

Apple

Jeton Apple contenant les demandes standards émanant de la [spécification OpenID Connect](#). Veuillez consulter [Authenticating Users with Sign in with Apple](#) dans la documentation Apple pour en savoir plus sur la demande disponible avec le jeton Apple. Le jeton Apple ne contient pas toujours `email`.

OpenID

Toutes les demandes contenues dans le jeton Open Id sont disponibles pour le mappage basé sur des règles. Pour plus d'informations sur les demandes standard, consultez la [spécification OpenID Connect](#). Consultez la documentation de votre fournisseur OpenID pour connaître les demandes supplémentaires disponibles.

SAML

Les demandes sont analysées à partir de l'assertion SAML. Toutes les demandes disponibles dans l'assertion SAML peuvent être utilisées pour le mappage basé sur des règles.

Bonnes pratiques pour le contrôle d'accès basé sur les rôles

Important

Si la demande qui vous mappez à un rôle peut être modifiée par l'utilisateur final, tout utilisateur final peut assumer votre rôle et définir la politique en conséquence. Mappez uniquement les demandes qui ne peuvent pas être définies directement par l'utilisateur final à des rôles avec des autorisations de niveau élevé. Dans un groupe d'utilisateurs Amazon Cognito, vous pouvez définir des autorisations de lecture et d'écriture par application pour chaque attribut utilisateur.

Important

Si vous définissez des rôles pour des groupes dans un groupe d'utilisateurs Amazon Cognito, ces rôles sont transmis via le jeton d'identification de l'utilisateur. Pour utiliser ces rôles, vous devez également définir Choose role from token (Choisir un rôle à partir d'un jeton) pour la sélection de rôle authentifié pour le groupe d'identités.

Vous pouvez utiliser le paramètre de résolution des rôles de la console et le `RoleMappings` paramètre de l'[SetIdentityPoolRoles](#) API pour spécifier le comportement par défaut lorsque le rôle correct ne peut pas être déterminé à partir du jeton.

Obtention des informations d'identification

Vous pouvez utiliser Amazon Cognito pour fournir des informations d'identification temporaires à privilèges limités à votre application, afin que vos utilisateurs puissent accéder aux ressources. AWS Cette section décrit comment obtenir des informations d'identification et comment récupérer une identité Amazon Cognito à partir d'un groupe d'identités.

Amazon Cognito prend en charge les identités authentifiées et non authentifiées. L'identité des utilisateurs non authentifiés n'est pas vérifiée. Ce rôle convient donc pour les utilisateurs invités de votre application ou dans les cas où il n'est pas important que les identités des utilisateurs soient

vérifiées. Les utilisateurs authentifiés se connectent à votre application via un fournisseur d'identité tiers, ou un groupe d'utilisateurs, qui vérifie leur identité. Assurez-vous de définir de façon appropriée les autorisations des ressources afin de ne pas y accorder l'accès aux utilisateurs non authentifiés.

Les identités Amazon Cognito ne sont pas des informations d'identification. Ils sont échangés contre des informations d'identification en utilisant le support de fédération d'identité Web dans le AWS Security Token Service (AWS STS). Afin d'obtenir des informations d'identification AWS pour les utilisateurs de votre application, nous vous recommandons d'utiliser `AWS.CognitoIdentityCredentials`. L'identité contenue dans l'objet d'informations d'identification est ensuite échangée contre des informations d'identification à l'aide de AWS STS.

Note

Si vous avez créé votre réserve d'identités avant février 2015, vous devez y réassocier vos rôles afin de pouvoir utiliser le constructeur `AWS.CognitoIdentityCredentials` sans les rôles en tant que paramètres. Pour ce faire, ouvrez la [console Amazon Cognito](#), choisissez Manage Identity Pools (Gérer les groupes d'identités), choisissez le groupe d'identités puis Edit Identity Pool (Modifier le groupe d'identités), spécifiez les rôles authentifiés et non authentifiés, puis enregistrez les modifications.

Les fournisseurs d'informations d'identification d'identité Web font partie de la chaîne de fournisseurs d'informations d'identification par défaut des kits AWS SDK. Pour définir votre jeton de pool d'identités dans un config fichier local pour un AWS SDK ou le AWS CLI, ajoutez une entrée `web_identity_token_file` de profil. Voir [Assumer le rôle de fournisseur d'informations d'identification](#) dans le guide de référence AWS des SDK et des outils.

Pour en savoir plus sur la façon de remplir les informations d'identification d'identité Web dans votre kit SDK, consultez le Guide du développeur du kit SDK. Pour de meilleurs résultats, démarrez votre projet avec l'intégration du pool d'identités intégrée à AWS Amplify.

AWS Ressources du SDK pour obtenir et définir des informations d'identification avec des pools d'identités

- [Fédération de réserve d'identités](#) (Android) (langue française non garantie) sur le site Amplify Dev Center
- [Fédération de réserve d'identités](#) (iOS) (langue française non garantie) sur le site Amplify Dev Center

- [Utilisation d'Amazon Cognito Identity pour authentifier les utilisateurs](#) dans le manuel du développeur AWS SDK for JavaScript
- [Fournisseur d'informations d'identification Amazon Cognito](#) dans le guide du développeur AWS SDK for .NET
- [Spécifiez les informations d'identification par programmation](#) dans le guide du développeur AWS SDK for Go
- [Fournissez des informations d'identification temporaires sous forme de code](#) dans le guide du AWS SDK for Java 2.x développeur
- [assumeRoleWithWebIdentityCredentialProvider](#) fournisseur dans le guide AWS SDK for PHP du développeur
- [Endosser un rôle avec le fournisseur d'identité Web](#) dans la documentation sur AWS SDK for Python (Boto3)
- [Spécification de vos informations d'identification et de votre région par défaut](#) dans le guide du Kit AWS SDK pour Rust développeur

Les sections suivantes fournissent des exemples de code dans certains anciens AWS SDK.

Android

Vous pouvez utiliser Amazon Cognito pour fournir des informations d'identification temporaires à privilèges limités à votre application, afin que vos utilisateurs puissent accéder aux ressources. AWS Amazon Cognito prend en charge les identités authentifiées et non authentifiées. Pour fournir des AWS informations d'identification à votre application, suivez les étapes ci-dessous.

Pour utiliser un pool d'identités Amazon Cognito dans une application Android, configurez-le. AWS Amplify Pour plus d'informations, consultez [Authentification](#) (langue française non garantie) sur le site Amplify Dev Center.

Récupération d'une identité Amazon Cognito

Si vous autorisez les utilisateurs non authentifiés, vous pouvez récupérer immédiatement un identifiant unique Amazon Cognito (ID d'identité) pour vos utilisateurs finaux. Si vous authentifiez des utilisateurs, vous pouvez extraire l'ID d'identité après avoir défini les jetons de connexion dans le fournisseur d'informations d'identification :

```
String identityId = credentialsProvider.getIdentityId();
Log.d("LogTag", "my ID is " + identityId);
```

Note

N'appellez pas `getIdentityId()`, `refresh()` ni `getCredentials()` dans le thread principal de votre application. À partir d'Android 3.0 (niveau d'API 11), votre application échouera automatiquement et lancera [NetworkOnMainThreadException](#) une E/S réseau sur le thread principal de l'application. Dans ce cas, vous devez transférer le code à un thread en arrière-plan avec `AsyncTask`. Pour plus d'informations, consultez la [documentation Android](#). Vous pouvez également appeler `getCachedIdentityId()` pour récupérer un ID, mais uniquement si un ID est déjà mis en cache localement. Sinon, la méthode renvoie une valeur nulle.

iOS : Objective-C

Vous pouvez utiliser Amazon Cognito pour fournir des informations d'identification temporaires à privilèges limités à votre application, afin que vos utilisateurs puissent accéder aux ressources. AWS Les groupes d'identités Amazon Cognito prennent en charge les identités authentifiées et non authentifiées. Pour fournir des AWS informations d'identification à votre application, procédez comme suit.

Pour utiliser un pool d'identités Amazon Cognito dans une application iOS, configurez-le. AWS Amplify Pour plus d'informations, consultez [Authentication Swift](#) et [Authentication Flutter](#) (langue française non garantie) sur le site Amplify Dev Center.

Récupération d'une identité Amazon Cognito

Vous pouvez récupérer immédiatement un identifiant Amazon Cognito unique (ID d'identité) pour l'utilisateur si vous autorisez les utilisateurs non authentifiés ou une fois que vous avez défini les jetons de connexion dans le fournisseur d'informations d'identification si vous authentifiez les utilisateurs :

```
// Retrieve your Amazon Cognito ID
[[credentialsProvider getIdentityId] continueWithBlock:^id(AWSTask *task) {
    if (task.error) {
        NSLog(@"Error: %@", task.error);
    }
    else {
        // the task result will contain the identity id
        NSString *cognitoId = task.result;
    }
}
```

```
    return nil;
  }];
```

Note

`getIdentityId` est un appel asynchrone. Si un ID d'identité est déjà configuré sur votre fournisseur, vous pouvez appeler `credentialsProvider.identityId` pour récupérer cette identité, qui est mise en cache localement. Toutefois, si un ID d'identité n'est pas défini sur votre fournisseur, l'appel `credentialsProvider.identityId` renvoie la valeur `nil`. Pour plus d'informations, consultez la [Référence du kit SDK Amplify iOS](#).

iOS : Swift

Vous pouvez utiliser Amazon Cognito pour fournir des informations d'identification temporaires à privilèges limités à votre application afin que vos utilisateurs puissent accéder aux ressources. AWS Amazon Cognito prend en charge les identités authentifiées et non authentifiées. Pour fournir des AWS informations d'identification à votre application, suivez les étapes ci-dessous.

Pour utiliser un pool d'identités Amazon Cognito dans une application iOS, configurez-le. AWS Amplify Pour plus d'informations, consultez [Authentification Swift](#) (langue française non garantie) sur le site Amplify Dev Center.

Récupération d'une identité Amazon Cognito

Vous pouvez récupérer immédiatement un identifiant Amazon Cognito unique (ID d'identité) pour l'utilisateur si vous autorisez les utilisateurs non authentifiés ou une fois que vous avez défini les jetons de connexion dans le fournisseur d'informations d'identification si vous authentifiez les utilisateurs :

```
// Retrieve your Amazon Cognito ID
credentialsProvider.getIdentityId().continewith(block: { (task) -> AnyObject? in
    if (task.error != nil) {
        print("Error: " + task.error!.localizedDescription)
    }
    else {
        // the task result will contain the identity id
        let cognitoId = task.result!
        print("Cognito id: \(cognitoId)")
    }
    return task;
```

```
});
```

Note

`getIdentityId` est un appel asynchrone. Si un ID d'identité est déjà configuré sur votre fournisseur, vous pouvez appeler `credentialsProvider.identityId` pour récupérer cette identité, qui est mise en cache localement. Toutefois, si un ID d'identité n'est pas défini sur votre fournisseur, l'appel `credentialsProvider.identityId` renvoie la valeur `nil`. Pour plus d'informations, consultez la [Référence du kit SDK Amplify iOS](#).

JavaScript

Si ce n'est pas déjà fait, créez un groupe d'identités dans la [console Amazon Cognito](#) avant d'utiliser `AWS.CognitoIdentityCredentials`.

Une fois que vous avez configuré un groupe d'identités avec vos fournisseurs d'identité, vous pouvez utiliser `AWS.CognitoIdentityCredentials` pour authentifier les utilisateurs. Pour configurer les informations d'identification de votre application afin d'utiliser `AWS.CognitoIdentityCredentials`, définissez la propriété `credentials` d'une configuration `AWS.Config` ou d'une configuration par service. L'exemple suivant utilise `AWS.Config` :

```
// Set the region where your identity pool exists (us-east-1, eu-west-1)
AWS.config.region = 'us-east-1';

// Configure the credentials provider to use your identity pool
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: 'IDENTITY_POOL_ID',
  Logins: { // optional tokens, used for authenticated login
    'graph.facebook.com': 'FBTOKEN',
    'www.amazon.com': 'AMAZONTOKEN',
    'accounts.google.com': 'GOOGLETOKEN',
    'appleid.apple.com': 'APPLETOKEN'
  }
});

// Make the call to obtain credentials
AWS.config.credentials.get(function(){

  // Credentials will be available when this function is called.
  var accessKeyId = AWS.config.credentials.accessKeyId;
```



```
var secretAccessKey = AWS.config.credentials.secretAccessKey;
var sessionToken = AWS.config.credentials.sessionToken;

});
```

La propriété facultative `Logins` est un mappage de noms de fournisseur d'identité avec les jetons d'identité de ces fournisseurs. La façon dont vous obtenez le jeton de la part de votre fournisseur d'identité dépend du fournisseur que vous utilisez. Par exemple, si Facebook est l'un de vos fournisseurs d'identité, vous pouvez utiliser la fonction `FB.login` du [kit SDK Facebook](#) pour obtenir un jeton de fournisseur d'identité :

```
FB.login(function (response) {
  if (response.authResponse) { // logged in
    AWS.config.credentials = new AWS.CognitoIdentityCredentials({
      IdentityPoolId: 'us-east-1:1699ebc0-7900-4099-b910-2df94f52a030',
      Logins: {
        'graph.facebook.com': response.authResponse.accessToken
      }
    });

    console.log('You are now logged in.');
```

```
  } else {
    console.log('There was a problem logging you in.');
```

```
  }
});
```

Récupération d'une identité Amazon Cognito

Vous pouvez récupérer immédiatement un identifiant Amazon Cognito unique (ID d'identité) pour l'utilisateur si vous autorisez les utilisateurs non authentifiés ou une fois que vous avez défini les jetons de connexion dans le fournisseur d'informations d'identification si vous authentifiez les utilisateurs :

```
var identityId = AWS.config.credentials.identityId;
```

Unity

Vous pouvez utiliser Amazon Cognito pour fournir des informations d'identification temporaires à privilèges limités à votre application, afin que vos utilisateurs puissent accéder aux ressources. AWS Amazon Cognito prend en charge les identités authentifiées et non authentifiées. Pour fournir des AWS informations d'identification à votre application, suivez les étapes ci-dessous.

Le kit [AWS SDK for Unity](#) fait désormais partie du kit [AWS SDK for .NET](#). Pour commencer à utiliser Amazon Cognito dans le AWS SDK for .NET, consultez la section relative au fournisseur d'informations d'[identification Amazon Cognito](#) dans AWS SDK for .NET le guide du développeur. Ou consultez le [centre de développement d'Amplify](#) pour découvrir les options permettant de créer une application avec. AWS Amplify

Récupération d'une identité Amazon Cognito

Vous pouvez récupérer immédiatement un identifiant Amazon Cognito unique (ID d'identité) pour l'utilisateur si vous autorisez les utilisateurs non authentifiés ou une fois que vous avez défini les jetons de connexion dans le fournisseur d'informations d'identification si vous authentifiez les utilisateurs :

```
credentials.GetIdentityIdAsync(delegate(AmazonCognitoIdentityResult<string> result) {
    if (result.Exception != null) {
        //Exception!
    }
    string identityId = result.Response;
});
```

Xamarin

Vous pouvez utiliser Amazon Cognito pour fournir des informations d'identification temporaires à privilèges limités à votre application afin que vos utilisateurs puissent accéder aux ressources. AWS Amazon Cognito prend en charge les identités authentifiées et non authentifiées. Pour fournir des AWS informations d'identification à votre application, suivez les étapes ci-dessous.

Le kit [AWS SDK for Xamarin](#) fait désormais partie du kit [AWS SDK for .NET](#). Pour commencer à utiliser Amazon Cognito dans le AWS SDK for .NET, consultez la section relative au fournisseur d'informations d'[identification Amazon Cognito](#) dans AWS SDK for .NET le guide du développeur. Ou consultez le [centre de développement d'Amplify](#) pour découvrir les options permettant de créer une application avec. AWS Amplify

Note

Remarque : si vous avez créé le groupe d'identités avant février 2015, vous devez y réassocier les rôles afin de pouvoir utiliser ce constructeur sans les rôles en tant que paramètres. Pour ce faire, ouvrez la [console Amazon Cognito](#), choisissez Manage Identity Pools (Gérer les groupes d'identités), choisissez le groupe d'identités puis Edit Identity

Pool (Modifier le groupe d'identités), spécifiez les rôles authentifiés et non authentifiés, puis enregistrez les modifications.

Récupération d'une identité Amazon Cognito

Vous pouvez récupérer immédiatement un identifiant Amazon Cognito unique (ID d'identité) pour l'utilisateur si vous autorisez les utilisateurs non authentifiés ou une fois que vous avez défini les jetons de connexion dans le fournisseur d'informations d'identification si vous authentifiez les utilisateurs :

```
var identityId = await credentials.GetIdentityIdAsync();
```

Accès aux AWS services

Après avoir configuré votre fournisseur d'informations d'identification Amazon Cognito et récupéré les AWS informations d'identification, vous pouvez créer un Service AWS client.

AWS Ressources du SDK pour créer un client

- [AWS Configuration du client](#) dans le guide du AWS SDK for C++ développeur
- [Utilisation de la AWS SDK for Go V2 Services AWS](#) dans le guide du AWS SDK for Go développeur
- [Configuration des clients HTTP](#) dans le guide du AWS SDK for Java 2.x développeur
- [Création et appel d'objets de service](#) dans le guide du AWS SDK for JavaScript développeur
- [Création de clients](#) dans la AWS SDK for Python (Boto3) documentation
- [Création d'un client de service](#) dans le guide du Kit AWS SDK pour Rust développeur
- [Utilisation des clients](#) dans le guide du Kit AWS SDK pour Swift développeur

L'extrait suivant initialise un client Amazon DynamoDB :

Android

Pour utiliser un pool d'identités Amazon Cognito dans une application Android, configurez-le. AWS Amplify Pour plus d'informations, consultez [Authentification](#) (langue française non garantie) sur le site Amplify Dev Center.

```
// Create a service client with the provider
AmazonDynamoDB client = new AmazonDynamoDBClient(credentialsProvider);
```

Le fournisseur d'informations d'identification communique avec Amazon Cognito, récupérant à la fois l'identifiant unique des utilisateurs authentifiés et non authentifiés, ainsi que des informations d'identification temporaires à privilèges AWS limités pour le SDK mobile. AWS Les informations d'identification récupérées sont valides pendant une heure. Le fournisseur les actualise lorsqu'elles expirent.

iOS : Objective-C

Pour utiliser un pool d'identités Amazon Cognito dans une application iOS, configurez-le. AWS Amplify Pour plus d'informations, consultez [Authentication Swift](#) et [Authentication Flutter](#) (langue française non garantie) sur le site Amplify Dev Center.

```
// create a configuration that uses the provider
AWSServiceConfiguration *configuration = [AWSServiceConfiguration
    configurationWithRegion:AWSRegionUSEast1 provider:credentialsProvider];
// get a client with the default service configuration
AWSDynamoDB *dynamoDB = [AWSDynamoDB defaultDynamoDB];
```

Le fournisseur d'informations d'identification communique avec Amazon Cognito, récupérant à la fois l'identifiant unique des utilisateurs authentifiés et non authentifiés, ainsi que des informations d'identification temporaires à privilèges AWS limités pour le SDK mobile. AWS Les informations d'identification récupérées sont valides pendant une heure. Le fournisseur les actualise lorsqu'elles expirent.

iOS : Swift

Pour utiliser un pool d'identités Amazon Cognito dans une application iOS, configurez-le. AWS Amplify Pour plus d'informations, consultez [Authentication Swift](#) (langue française non garantie) sur le site Amplify Dev Center.

```
// get a client with the default service configuration
let dynamoDB = AWSDynamoDB.default()

// get a client with a custom configuration
AWSDynamoDB.register(with: configuration!, forKey: "USWest2DynamoDB");
let dynamoDBCustom = AWSDynamoDB(forKey: "USWest2DynamoDB")
```

Le fournisseur d'informations d'identification communique avec Amazon Cognito, récupérant à la fois l'identifiant unique des utilisateurs authentifiés et non authentifiés, ainsi que des informations d'identification temporaires à privilèges AWS limités pour le SDK mobile. AWS Les informations d'identification récupérées sont valides pendant une heure. Le fournisseur les actualise lorsqu'elles expirent.

JavaScript

```
// Create a service client with the provider
var dynamodb = new AWS.DynamoDB({region: 'us-west-2'});
```

Le fournisseur d'informations d'identification communique avec Amazon Cognito, récupérant à la fois l'identifiant unique pour les utilisateurs authentifiés et non authentifiés ainsi que les informations d'identification temporaires à privilèges limités AWS pour le SDK mobile. AWS Les informations d'identification récupérées sont valides pendant une heure. Le fournisseur les actualise lorsqu'elles expirent.

Unity

Le kit [AWS SDK for Unity](#) fait désormais partie du kit [AWS SDK for .NET](#). Pour commencer à utiliser Amazon Cognito dans le AWS SDK for .NET, consultez la section relative au fournisseur d'informations d'[identification Amazon Cognito](#) dans AWS SDK for .NET le guide du développeur. Ou consultez le [centre de développement d'Amplify](#) pour découvrir les options permettant de créer une application avec. AWS Amplify

```
// create a service client that uses credentials provided by Cognito
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials, REGION);
```

Le fournisseur d'informations d'identification communique avec Amazon Cognito, récupérant à la fois l'identifiant unique pour les utilisateurs authentifiés et non authentifiés ainsi que les informations d'identification temporaires à privilèges limités AWS pour le SDK mobile. AWS Les informations d'identification récupérées sont valides pendant une heure. Le fournisseur les actualise lorsqu'elles expirent.

Xamarin

Le kit [AWS SDK for Xamarin](#) fait désormais partie du kit [AWS SDK for .NET](#). Pour commencer à utiliser Amazon Cognito dans le AWS SDK for .NET, consultez la section relative au fournisseur

d'informations d'[identification Amazon Cognito](#) dans AWS SDK for .NET le guide du développeur. Ou consultez le [centre de développement d'Amplify](#) pour découvrir les options permettant de créer une application avec. AWS Amplify

```
// create a service client that uses credentials provided by Cognito
var client = new AmazonDynamoDBClient(credentials, REGION)
```

Le fournisseur d'informations d'identification communique avec Amazon Cognito, récupérant à la fois l'identifiant unique pour les utilisateurs authentifiés et non authentifiés ainsi que les informations d'identification temporaires à privilèges limités AWS pour le SDK mobile. Les informations d'identification récupérées sont valides pendant une heure. Le fournisseur les actualise lorsqu'elles expirent.

Fournisseurs d'identité externes aux groupes d'identités

Avec la propriété `logins`, vous pouvez définir des informations d'identification reçues d'un fournisseur d'identité. En outre, vous pouvez associer un pool d'identités à plusieurs IdPs. Par exemple, vous pouvez définir des jetons Facebook et Google dans la propriété `logins` afin d'associer l'identité Amazon Cognito unique aux connexions de ces deux fournisseurs d'identité. L'utilisateur peut s'authentifier avec l'un ou l'autre des comptes, mais Amazon Cognito renvoie le même identifiant d'utilisateur.

Les instructions suivantes vous guident tout au long de l'authentification avec les pools IdPs d'identités pris en charge par Amazon Cognito.

Rubriques

- [Configuration de Facebook en tant qu'IdP de pool d'identités](#)
- [Configuration de Login with Amazon en tant qu'IdP de pool d'identités](#)
- [Configuration de Google en tant qu'IdP de pool d'identités](#)
- [Configuration de la connexion avec Apple en tant qu'IdP du pool d'identités](#)
- [Configuration d'un fournisseur OIDC en tant qu'IdP de pool d'identités](#)
- [Configuration d'un fournisseur SAML en tant qu'IdP de pool d'identités](#)

Configuration de Facebook en tant qu'IdP de pool d'identités

Les groupes d'identités Amazon Cognito s'intègrent avec Facebook pour fournir une authentification fédérée pour les utilisateurs de votre application mobile. Cette section explique comment enregistrer et configurer votre application avec Facebook comme fournisseur d'identité.

Configuration de Facebook

Enregistrez votre application auprès de Facebook avant de pouvoir authentifier des utilisateurs Facebook et interagir avec les API de Facebook.

Le [portail des développeurs Facebook](#) vous aide à configurer votre application. Procédez comme suit avant d'intégrer Facebook dans votre groupe d'identités Amazon Cognito :

Configuration de Facebook

1. Sur le [portail des développeurs Facebook](#), connectez-vous avec vos informations d'identification Facebook.
2. Dans le menu Apps (Applications), sélectionnez Add a New App (Ajouter une nouvelle application).
3. Sélectionnez une plateforme et terminer le processus de démarrage rapide.

Android

Pour plus d'informations sur la façon d'intégrer des applications Android avec Facebook Login, consultez le [Guide de démarrage Facebook](#).

iOS : Objective-C

Pour plus d'informations sur la façon d'intégrer des applications iOS Objective-C avec Facebook Login, consultez le [Guide de démarrage Facebook](#).

iOS : Swift

Pour plus d'informations sur la façon d'intégrer des applications iOS Swift avec Facebook Login, consultez le [Guide de démarrage Facebook](#).

JavaScript

Pour plus d'informations sur la façon d'intégrer des applications JavaScript Web à Facebook Login, consultez le [guide de démarrage de Facebook](#).

Unity

Pour plus d'informations sur la façon d'intégrer des applications Unity avec Facebook Login, consultez le [Guide de démarrage Facebook](#).

Xamarin

Pour ajouter l'authentification Facebook, commencez par suivre le flux approprié ci-dessous afin d'intégrer le kit SDK Facebook à votre application. Les groupes d'identité Amazon Cognito utilisent le jeton d'accès Facebook pour générer un identifiant utilisateur unique associé à une identité Amazon Cognito.

- [Kit SDK Facebook iOS par Xamarin](#)
- [Kit SDK Facebook Android par Xamarin](#)

Configuration d'un fournisseur d'identité dans la console des réserves d'identités Amazon Cognito

Utilisez la procédure suivante pour configurer votre fournisseur d'identité.

Pour ajouter un fournisseur d'identité (IdP) Facebook

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Facebook.
5. Saisissez l'ID de l'application du projet OAuth que vous avez créé sur [Meta for Developers](#). Pour plus d'informations, consultez [Facebook Login](#) (langue française non garantie) dans la documentation de Meta for Developers.
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.

- i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
8. Sélectionnez Enregistrer les modifications.

Utilisation de Facebook

Android

Pour ajouter l'authentification Facebook, commencez par suivre le [Guide Facebook](#) afin d'intégrer le kit SDK Facebook à votre application. Ajoutez ensuite un [bouton Login with Facebook \(Se connecter avec Facebook\)](#) dans votre interface utilisateur Android. Le kit SDK Facebook utilise un objet session pour suivre son état. Amazon Cognito utilise le jeton d'accès de cet objet de session pour authentifier l'utilisateur, générer l'identifiant unique et, si nécessaire, accorder à l'utilisateur l'accès à d'autres ressources. AWS

Après avoir authentifié votre utilisateur avec le kit SDK Facebook, ajoutez le jeton de session au fournisseur d'informations d'identification Amazon Cognito.

Kit SDK Facebook 4.0 ou version ultérieure :

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("graph.facebook.com", AccessToken.getCurrentAccessToken().getToken());
credentialsProvider.setLogins(logins);
```

Kit SDK Facebook avant la version 4.0 :

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("graph.facebook.com", Session.getActiveSession().getAccessToken());
credentialsProvider.setLogins(logins);
```

Le processus de connexion Facebook initialise une session singleton dans son kit SDK. L'objet de session Facebook contient un jeton OAuth qu'Amazon Cognito utilise pour AWS générer les informations d'identification de votre utilisateur final authentifié. Amazon Cognito utilise également ce jeton pour rechercher dans votre base de données utilisateur l'existence d'un utilisateur qui correspond à cette identité Facebook particulière. Si l'utilisateur existe déjà, l'API renvoie l'identifiant correspondant. Dans le cas contraire, elle renvoie un nouvel identifiant. Les identifiants sont automatiquement mis en cache par le kit SDK client sur l'appareil local.

Note

Après avoir défini le mappage des connexions, appelez `refresh` ou récupérez `get` les AWS informations d'identification.

iOS : Objective-C

Pour ajouter l'authentification Facebook, commencez par suivre le [Guide Facebook](#) afin d'intégrer le kit SDK Facebook à votre application. Ajoutez ensuite un [bouton « Login with Facebook »](#) à votre interface utilisateur. Le kit SDK Facebook utilise un objet session pour suivre son état. Amazon Cognito utilise le jeton d'accès de cet objet session pour authentifier l'utilisateur et le lier à un groupe d'identités Amazon Cognito (Identités fédérées) unique.

Pour fournir le jeton d'accès Facebook à Amazon Cognito, implémentez le protocole [AWSIdentityProviderManager](#).

Quand vous implémentez la méthode `logins`, renvoyez un dictionnaire contenant `AWSIdentityProviderFacebook`. Ce dictionnaire fait office de clé et le jeton d'accès actuel de l'utilisateur Facebook authentifié fait office de valeur, comme dans l'exemple de code suivant.

```

- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    FBSDKAccessToken* fbToken = [FBSDKAccessToken currentAccessToken];
    if(fbToken){
        NSString *token = fbToken.tokenString;
        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook : token }]];
    }else{
        return [AWSTask taskWithError:[NSError errorWithDomain:@"Facebook Login"
                                                    code:-1
                                                    userInfo:@{@"error":@"No current
Facebook access token"}]];
    }
}

```

Lorsque vous instanciez `AWSCognitoCredentialsProvider`, transmettez la classe qui implémente `AWSIdentityProviderManager` comme valeur de `identityProviderManager` dans le constructeur. Pour plus d'informations, rendez-vous sur la page de [AWSCognitoCredentialsProvider](#) référence et choisissez `initWithRegionType : identityPoolId : identityProviderManager`.

iOS : Swift

Pour ajouter l'authentification Facebook, commencez par suivre le [Guide Facebook](#) afin d'intégrer le kit SDK Facebook à votre application. Ajoutez ensuite un [bouton « Login with Facebook »](#) à votre interface utilisateur. Le kit SDK Facebook utilise un objet session pour suivre son état. Amazon Cognito utilise le jeton d'accès de cet objet session pour authentifier l'utilisateur et le lier à un groupe d'identités Amazon Cognito (Identités fédérées) unique.

Pour fournir le jeton d'accès Facebook à Amazon Cognito, implémentez le protocole [AWSIdentityProviderManager](#).

Quand vous implémentez la méthode `logins`, renvoyez un dictionnaire contenant `AWSIdentityProviderFacebook`. Ce dictionnaire fait office de clé et le jeton d'accès actuel de l'utilisateur Facebook authentifié fait office de valeur, comme dans l'exemple de code suivant.

```

class FacebookProvider: NSObject, AWSIdentityProviderManager {
    func logins() -> AWSTask<NSDictionary> {
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error:NSError(domain: "Facebook Login", code: -1 , userInfo:
["Facebook" : "No current Facebook access token"]))
    }
}

```

```
}  
}
```

Lorsque vous instanciez `AWSCognitoCredentialsProvider`, transmettez la classe qui implémente `AWSCognitoIdentityProviderManager` comme valeur de `identityProviderManager` dans le constructeur. Pour plus d'informations, rendez-vous sur la page de [AWSCognitoCredentialsProvider](#) référence et choisissez `initWithRegionType : identityPoolId : identityProviderManager`.

JavaScript

Pour ajouter l'authentification Facebook, suivez les instructions fournies dans la page [Facebook Login pour le web](#) et ajoutez le bouton Login with Facebook (Se connecter avec Facebook) sur votre site web. Le kit SDK Facebook utilise un objet session pour suivre son état. Amazon Cognito utilise le jeton d'accès de cet objet de session pour authentifier l'utilisateur, générer l'identifiant unique et, si nécessaire, accorder à l'utilisateur l'accès à d'autres ressources. AWS

Après avoir authentifié votre utilisateur avec le kit SDK Facebook, ajoutez le jeton de session au fournisseur d'informations d'identification Amazon Cognito.

```
FB.login(function (response) {  
  
    // Check if the user logged in successfully.  
    if (response.authResponse) {  
  
        console.log('You are now logged in.');  
        // Add the Facebook access token to the Amazon Cognito credentials login map.  
        AWS.config.credentials = new AWS.CognitoIdentityCredentials({  
            IdentityPoolId: 'IDENTITY_POOL_ID',  
            Logins: {  
                'graph.facebook.com': response.authResponse.accessToken  
            }  
        });  
  
        // Obtain AWS credentials  
        AWS.config.credentials.get(function(){  
            // Access AWS resources here.  
        });  
  
    } else {  
        console.log('There was a problem logging you in.');    }  
});
```

```
}  
  
});
```

Le SDK Facebook obtient un jeton OAuth qu'Amazon Cognito utilise pour générer les informations d'AWS identification de votre utilisateur final authentifié. Amazon Cognito utilise également ce jeton pour rechercher dans votre base de données utilisateur l'existence d'un utilisateur correspondant à cette identité Facebook particulière. Si l'utilisateur existe déjà, l'API renvoie l'identifiant correspondant. Dans le cas contraire, elle renvoie un nouvel identifiant. Les identifiants sont automatiquement mis en cache par le kit SDK client sur le dispositif local.

Note

Après avoir configuré le mappage des connexions, effectuez un appel à `refresh` ou `get` pour obtenir les informations d'identification. Pour un exemple de code, voir « Cas d'utilisation 17, intégration de groupes d'utilisateurs à Cognito Identity » dans le [JavaScript fichier README](#).

Unity

Pour ajouter l'authentification Facebook, commencez par suivre le [Guide Facebook](#) afin d'intégrer le kit SDK Facebook à votre application. Amazon Cognito utilise le jeton d'accès Facebook de l'objet FB pour générer un identifiant d'utilisateur unique associé à une identité Amazon Cognito.

Après avoir authentifié votre utilisateur avec le kit SDK Facebook, ajoutez le jeton de session au fournisseur d'informations d'identification Amazon Cognito :

```
void Start()  
{  
    FB.Init(delegate() {  
        if (FB.IsLoggedIn) { //User already logged in from a previous session  
            AddFacebookTokenToCognito();  
        } else {  
            FB.Login ("email", FacebookLoginCallback);  
        }  
    });  
}  
  
void FacebookLoginCallback(FBResult result)  
{
```

```
if (FB.IsLoggedIn)
{
    AddFacebookTokenToCognito();
}
else
{
    Debug.Log("FB Login error");
}
}

void AddFacebookTokenToCognito()
{
    credentials.AddLogin ("graph.facebook.com",
    AccessToken.CurrentAccessToken.TokenString);
}
```

Avant d'utiliser `FB.AccessToken`, appelez `FB.Login()` et assurez-vous que `FB.IsLoggedIn` a pour valeur `true`.

Xamarin

Xamarin pour Android :

```
public void InitializeFacebook() {
    FacebookSdk.SdkInitialize(this.ApplicationContext);
    callbackManager = CallbackManagerFactory.Create();
    LoginManager.Instance.RegisterCallback(callbackManager, new FacebookCallback <
LoginResult > () {
    HandleSuccess = loginResult = > {
        var accessToken = loginResult.AccessToken;
        credentials.AddLogin("graph.facebook.com", accessToken.Token);
        //open new activity
    },
    HandleCancel = () = > {
        //throw error message
    },
    HandleError = loginError = > {
        //throw error message
    }
});
    LoginManager.Instance.LoginWithReadPermissions(this, new List < string > {
        "public_profile"
    });
}
```

```
}
```

Xamarin pour iOS :

```
public void InitializeFacebook() {
    LoginManager login = new LoginManager();
    login.LogInWithReadPermissions(readPermissions.ToArray(),
    delegate(LoginManagerLoginResult result, NSError error) {
        if (error != null) {
            //throw error message
        } else if (result.IsCancelled) {
            //throw error message
        } else {
            var accessToken = loginResult.AccessToken;
            credentials.AddLogin("graph.facebook.com", accessToken.Token);
            //open new view controller
        }
    });
}
```

Configuration de Login with Amazon en tant qu'IdP de pool d'identités

Amazon Cognito s'intègre avec Login with Amazon afin de fournir une authentification fédérée pour vos utilisateurs d'applications mobiles et web. Cette section explique comment enregistrer et configurer votre application avec Login with Amazon comme fournisseur d'identité.

Configurez Login with Amazon pour utiliser Amazon Cognito dans le [portail des développeurs](#). Pour plus d'informations, consultez [Configuration de Login with Amazon](#) dans les questions fréquentes sur Login with Amazon.

Note

Pour intégrer Login with Amazon dans une application Xamarin, suivez les instructions fournies dans la page [Bien démarrer avec Xamarin](#).

Note

Vous ne pouvez pas intégrer Login with Amazon en mode natif sur la plateforme Unity. Au lieu de cela, utilisez une vue web et suivez le processus de connexion dans le navigateur.

Configuration de Login with Amazon

Implémentation de Login with Amazon

Dans le [portail pour développeurs Amazon](#), vous pouvez configurer une application OAuth pour l'intégrer à votre groupe d'identités, trouver la documentation Login with Amazon et télécharger les kits SDK. Choisissez Developer console (Console pour développeurs), puis Login with Amazon dans le portail des développeurs. Vous pouvez créer un profil de sécurité pour votre application, puis générer des mécanismes d'authentification Login with Amazon dans votre application. Consultez [Obtention des informations d'identification](#) pour plus d'informations sur la façon d'intégrer l'authentification Login with Amazon avec votre application.

Amazon émet un ID client OAuth 2.0 pour votre nouveau profil de sécurité. Vous pouvez trouver cet ID client dans l'onglet Web Settings (Paramètres web) du profil de sécurité. Saisissez l'ID du profil de sécurité dans le champ ID d'application du fournisseur d'identité Login with Amazon de votre réserve d'identités.

Note

Saisissez l'ID du profil de sécurité dans le champ ID d'application du fournisseur d'identité Login with Amazon de votre réserve d'identités. De leur côté, les groupes d'utilisateurs utilisent l'ID client.

Configurer le fournisseur externe dans la console Amazon Cognito

Pour ajouter un fournisseur d'identité (IdP) Login with Amazon

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Login with Amazon.
5. Saisissez l'ID de l'application du projet OAuth que vous avez créé sur [Login with Amazon](#). Pour plus d'informations, consultez [Login with Amazon Documentation](#) (langue française non garantie).
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.

- Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
- 7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
- 8. Sélectionnez Enregistrer les modifications.

Utiliser Login with Amazon : Android

Après avoir authentifié la connexion Amazon, vous pouvez transmettre le jeton au fournisseur d'informations d'identification Amazon Cognito dans la méthode `onSuccess` de l'interface.

`TokenListener` Ce code se présente sous la forme suivante :

```
@Override
public void onSuccess(Bundle response) {
    String token = response.getString(AuthzConstants.BUNDLE_KEY.TOKEN.val);
    Map<String, String> logins = new HashMap<String, String>();
```

```
logins.put("www.amazon.com", token);
credentialsProvider.setLogins(logins);
}
```

Utiliser Login with Amazon : iOS - Objective-C

Après avoir authentifié la connexion Amazon, vous pouvez transmettre le jeton au fournisseur d'informations d'identification Amazon Cognito selon `requestDidSucceed` la méthode de l'`AMZN : AccessTokenDelegate`

```
- (void)requestDidSucceed:(APIResult \*)apiResult {
    if (apiResult.api == kAPIAuthorizeUser) {
        [AIMobileLib getAccessTokenForScopes:[NSArray arrayWithObject:@"profile"]
withOverrideParams:nil delegate:self];
    }
    else if (apiResult.api == kAPIGetAccessToken) {
        credentialsProvider.logins = @[ @(AWSCognitoLoginProviderKeyLoginWithAmazon):
apiResult.result ];
    }
}}
```

Utiliser Login with Amazon : iOS - Swift

Après avoir authentifié la connexion Amazon, vous pouvez transmettre le jeton au fournisseur d'informations d'identification Amazon Cognito dans la méthode `requestDidSucceed` de `AMZNAccessTokenDelegate` :

```
func requestDidSucceed(apiResult: APIResult!) {
    if apiResult.api == API.AuthorizeUser {
        AIMobileLib.getAccessTokenForScopes(["profile"], withOverrideParams: nil,
delegate: self)
    } else if apiResult.api == API.GetAccessToken {
        credentialsProvider.logins =
[AWSCognitoLoginProviderKey.LoginWithAmazon.rawValue: apiResult.result]
    }
}
```

Utilisez Login with Amazon : JavaScript

Une fois que l'utilisateur s'authentifie avec Login with Amazon et qu'il est redirigé vers votre site web, le jeton d'accès Login with Amazon est fourni dans la chaîne de requête. Ajoutez ce jeton à la carte de connexions des informations d'identification.

```
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: 'IDENTITY_POOL_ID',
  Logins: {
    'www.amazon.com': 'Amazon Access Token'
  }
});
```

Utiliser Login with Amazon : Xamarin

Xamarin pour Android

```
AmazonAuthorizationManager manager = new AmazonAuthorizationManager(this,
Bundle.Empty);

var tokenListener = new APIListener {
  Success = response => {
    // Get the auth token
    var token = response.GetString(AuthzConstants.BUNDLE_KEY.Token.Val);
    credentials.AddLogin("www.amazon.com", token);
  }
};

// Try and get existing login
manager.GetToken(new[] {
  "profile"
}, tokenListener);
```

Xamarin pour iOS

Dans `AppDelegate.cs`, insérer ce qui suit :

```
public override bool OpenUrl (UIApplication application, NSURL url, string
sourceApplication, NSObject annotation)
{
  // Pass on the url to the SDK to parse authorization code from the url
```

```
bool isValidRedirectSignInURL = AIMobileLib.HandleOpenUrl (url, sourceApplication);
if(!isValidRedirectSignInURL)
    return false;

// App may also want to handle url
return true;
}
```

Puis, dans `ViewController.cs`, procédez comme suit :

```
public override void ViewDidLoad ()
{
    base.LoadView ();

    // Here we create the Amazon Login Button
    btnLogin = UIButton.FromType (UIButtonType.RoundedRect);
    btnLogin.Frame = new RectangleF (55, 206, 209, 48);
    btnLogin.SetTitle ("Login using Amazon", UIControlState.Normal);
    btnLogin.TouchUpInside += (sender, e) => {
        AIMobileLib.AuthorizeUser (new [] { "profile"}, new AMZNAuthorizationDelegate
    ());
    };
    View.AddSubview (btnLogin);
}

// Class that handles Authentication Success/Failure
public class AMZNAuthorizationDelegate : AIAAuthenticationDelegate
{
    public override void RequestDidSucceed(ApiResult apiResult)
    {
        // Your code after the user authorizes application for requested scopes
        var token = apiResult["access_token"];
        credentials.AddLogin("www.amazon.com", token);
    }

    public override void RequestDidFail(ApiError errorResponse)
    {
        // Your code when the authorization fails
        InvokeOnMainThread(() => new UIAlertView("User Authorization Failed",
errorResponse.Error.Message, null, "Ok", null).Show());
    }
}
```

Configuration de Google en tant qu'IdP de pool d'identités

Amazon Cognito s'intègre à Google pour fournir une authentification fédérée pour les utilisateurs de votre application mobile. Cette section explique comment enregistrer et configurer votre application avec Google comme fournisseur d'identité.

Android

Note

Si votre application utilise Google et qu'elle est disponible sur plusieurs plateformes mobiles, vous devez configurer Google en tant que [fournisseur OpenID Connect](#). Ajoutez tous les identifiants client créés en tant que valeurs supplémentaires pour le public, pour une meilleure intégration. Pour en savoir plus sur le modèle d'identité à plusieurs clients de Google, consultez [cette page](#).

Configuration de Google

Pour activer la connexion Google pour Android, créez un projet de console Google Developers pour votre application.

1. Accédez à la [console Google Developers](#) et créez un projet.
2. Choisissez API et services, puis Écran de consentement OAuth. Personnalisez les informations que Google montre à vos utilisateurs quand il leur demande de consentir à partager leurs données de profil avec votre application.
3. Choisissez Identifiants, puis Créer des identifiants. Choisissez ID client OAuth. Sélectionnez Android comme Type d'application. Créez un ID client distinct pour chaque plateforme où vous développez votre application.
4. Dans Identifiants, choisissez Gérer les comptes de service. Choisissez Créer un compte de service. Saisissez les détails de votre compte de service, puis choisissez Créer et continuer.
5. Accordez au compte de service l'accès à votre projet. Accordez aux utilisateurs l'accès au compte de service selon les besoins de votre application.
6. Choisissez votre nouveau compte de service, choisissez l'onglet Clés et Ajouter une clé. Créez et téléchargez une nouvelle clé JSON.

Pour plus d'informations sur l'utilisation de la console Google Developers, consultez [Créer et gérer des projets](#) dans la documentation Google Cloud.

Pour plus d'informations sur la façon d'intégrer Google à votre application Android, consultez [Authentifier les utilisateurs avec Sign in with Google](#) dans la documentation Google Identity.

Pour ajouter un fournisseur d'identité (IdP) Google

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Google.
5. Entrez l'ID du client du projet OAuth que vous avez créé sur [Google Cloud Platform](#). Pour plus d'informations, consultez [Configuration d'OAuth 2.0](#) (langue française non garantie) dans l'aide de la console Google Cloud Platform.
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.

- b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
8. Sélectionnez Enregistrer les modifications.

Utilisation de Google

Pour activer la connexion avec Google dans votre application, suivez les instructions fournies dans la [documentation Google pour Android](#). Lorsqu'un utilisateur se connecte, il demande un jeton d'authentification OpenID Connect à Google. Amazon Cognito utilise alors ce jeton pour authentifier l'utilisateur et générer un identifiant unique.

L'exemple de code suivant illustre comment récupérer le jeton d'authentification auprès du service Google Play :

```
GooglePlayServicesUtil.isGooglePlayServicesAvailable(getApplicationContext());
AccountManager am = AccountManager.get(this);
Account[] accounts = am.getAccountsByType(GoogleAuthUtil.GOOGLE_ACCOUNT_TYPE);
String token = GoogleAuthUtil.getToken(getApplicationContext(), accounts[0].name,
    "audience:server:client_id:YOUR_GOOGLE_CLIENT_ID");
Map<String, String> logins = new HashMap<String, String>();
logins.put("accounts.google.com", token);
credentialsProvider.setLogins(logins);
```

iOS : Objective-C

Note

Si votre application utilise Google et qu'elle est disponible sur plusieurs plateformes mobiles, configurez Google en tant que [fournisseur OpenID Connect](#). Ajoutez tous les identifiants client créés en tant que valeurs supplémentaires pour le public, pour une meilleure intégration. Pour en savoir plus sur le modèle d'identité à plusieurs clients de Google, consultez [cette page](#).

Configuration de Google

Pour activer la connexion Google pour iOS, créez un projet de console Google Developers pour votre application.

1. Accédez à la [console Google Developers](#) et créez un projet.
2. Choisissez API et services, puis Écran de consentement OAuth. Personnalisez les informations que Google montre à vos utilisateurs quand il leur demande de consentir à partager leurs données de profil avec votre application.
3. Choisissez Identifiants, puis Créer des identifiants. Choisissez ID client OAuth. Sélectionnez iOS comme Type d'application. Créez un ID client distinct pour chaque plateforme où vous développez votre application.
4. Dans Identifiants, choisissez Gérer les comptes de service. Choisissez Créer un compte de service. Saisissez les détails de votre compte de service, puis choisissez Créer et continuer.
5. Accordez au compte de service l'accès à votre projet. Accordez aux utilisateurs l'accès au compte de service selon les besoins de votre application.
6. Choisissez votre nouveau compte de service. Choisissez l'onglet Clés et Ajouter une clé. Créez et téléchargez une nouvelle clé JSON.

Pour plus d'informations sur l'utilisation de la console Google Developers, consultez [Créer et gérer des projets](#) dans la documentation Google Cloud.

Pour plus d'informations sur l'intégration de Google dans votre application iOS, consultez [Google Sign-In for iOS](#) dans la documentation Google Identity.

Pour ajouter un fournisseur d'identité (IdP) Google

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Google.
5. Entrez l'ID du client du projet OAuth que vous avez créé sur [Google Cloud Platform](#). Pour plus d'informations, consultez [Configuration d'OAuth 2.0](#) (langue française non garantie) dans l'aide de la console Google Cloud Platform.

6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
8. Sélectionnez Enregistrer les modifications.

Utilisation de Google

Pour activer la connexion avec Google dans votre application, reportez-vous à la [documentation Google pour iOS](#). Une authentification réussie produit un jeton d'authentification OpenID Connect, qu'Amazon Cognito utilise pour authentifier l'utilisateur et générer un identifiant unique.

Une authentification réussie produit un objet `GTM0Auth2Authentication` contenant un `id_token` qu'Amazon Cognito utilise pour authentifier l'utilisateur et générer un identifiant unique :

```
- (void)finishedWithAuth: (GTMOAuth2Authentication *)auth error: (NSError *) error {
    NSString *idToken = [auth.parameters objectForKey:@"id_token"];
    credentialsProvider.logins = @{ @(AWSCognitoLoginProviderKeyGoogle): idToken };
}
```

iOS : Swift

Note

Si votre application utilise Google et qu'elle est disponible sur plusieurs plateformes mobiles, configurez Google en tant que [fournisseur OpenID Connect](#). Ajoutez tous les identifiants client créés en tant que valeurs supplémentaires pour le public, pour une meilleure intégration. Pour en savoir plus sur le modèle d'identité à plusieurs clients de Google, consultez [cette page](#).

Configuration de Google

Pour activer la connexion Google pour iOS, créez un projet de console Google Developers pour votre application.

1. Accédez à la [console Google Developers](#) et créez un projet.
2. Choisissez API et services, puis Écran de consentement OAuth. Personnalisez les informations que Google montre à vos utilisateurs quand il leur demande de consentir à partager leurs données de profil avec votre application.
3. Choisissez Identifiants, puis Créer des identifiants. Choisissez ID client OAuth. Sélectionnez iOS comme Type d'application. Créez un ID client distinct pour chaque plateforme où vous développez votre application.
4. Dans Identifiants, choisissez Gérer les comptes de service. Choisissez Créer un compte de service. Saisissez les détails de votre compte de service, puis choisissez Créer et continuer.
5. Accordez au compte de service l'accès à votre projet. Accordez aux utilisateurs l'accès au compte de service selon les besoins de votre application.
6. Choisissez votre nouveau compte de service, choisissez l'onglet Clés et Ajouter une clé. Créez et téléchargez une nouvelle clé JSON.

Pour plus d'informations sur l'utilisation de la console Google Developers, consultez [Créer et gérer des projets](#) dans la documentation Google Cloud.

Pour plus d'informations sur l'intégration de Google dans votre application iOS, consultez [Google Sign-In for iOS](#) dans la documentation Google Identity.

Choisissez Manage Identity Pools (Gérer les groupes d'identité) dans la [page d'accueil de la console Amazon Cognito](#).

Configuration du fournisseur externe dans la console Amazon Cognito

1. Choisissez le nom du groupe d'identités dans lequel vous souhaitez activer Google comme fournisseur externe. La page Dashboard (Tableau de bord) correspondant à votre groupe d'identités s'affiche.
2. Dans l'angle supérieur droit de la page Dashboard (Tableau de bord), choisissez Edit identity pool (Modifier le groupe d'identités). La page Edit identity pool (Modifier le groupe d'identités) s'affiche.
3. Faites défiler l'affichage vers le bas et choisissez Authentication providers (Fournisseurs d'authentification) pour développer cette option.
4. Choisissez l'onglet Google.
5. Sélectionnez Unlock (Déverrouiller).
6. Saisissez l'ID client Google que vous avez obtenu auprès de Google, puis choisissez Save Changes (Enregistrer les modifications).

Utilisation de Google

Pour activer la connexion avec Google dans votre application, reportez-vous à la [documentation Google pour iOS](#). Une authentification réussie produit un jeton d'authentification OpenID Connect, qu'Amazon Cognito utilise pour authentifier l'utilisateur et générer un identifiant unique.

Une authentification réussie produit un objet `GTM0Auth2Authentication` qui contient un jeton `id_token`. Amazon Cognito utilise ce jeton pour authentifier l'utilisateur et générer un identifiant unique :

```
func finishedWithAuth(auth: GTM0Auth2Authentication!, error: NSError!) {
    if error != nil {
        print(error.localizedDescription)
    }
    else {
        let idToken = auth.parameters.objectForKey("id_token")
        credentialsProvider.logins = [AWSCognitoLoginProviderKey.Google.rawValue:
idToken!]
    }
}
```

```
}  
}
```

JavaScript

Note

Si votre application utilise Google et qu'elle est disponible sur plusieurs plateformes mobiles, vous devez configurer Google en tant que [fournisseur OpenID Connect](#). Ajoutez tous les identifiants client créés en tant que valeurs supplémentaires pour le public, pour une meilleure intégration. Pour en savoir plus sur le modèle d'identité à plusieurs clients de Google, consultez [cette page](#).

Configuration de Google

Pour activer Google Sign-in pour une application JavaScript Web, créez un projet de console Google Developers pour votre application.

1. Accédez à la [console Google Developers](#) et créez un projet.
2. Choisissez API et services, puis Écran de consentement OAuth. Personnalisez les informations que Google montre à vos utilisateurs quand il leur demande de consentir à partager leurs données de profil avec votre application.
3. Choisissez Identifiants, puis Créer des identifiants. Choisissez ID client OAuth. Sélectionnez Application web comme Type d'application. Créez un ID client distinct pour chaque plateforme où vous développez votre application.
4. Dans Identifiants, choisissez Gérer les comptes de service. Choisissez Créer un compte de service. Saisissez les détails de votre compte de service, puis choisissez Créer et continuer.
5. Accordez au compte de service l'accès à votre projet. Accordez aux utilisateurs l'accès au compte de service selon les besoins de votre application.
6. Choisissez votre nouveau compte de service, choisissez l'onglet Clés et Ajouter une clé. Créez et téléchargez une nouvelle clé JSON.

Pour plus d'informations sur l'utilisation de la console Google Developers, consultez [Créer et gérer des projets](#) dans la documentation Google Cloud.

Pour plus d'informations sur l'intégration de Google dans votre application web, consultez [Se connecter avec Google](#) dans la documentation Google Identity.

Configuration du fournisseur externe dans la console Amazon Cognito

Pour ajouter un fournisseur d'identité (IdP) Google

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Google.
5. Entrez l'ID du client du projet OAuth que vous avez créé sur [Google Cloud Platform](#). Pour plus d'informations, consultez [Configuration d'OAuth 2.0](#) (langue française non garantie) dans l'aide de la console Google Cloud Platform.
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.

- b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
8. Sélectionnez Enregistrer les modifications.

Utilisation de Google

Pour activer la connexion avec Google dans votre application, reportez-vous à la [documentation Google pour le web](#).

Une authentification réussie produit un objet de réponse contenant un jeton `id_token` qu'Amazon Cognito utilise pour authentifier l'utilisateur et générer un identifiant unique :

```
function signinCallback(authResult) {
  if (authResult['status']['signed_in']) {

    // Add the Google access token to the Amazon Cognito credentials login map.
    AWS.config.credentials = new AWS.CognitoIdentityCredentials({
      IdentityPoolId: 'IDENTITY_POOL_ID',
      Logins: {
        'accounts.google.com': authResult['id_token']
      }
    });

    // Obtain AWS credentials
    AWS.config.credentials.get(function(){
      // Access AWS resources here.
    });
  }
}
```

Unity

Configuration de Google

Pour activer la connexion Google pour une application Unity, créez un projet de console Google Developers pour votre application.

1. Accédez à la [console Google Developers](#) et créez un projet.
2. Choisissez API et services, puis Écran de consentement OAuth. Personnalisez les informations que Google montre à vos utilisateurs quand il leur demande de consentir à partager leurs données de profil avec votre application.
3. Choisissez Identifiants, puis Créer des identifiants. Choisissez ID client OAuth. Sélectionnez Application web comme Type d'application. Créez un ID client distinct pour chaque plateforme où vous développez votre application.
4. Pour Unity, créez un ID client OAuth pour Android, et un autre pour iOS.
5. Dans Identifiants, choisissez Gérer les comptes de service. Choisissez Créer un compte de service. Saisissez les détails de votre compte de service, puis choisissez Créer et continuer.
6. Accordez au compte de service l'accès à votre projet. Accordez aux utilisateurs l'accès au compte de service selon les besoins de votre application.
7. Choisissez votre nouveau compte de service, choisissez l'onglet Clés et Ajouter une clé. Créez et téléchargez une nouvelle clé JSON.

Pour plus d'informations sur l'utilisation de la console Google Developers, consultez [Créer et gérer des projets](#) dans la documentation Google Cloud.

Création d'un fournisseur OpenID dans la console IAM

1. Créez un fournisseur OpenID dans la console IAM. Pour obtenir des informations sur la façon de configurer un fournisseur OpenID, consultez la page [Utilisation des fournisseurs d'identité OpenID Connect](#).
2. Lorsque vous êtes invité à indiquer l'URL du fournisseur, saisissez "https://accounts.google.com".
3. Lorsque vous êtes invité à saisir une valeur dans le champ Audience (Public), indiquez l'un des trois ID client créés lors des étapes précédentes.
4. Choisissez le nom du fournisseur et ajoutez deux autres publics ciblés avec les deux autres ID client.

Configuration du fournisseur externe dans la console Amazon Cognito

Choisissez Manage Identity Pools (Gérer les groupes d'identité) dans la [page d'accueil de la console Amazon Cognito](#).

Pour ajouter un fournisseur d'identité (IdP) Google

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Google.
5. Entrez l'ID du client du projet OAuth que vous avez créé sur [Google Cloud Platform](#). Pour plus d'informations, consultez [Configuration d'OAuth 2.0](#) (langue française non garantie) dans l'aide de la console Google Cloud Platform.
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que

vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.

8. Sélectionnez Enregistrer les modifications.

Installation du plugin Unity Google

1. Ajoutez le [plugin Google Play Games pour Unity](#) au projet Unity.
2. Dans Unity, à partir du menu Windows, utilisez les trois identifiants créés pour les plateformes Android et iOS pour configurer le plugin.

Utilisation de Google

L'exemple de code suivant illustre comment récupérer le jeton d'authentification auprès du service Google Play :

```
void Start()
{
    PlayGamesClientConfiguration config = new
    PlayGamesClientConfiguration.Builder().Build();
    PlayGamesPlatform.InitializeInstance(config);
    PlayGamesPlatform.DebugLogEnabled = true;
    PlayGamesPlatform.Activate();
    Social.localUser.Authenticate(GoogleLoginCallback);
}

void GoogleLoginCallback(bool success)
{
    if (success)
    {
        string token = PlayGamesPlatform.Instance.GetIdToken();
        credentials.AddLogin("accounts.google.com", token);
    }
    else
    {
        Debug.LogError("Google login failed. If you are not running in an actual Android/
        iOS device, this is expected.");
    }
}
```

Xamarin

Note

Amazon Cognito ne prend pas en charge Google en mode natif sur la plateforme Xamarin. Elle nécessite actuellement l'utilisation d'une vue Web pour suivre la procédure de connexion dans le navigateur. Pour découvrir comment l'intégration de Google fonctionne avec les autres kits SDK, veuillez sélectionner une autre plateforme.

Pour activer la connexion avec Google dans votre application, authentifiez vos utilisateurs et obtenez d'eux un jeton OpenID Connect. Amazon Cognito utilise ce jeton pour générer un identifiant utilisateur unique associé à une identité Amazon Cognito. Malheureusement, le kit Google SDK pour Xamarin ne vous permet pas de récupérer le jeton OpenID Connect. Vous devez donc utiliser un autre client ou le flux web dans une vue web.

Une fois que vous avez ce jeton, vous pouvez le définir dans `CognitoAWSCredentials` :

```
credentials.AddLogin("accounts.google.com", token);
```

Note

Si votre application utilise Google et qu'elle est disponible sur plusieurs plateformes mobiles, vous devez configurer Google en tant que [fournisseur OpenID Connect](#). Ajoutez tous les identifiants client créés en tant que valeurs supplémentaires pour le public, pour une meilleure intégration. Pour en savoir plus sur le modèle d'identité à plusieurs clients de Google, consultez [cette page](#).

Configuration de la connexion avec Apple en tant qu'IdP du pool d'identités

Amazon Cognito s'intègre avec Sign in with Apple pour fournir une authentification fédérée pour les utilisateurs de votre application mobile et de votre application web. Cette section explique comment enregistrer et configurer votre application avec la fonctionnalité Se connecter avec Apple comme fournisseur d'identité.

Pour ajouter la fonctionnalité Se connecter avec Apple en tant que fournisseur d'authentification à un groupe d'identités, vous devez mener à bien deux procédures. Tout d'abord, intégrez la

fonctionnalité Se connecter avec Apple dans une application, puis configurez Se connecter avec Apple dans les groupes d'identités. Pour en up-to-date savoir plus sur la configuration de Sign in with Apple, consultez la [section Configuration de votre environnement pour Sign in with Apple](#) dans la documentation destinée aux développeurs Apple.

Configurer Se connecter avec Apple

Pour configurer la fonctionnalité Se connecter avec Apple en tant que fournisseur d'identité, enregistrez votre application auprès d'Apple pour recevoir un ID client.

1. Créez un [compte développeur Apple](#).
2. [Connectez-vous](#) avec vos informations d'identification Apple.
3. Dans le volet de navigation de gauche, choisissez Certificates, IDs & Profiles (Certificats, ID et profils).
4. Dans le volet de navigation de gauche, choisissez Identifiants.
5. Dans la page Identifiants, choisissez l'icône +.
6. Dans la page Register a New Identifier (Enregistrer un nouvel identifiant), choisissez App IDs (ID application), puis Continue (Continuer).
7. Dans la page Register an App ID (Enregistrer un ID d'application), procédez comme suit :
 - a. Dans Description, saisissez une description.
 - b. Sous Bundle ID (ID de solution groupée), tapez un identifiant. Notez cet ID de solution groupée, car vous aurez besoin de cette valeur pour configurer Apple en tant que fournisseur dans le groupe d'identités.
 - c. Sous Capabilities (Capacités), choisissez Sign In with Apple (Connexion avec Apple), puis Edit (Modifier).
 - d. Sur la page Se connecter avec Apple : configuration de l'ID d'application, sélectionnez le paramètre approprié pour votre application. Ensuite, choisissez Save (Enregistrer).
 - e. Choisissez Continuer.
8. Dans la page Confirm your App ID (Confirmer votre ID d'application), choisissez Register (Inscrire).
9. Passez à l'étape 10 si vous souhaitez intégrer Sign in with Apple avec une application iOS native. L'étape 11 concerne les applications auxquelles vous souhaitez intégrer la connexion avec Apple JS.

10. Dans la page Identifiants, choisissez le menu ID d'application, puis ID de services. Choisissez l'icône +.
11. Dans la page Register a New Identifier (Enregistrer un nouvel identifiant), choisissez Services IDs (ID de service), puis Continue (Continuer).
12. Dans la page Register an App ID (Enregistrer un ID d'application), procédez comme suit :
 - a. Dans Description, saisissez une description.
 - b. Sous Identifier (Identifiant), saisissez un identifiant. Notez cet ID de services, car vous aurez besoin de cette valeur pour configurer Apple en tant que fournisseur dans votre groupe d'identités.
 - c. Sélectionnez Sign In with Apple (Connexion avec Apple), puis choisissez Configure (Configurer).
 - d. Dans la page Web Authentication Configuration (Configuration de l'authentification web), choisissez un Primary App ID (ID d'application principale). Sous URL de site web, choisissez l'icône +. Pour Domains and Subdomains (Domaines et Sous-domaines), saisissez le nom de domaine de votre application. Dans URL de retour, saisissez l'URL de rappel vers laquelle l'autorisation redirige l'utilisateur après son authentification via la fonctionnalité Se connecter avec Apple.
 - e. Choisissez Suivant.
 - f. Choisissez Continue (Continuer), puis Register (Enregistrer).
13. Dans le volet de navigation de gauche, choisissez Keys (Clés).
14. Dans la page Keys (Clés), choisissez l'icône +.
15. Dans la page Register a New Key (Enregistrer une nouvelle clé), procédez comme suit :
 - a. Sous Key Name (Nom de clé), saisissez un nom de clé.
 - b. Sélectionnez Sign In with Apple (Connexion avec Apple), puis choisissez Configure (Configurer).
 - c. Dans la page Configure Key (Configurer la clé), choisissez un Primary App ID (ID d'application principal), puis Save (Enregistrer).
 - d. Choisissez Continue (Continuer), puis Register (Enregistrer).

Note

Pour intégrer la connexion avec Apple à l'aide d'une application iOS native, consulter [Implementing User Authentication with Sign in with Apple \(Mise en place de l'authentification utilisateur à l'aide de la connexion avec Apple\)](#).

Pour intégrer la connexion avec Apple dans une plateforme native non iOS, consultez [Sign in with Apple JS \(Connexion avec Apple JS\)](#).

Configurer le fournisseur externe dans la console Identités fédérées Amazon Cognito

Utilisez la procédure suivante pour configurer votre fournisseur externe.

Pour ajouter un fournisseur d'identité (IdP) Se connecter avec Apple

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Se connecter avec Apple.
5. Saisissez l'ID des services du projet OAuth que vous avez créé avec [Apple Developer](#). Pour plus d'informations, consultez [Authentification des utilisateurs avec Se connecter avec Apple](#) (langue française non garantie) dans la documentation de Se connecter avec Apple.
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.

- ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
8. Sélectionnez Enregistrer les modifications.

Exemples CLI d'utilisation de Sign in with Apple en tant que fournisseur dans les Identités fédérées Amazon Cognito

Cet exemple crée un groupe d'identités nommé `MyIdentityPool` avec la fonctionnalité Se connecter avec Apple comme fournisseur d'identité.

```
aws cognito-identity create-identity-pool --identity-pool-name MyIdentityPool --supported-login-providers appleid.apple.com="sameple.apple.clientid"
```

Pour plus d'informations, consultez [Create identity pool \(Créer un groupe d'identités\)](#)

Générer un ID d'identité Amazon Cognito

Cet exemple génère (ou extrait) un ID Amazon Cognito. Il s'agit d'une API publique, vous n'avez donc pas besoin d'informations d'identification pour appeler cette API.

```
aws cognito-identity get-id --identity-pool-id SampleIdentityPoolId --logins appleid.apple.com="SignInWithAppleIdToken"
```

Pour plus d'informations, consultez [get-id.\(obtenir-id\)](#)

Obtenir des informations d'identification pour un ID d'identité Amazon Cognito

Cet exemple renvoie les informations d'identification pour l'ID d'identité fourni et la connexion avec un identifiant Apple. Il s'agit d'une API publique, vous n'avez donc pas besoin d'informations d'identification pour appeler cette API.

```
aws cognito-identity get-credentials-for-identity --identity-id
SampleIdentityId --logins appleid.apple.com="SignInWithAppleIdToken"
```

Pour plus d'informations, voir [get-credentials-for-identity](#)

Utiliser Se connecter avec Apple : Android

Apple ne fournit pas de SDK compatible avec la connexion avec Apple pour Android. Vous pouvez utiliser le flux Web dans une vue Web à la place.

- Pour configurer la connexion avec Apple dans votre application, veuillez consulter [Configuring Your Webpage for Sign In with Apple \(Configurer votre page Web avec la connexion avec Apple\)](#) dans la documentation Apple.
- Pour ajouter un bouton Sign in with Apple (Se connecter avec Apple) à votre interface utilisateur Android, veuillez consulter [Displaying and Configuring Sign In with Apple Buttons](#) dans la documentation Apple.
- Pour authentifier les utilisateurs de manière sécurisée avec la fonctionnalité Se connecter avec Apple, suivez les instructions d'[authentification des utilisateurs avec la fonctionnalité Se connecter avec Apple](#) dans la documentation Apple.

Sign in with Apple utilise un objet session pour suivre son état. Amazon Cognito utilise le jeton d'identification de cet objet de session pour authentifier l'utilisateur, générer l'identifiant unique et, si nécessaire, accorder à l'utilisateur l'accès à d'autres ressources. AWS

```
@Override
public void onSuccess(Bundle response) {
    String token = response.getString("id_token");
    Map<String, String> logins = new HashMap<String, String>();
    logins.put("appleid.apple.com", token);
    credentialsProvider.setLogins(logins);
}
```

Utiliser Se connecter avec Apple : iOS - Objective-C

Apple a fourni la prise en charge de SDK pour Se connecter avec Apple dans les applications iOS natives. Pour mettre en place l'authentification utilisateur par le biais de la connexion avec Apple dans les appareils iOS natifs, veuillez consulter [Implementing User Authentication with Sign in with Apple \(Mettre en place l'authentification de l'utilisateur par le biais de la connexion avec Apple\)](#) dans la documentation Apple.

Amazon Cognito utilise le jeton d'identification pour authentifier l'utilisateur, générer l'identifiant unique et, si nécessaire, accorder à l'utilisateur l'accès à d'autres ressources. AWS

```
(void)finishedWithAuth: (ASAuthorizationAppleIDCredential *)auth error: (NSError *)
error {
    NSString *idToken = [ASAuthorizationAppleIDCredential
objectForKey:@"identityToken"];
    credentialsProvider.logins = @{ "appleid.apple.com": idToken };
}
```

Utiliser Se connecter avec Apple : iOS - Swift

Apple a fourni la prise en charge de SDK pour Se connecter avec Apple dans les applications iOS natives. Pour mettre en place l'authentification utilisateur par le biais de la connexion avec Apple dans les appareils iOS natifs, veuillez consulter [Implementing User Authentication with Sign in with Apple \(Mettre en place l'authentification de l'utilisateur par le biais de la connexion avec Apple\)](#) dans la documentation Apple.

Amazon Cognito utilise le jeton d'identification pour authentifier l'utilisateur, générer l'identifiant unique et, si nécessaire, accorder à l'utilisateur l'accès à d'autres ressources. AWS

Pour plus d'informations sur la configuration de la fonctionnalité Se connecter avec Apple dans iOS, consultez les instructions de [configuration de la fonctionnalité Se connecter avec Apple](#)

```
func finishedWithAuth(auth: ASAuthorizationAppleIDCredential!, error: NSError!) {
    if error != nil {
        print(error.localizedDescription)
    }
    else {
        let idToken = auth.identityToken,
        credentialsProvider.logins = ["appleid.apple.com": idToken!]
    }
}
```


Utilisez Se connecter avec Apple : JavaScript

Apple ne fournit pas de SDK compatible avec Sign in with Apple for JavaScript. Vous pouvez utiliser le flux Web dans une vue Web à la place.

- Pour configurer la connexion avec Apple dans votre application, veuillez consulter [Configuring Your Webpage for Sign In with Apple \(Configurer votre page Web avec la connexion avec Apple\)](#) dans la documentation Apple.
- Pour ajouter un bouton Connexion avec Apple à votre interface JavaScript utilisateur, suivez la section [Affichage et configuration de la connexion à l'aide des boutons Apple](#) dans la documentation Apple.
- Pour authentifier en toute sécurité les utilisateurs via la fonctionnalité Se connecter avec Apple, suivez les instructions de [configuration de votre page web pour la fonctionnalité Se connecter avec Apple](#) dans la documentation Apple.

Sign in with Apple utilise un objet session pour suivre son état. Amazon Cognito utilise le jeton d'identification de cet objet de session pour authentifier l'utilisateur, générer l'identifiant unique et, si nécessaire, accorder à l'utilisateur l'accès à d'autres ressources. AWS

```
function signinCallback(authResult) {
    // Add the apple's id token to the Amazon Cognito credentials login map.
    AWS.config.credentials = new AWS.CognitoIdentityCredentials({
        IdentityPoolId: 'IDENTITY_POOL_ID',
        Logins: {
            'appleid.apple.com': authResult['id_token']
        }
    });

    // Obtain AWS credentials
    AWS.config.credentials.get(function(){
        // Access AWS resources here.
    });
}
```

Utiliser Se connecter avec Apple : Xamarin

Nous n'avons pas de SDK compatible avec la connexion avec Apple pour Xamarin. Vous pouvez utiliser le flux Web dans une vue Web à la place.

- Pour configurer la connexion avec Apple dans votre application, veuillez consulter [Configuring Your Webpage for Sign In with Apple \(Configurer votre page Web avec la connexion avec Apple\)](#) dans la documentation Apple.
- Pour ajouter un bouton Sign in with Apple (Se connecter avec Apple) à votre interface utilisateur Xamarin, veuillez consulter [Displaying and Configuring Sign In with Apple Buttons](#) dans la documentation Apple.
- Pour authentifier en toute sécurité les utilisateurs via la fonctionnalité Se connecter avec Apple, suivez les instructions de [configuration de votre page web pour la fonctionnalité Se connecter avec Apple](#) dans la documentation Apple.

Sign in with Apple utilise un objet session pour suivre son état. Amazon Cognito utilise le jeton d'identification de cet objet de session pour authentifier l'utilisateur, générer l'identifiant unique et, si nécessaire, accorder à l'utilisateur l'accès à d'autres ressources. AWS

Une fois que vous avez ce jeton, vous pouvez le définir dans `CognitoAWSCredentials` :

```
credentials.AddLogin("appleid.apple.com", token);
```

Configuration d'un fournisseur OIDC en tant qu'IdP de pool d'identités

[OpenID Connect](#) est une norme ouverte d'authentification que de nombreux fournisseurs de connexion prennent en charge. Amazon Cognito vous aide à lier des identités aux fournisseurs OpenID Connect que vous configurez via [AWS Identity and Access Management](#).

Ajout d'un fournisseur OpenID Connect

Pour obtenir des informations sur la création d'un fournisseur OpenID Connect, consultez [Création de fournisseurs d'identité OpenID Connect \(OIDC\)](#) dans le Guide de l'utilisateur AWS Identity and Access Management .

Association d'un fournisseur à Amazon Cognito

Pour ajouter un fournisseur d'identité (IdP) OIDC

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.

4. Choisissez OpenID Connect (OIDC).
5. Choisissez un fournisseur d'identité OIDC parmi l'IAM IdPs de votre. Compte AWS Pour ajouter un nouveau fournisseur SAML, choisissez Créer un nouveau fournisseur afin d'accéder à la console IAM.
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.
7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
8. Sélectionnez Enregistrer les modifications.

Vous pouvez associer plusieurs fournisseurs OpenID Connect à un seul groupe d'identités.

Utilisation d'OpenID Connect

Reportez-vous à la documentation de votre fournisseur pour découvrir comment vous connecter et recevoir un jeton d'identification.

Une fois que vous avez un jeton, ajoutez-le au mappage des connexions. Utilisez l'URI de votre fournisseur comme clé.

Validation d'un jeton OpenID Connect

Lors de l'intégration initiale avec Amazon Cognito, vous pouvez recevoir une exception `InvalidToken`. Il est important de comprendre comment Amazon Cognito valide les jetons OpenID Connect (OIDC).

Note

Comme indiqué ici (<https://tools.ietf.org/html/rfc7523>), Amazon Cognito fournit une période de grâce de 5 minutes pour prendre en compte un éventuel décalage d'horloge entre systèmes.

1. Le paramètre `iss` doit correspondre à la clé utilisée dans le mappage des connexions (par exemple, `login.provider.com`).
2. La signature doit être valide. Elle doit être vérifiable via une clé publique RSA.
3. L'empreinte de la clé publique du certificat correspond à l'empreinte que vous avez définie dans IAM quand vous avez créé le fournisseur OIDC.
4. Si le paramètre `azp` est présent, vérifiez cette valeur par rapport aux ID client répertoriés dans le fournisseur OIDC.
5. Si le paramètre `azp` n'est pas présent, vérifiez le paramètre `aud` par rapport aux ID client répertoriés dans le fournisseur OIDC.

Le site web jwt.io est une ressource précieuse que vous pouvez utiliser pour décoder les jetons et vérifier ces valeurs.

Android

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("login.provider.com", token);
credentialsProvider.setLogins(logins);
```

iOS : Objective-C

```
credentialsProvider.logins = @{ "login.provider.com": token }
```

iOS : Swift

Pour fournir le jeton d'identification OIDC à Amazon Cognito, implémentez le protocole `AWSIIdentityProviderManager`.

Quand vous implémentez la méthode `logins`, renvoyez un dictionnaire contenant le nom du fournisseur OIDC que vous avez configuré. Ce dictionnaire fait office de clé et le jeton d'identification actuel de l'utilisateur authentifié fait office de valeur, comme dans l'exemple de code suivant.

```
class OIDCProvider: NSObject, AWSIdentityProviderManager {
    func logins() -> AWSTask<NSDictionary> {
        let completion = AWSTaskCompletionSource<NSString>()
        getToken(tokenCompletion: completion)
        return completion.task.continueOnSuccessWith { (task) -> AWSTask<NSDictionary>?
in
            //login.provider.name is the name of the OIDC provider as setup in the
Amazon Cognito console
            return AWSTask(result:["login.provider.name":task.result!])
        } as! AWSTask<NSDictionary>

    }

    func getToken(tokenCompletion: AWSTaskCompletionSource<NSString>) -> Void {
        //get a valid oidc token from your server, or if you have one that hasn't
expired cached, return it

        //TODO code to get token from your server
        //...

        //if error getting token, set error appropriately
        tokenCompletion.set(error:NSError(domain: "OIDC Login", code: -1 , userInfo:
["Unable to get OIDC token" : "Details about your error"]))
        //else
        tokenCompletion.set(result:"result from server id token")
    }
}
```

Lorsque vous instanciez le `AWSCognitoCredentialsProvider`, transmettez la classe qui implémente `AWSSignInProviderManager` comme valeur de `identityProviderManager` dans le constructeur. Pour plus d'informations, rendez-vous sur la page de [AWSCognitoCredentialsProvider](#) référence et choisissez `initWithRegionType : identityPoolId : identityProviderManager`.

JavaScript

```
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: 'IDENTITY_POOL_ID',
  Logins: {
    'login.provider.com': token
  }
});
```

Unity

```
credentials.AddLogin("login.provider.com", token);
```

Xamarin

```
credentials.AddLogin("login.provider.com", token);
```

Configuration d'un fournisseur SAML en tant qu'IdP de pool d'identités

Amazon Cognito prend en charge l'authentification auprès des fournisseurs d'identité (IdPs) via le langage SAML 2.0 (Security Assertion Markup Language 2.0). Vous pouvez utiliser un fournisseur d'identité qui prend en charge le langage SAML avec Amazon Cognito pour faciliter l'intégration de vos utilisateurs. Votre fournisseur d'identité prenant en charge SAML spécifie les rôles IAM que vos utilisateurs peuvent endosser. De cette façon, différents utilisateurs peuvent recevoir différents ensembles d'autorisations.

Configuration de votre groupe d'identités pour un fournisseur d'identité SAML

Les étapes suivantes expliquent comment configurer votre groupe d'identités pour utiliser un fournisseur d'identité basé sur SAML.

Note

Avant de configurer votre groupe d'identités pour prendre en charge un fournisseur SAML, configurez le fournisseur d'identité SAML dans la [console IAM](#). Pour plus d'informations, consultez [Intégration de prestataires de solution SAML tiers avec AWS](#) dans le Guide de l'utilisateur IAM.

Pour ajouter un fournisseur d'identité (IdP) SAML

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez SAML.
5. Choisissez un fournisseur d'identité SAML IdPs dans l'IAM de votre. Compte AWS Pour ajouter un nouveau fournisseur SAML, choisissez Créer un nouveau fournisseur afin d'accéder à la console IAM.
6. Pour définir le rôle demandé par Amazon Cognito lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Paramètres de rôle.
 - Vous pouvez attribuer aux utilisateurs de ce fournisseur d'identité le rôle par défaut que vous avez configuré lorsque vous avez configuré votre rôle authentifié, ou vous pouvez sélectionner Choisir un rôle avec des règles.
 - i. Si vous avez choisi Choisir un rôle avec des règles, saisissez la demande source issue de l'authentification de votre utilisateur, l'opérateur avec lequel vous souhaitez comparer ce champ standard, la valeur qui entraînera une correspondance avec ce choix de rôle et le rôle que vous souhaitez attribuer si l'attribution de rôle correspond. Sélectionnez Ajouter un autre pour créer une règle supplémentaire basée sur une condition différente.
 - ii. Choisissez une résolution de rôle. Lorsque les champs standard de votre utilisateur ne correspondent pas à vos règles, vous pouvez refuser les informations d'identification ou émettre des informations d'identification pour votre rôle authentifié.

7. Pour modifier les balises de principal qu'Amazon Cognito attribue lorsqu'il délivre des informations d'identification aux utilisateurs qui se sont authentifiés auprès de ce fournisseur, configurez Attributs de contrôle d'accès.
 - a. Pour n'appliquer aucune balise de principal, choisissez Inactif.
 - b. Pour appliquer les balises de principal en fonction des champs standard sub et aud, choisissez Utiliser les mappages par défaut.
 - c. Pour créer votre propre schéma personnalisé d'attributs pour les balises de principal, choisissez Utiliser des mappages personnalisés. Saisissez ensuite une clé de balise que vous souhaitez obtenir à partir de chaque demande que vous souhaitez représenter dans une balise.
8. Sélectionnez Enregistrer les modifications.

Configuration de votre fournisseur d'identité SAML

Après avoir créé le fournisseur SAML, configurez votre fournisseur d'identité SAML pour ajouter une relation d'approbation des parties utilisatrices entre votre fournisseur d'identité et AWS. Dans de nombreux IdPs cas, vous pouvez spécifier une URL que l'IdP peut utiliser pour lire les informations et les certificats des parties fiables à partir d'un document XML. Pour AWS cela, vous pouvez utiliser <https://signin.aws.amazon.com/static/saml-metadata.xml>. L'étape suivante consiste à configurer la réponse d'assertion SAML de votre IdP pour renseigner les demandes nécessaires. AWS Pour plus d'informations sur la configuration des demandes, consultez la page [Configuration des assertions SAML pour la réponse d'authentification](#).

Lorsque votre fournisseur d'identités SAML inclut plusieurs certificats de signature dans les métadonnées SAML, lors de la connexion, votre groupe d'utilisateurs détermine que l'assertion SAML est valide si elle correspond à un certificat figurant dans les métadonnées SAML.

Personnalisation d'un rôle d'utilisateur avec SAML

Quand vous utilisez SAML avec Amazon Cognito Identity, vous pouvez personnaliser le rôle pour l'utilisateur final. Amazon Cognito prend uniquement en charge le [flux amélioré](#) avec le fournisseur d'identité basé sur SAML. Vous n'avez pas besoin de spécifier de rôle authentifié ou non authentifié pour que le groupe d'identités utilise un fournisseur d'identité basé sur SAML. L'attribut de la demande `https://aws.amazon.com/SAML/Attributes/Role` spécifie une ou plusieurs paires d'ARN (Amazon Resource Name) de fournisseur et de rôle, délimité par une virgule. Ce sont les rôles que l'utilisateur peut endosser. Vous pouvez configurer le fournisseur d'identité

SAML pour renseigner les attributs de rôle en fonction des informations d'attribut utilisateur qu'il met à disposition. Si vous recevez plusieurs rôles dans l'assertion SAML, renseignez le paramètre `customRoleArn` facultatif quand vous appelez `getCredentialsForIdentity`. L'utilisateur endosse ce `customRoleArn` si le rôle correspond à un rôle dans la demande, dans l'assertion SAML.

Authentification des utilisateurs avec un fournisseur d'identité SAML

Pour fédérer avec l'IdP basé sur SAML, déterminez l'URL à laquelle l'utilisateur initie la connexion. AWS la fédération utilise une connexion initiée par l'IDP. Dans AD FS 2.0, l'URL prend la forme `https://<fqdn>/adfs/ls/IdpInitiatedSignOn.aspx?loginToRp=urn:amazon:webservices`.

Pour ajouter la prise en charge de votre fournisseur d'identité SAML dans Amazon Cognito, commencez par authentifier les utilisateurs avec votre fournisseur d'identité SAML à partir de votre application iOS ou Android. Le code que vous utilisez pour intégrer et authentifier avec le fournisseur d'identité SAML est spécifique aux fournisseurs SAML. Après avoir authentifié votre utilisateur, vous pouvez utiliser les API Amazon Cognito pour fournir l'assertion SAML obtenue à Amazon Cognito Identity.

Vous ne pouvez pas répéter ni réutiliser une assertion SAML dans la carte Logins de votre demande d'API de groupe d'identités. Une assertion SAML réutilisée possède un ID d'assertion qui duplique l'ID d'une demande d'API précédente. Les opérations d'API qui peuvent accepter une assertion SAML sur la Logins carte incluent [GetId](#), [GetCredentialsForIdentity](#), [GetOpenIdToken](#), et [GetOpenIdTokenForDeveloperIdentity](#). Vous pouvez rejouer un ID d'assertion SAML une fois par demande d'API dans un flux d'authentification des groupes d'identités. Par exemple, vous pouvez fournir la même assertion SAML dans une demande `GetId` et dans une demande `GetCredentialsForIdentity` ultérieure, mais pas dans une seconde demande `GetId`.

Android

Si vous utilisez le kit SDK Android, vous pouvez renseigner le mappage des connexions avec l'assertion SAML comme suit.

```
Map logins = new HashMap();
logins.put("arn:aws:iam::aws account id:saml-provider/name", "base64 encoded assertion
response");
// Now this should be set to CognitoCachingCredentialsProvider object.
CognitoCachingCredentialsProvider credentialsProvider = new
CognitoCachingCredentialsProvider(context, identity pool id, region);
```

```
credentialsProvider.setLogins(logins);
// If SAML assertion contains multiple roles, resolve the role by setting the custom
  role
credentialsProvider.setCustomRoleArn("arn:aws:iam::aws account id:role/
customRoleName");
// This should trigger a call to the Amazon Cognito service to get the credentials.
credentialsProvider.getCredentials();
```

iOS

Si vous utilisez le SDK iOS, vous pouvez fournir l'assertion SAML dans `AWSCognitoIdentityProviderManager` comme suit.

```
- (AWSTask<NSDictionary<NSString*,NSString*> *> *) logins {
    //this is hardcoded for simplicity, normally you would asynchronously go to your
    SAML provider
    //get the assertion and return the logins map using a AWSTaskCompletionSource
    return [AWSTask taskWithResult:@{@"arn:aws:iam::aws account id:saml-provider/
name":@"base64 encoded assertion response"}];
}

// If SAML assertion contains multiple roles, resolve the role by setting the custom
  role.
// Implementing this is optional if there is only one role.
- (NSString *)customRoleArn {
    return @"arn:aws:iam::accountId:role/customRoleName";
}
```

Identités authentifiées par le développeur (réserves d'identités)

Amazon Cognito prend en charge les identités authentifiées par le développeur en plus de la fédération d'identité Web via [Configuration de Facebook en tant qu'IdP de pool d'identités](#), [Configuration de Google en tant qu'IdP de pool d'identités](#), [Configuration de Login with Amazon en tant qu'IdP de pool d'identités](#) et [Configuration de la connexion avec Apple en tant qu'IdP du pool d'identités](#). Grâce aux identités authentifiées par les développeurs, vous pouvez enregistrer et authentifier les utilisateurs par le biais de votre propre processus d'authentification existant, tout en utilisant Amazon Cognito pour synchroniser les données utilisateur et accéder aux ressources. AWS Leur utilisation implique l'interaction entre l'appareil de l'utilisateur final, votre système backend d'authentification et Amazon Cognito. Pour plus de détails, consultez la section [Comprendre](#)

[l'authentification Amazon Cognito, partie 2 : Identités authentifiées par les développeurs](#) sur le blog.

AWS

Présentation du flux d'authentification

Le fonctionnement de l'[GetOpenIdTokenForDeveloperIdentity](#) API peut initier l'authentification du développeur pour l'authentification améliorée et l'authentification de base. Cette API authentifie une demande à l'aide d'informations d'identification administratives. La Logins carte est un nom de fournisseur de développeur de pool d'identités `login.mydevprovider` associé à un identifiant personnalisé.

Exemple :

```
"Logins": {
  "login.mydevprovider": "my developer identifier"
}
```

Authentification améliorée

Appelez l'opération [GetCredentialsForIdentity](#) API avec une Logins carte avec le nom `cognito-identity.amazonaws.com` et la valeur du jeton de `GetOpenIdTokenForDeveloperIdentity`.

Exemple :

```
"Logins": {
  "cognito-identity.amazonaws.com": "eyJra12345EXAMPLE"
}
```

`GetCredentialsForIdentity` avec des identités authentifiées par le développeur renvoie des informations d'identification temporaires pour le rôle authentifié par défaut du pool d'identités.

Authentification de base

Appelez l'opération [AssumeRoleWithWebIdentity](#) API et demandez le nom `RoleArn` de n'importe quel rôle IAM pour lequel une [relation de confiance appropriée est définie](#). Définissez la valeur de `WebIdentityToken` le jeton obtenu auprès de `GetOpenIdTokenForDeveloperIdentity`.

Pour plus d'informations sur le flux d'authentification des identités authentifiées par le développeur et sur la façon dont il diffère des identités des fournisseurs externes, voir [Flux d'authentification de groupes d'identités \(identités fédérées\)](#)

Définir un nom de fournisseur de développement et l'associer à un groupe d'identités

Pour utiliser les identités authentifiées par le développeur, vous avez besoin d'une réserve d'identités associée à votre fournisseur de développement. Pour ce faire, procédez comme suit :

Pour ajouter un fournisseur de développement personnalisé

1. Choisissez Groupes d'identités dans la [console Amazon Cognito](#). Sélectionnez une réserve d'identités.
2. Choisissez l'onglet Accès utilisateur.
3. Sélectionnez Ajouter un fournisseur d'identité.
4. Choisissez Fournisseur du développeur personnalisé.
5. Saisissez le nom du fournisseur de développement. Vous ne pouvez pas modifier ni supprimer votre fournisseur de développement après l'avoir ajouté.
6. Sélectionnez Enregistrer les modifications.

Remarque : une fois que le nom du fournisseur a été défini, il ne peut pas être modifié.

Pour des instructions supplémentaires sur l'utilisation de la console Amazon Cognito, consultez [Utilisation de la console Amazon Cognito](#).

Implémentation d'un fournisseur d'identité

Android

Pour utiliser des identités authentifiées par le développeur, implémentez votre propre classe de fournisseur d'identité qui étend `AWSAbstractCognitoIdentityProvider`. Votre classe de fournisseur d'identité doit retourner un objet de réponse contenant le jeton comme attribut.

Vous trouverez ci-après un exemple de base d'un fournisseur d'identité.

```
public class DeveloperAuthenticationProvider extends
    AWSAbstractCognitoDeveloperIdentityProvider {

    private static final String developerProvider = "<Developer_provider_name>";

    public DeveloperAuthenticationProvider(String accountId, String identityPoolId,
        Regions region) {
```

```
    super(accountId, identityPoolId, region);
    // Initialize any other objects needed here.
}

// Return the developer provider name which you choose while setting up the
// identity pool in the &COG; Console

@Override
public String getProviderName() {
    return developerProvider;
}

// Use the refresh method to communicate with your backend to get an
// identityId and token.

@Override
public String refresh() {

    // Override the existing token
    setToken(null);

    // Get the identityId and token by making a call to your backend
    // (Call to your backend)

    // Call the update method with updated identityId and token to make sure
    // these are ready to be used from Credentials Provider.

    update(identityId, token);
    return token;
}

// If the app has a valid identityId return it, otherwise get a valid
// identityId from your backend.

@Override
public String getIdentityId() {

    // Load the identityId from the cache
    identityId = cachedIdentityId;

    if (identityId == null) {
        // Call to your backend
    } else {
```

```
        return identityId;
    }
}
}
```

Pour utiliser ce fournisseur d'identité, vous devez le transmettre dans `CognitoCachingCredentialsProvider`. Voici un exemple :

```
DeveloperAuthenticationProvider developerProvider = new
    DeveloperAuthenticationProvider( null, "IDENTITYPOOLID", context, Regions.USEAST1);
CognitoCachingCredentialsProvider credentialsProvider = new
    CognitoCachingCredentialsProvider( context, developerProvider, Regions.USEAST1);
```

iOS – objective-C

Pour utiliser des identités authentifiées par le développeur, implémentez votre propre classe de fournisseur d'identité qui étend [AWSCognitoCredentialsProviderHelper](#). Votre classe de fournisseur d'identité doit retourner un objet de réponse contenant le jeton comme attribut.

```
@implementation DeveloperAuthenticatedIdentityProvider
/*
 * Use the token method to communicate with your backend to get an
 * identityId and token.
 */
- (AWSTask <NSString*> *) token {
    //Write code to call your backend:
    //Pass username/password to backend or some sort of token to authenticate user
    //If successful, from backend call getOpenIdTokenForDeveloperIdentity with logins
    map
    //containing "your.provider.name":"enduser.username"
    //Return the identity id and token to client
    //You can use AWSTaskCompletionSource to do this asynchronously

    // Set the identity id and return the token
    self.identityId = response.identityId;
    return [AWSTask taskWithResult:response.token];
}

@end
```

Pour utiliser ce fournisseur d'identité, transmettez-le dans `AWSCognitoCredentialsProvider` comme illustré dans l'exemple suivant :

```
DeveloperAuthenticatedIdentityProvider * devAuth =
[[DeveloperAuthenticatedIdentityProvider alloc]
 initWithRegionType:AWSRegionYOUR_IDENTITY_POOL_REGION
                identityPoolId:@"YOUR_IDENTITY_POOL_ID"
                useEnhancedFlow:YES
                identityProviderManager:nil];
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider
 alloc]

 initWithRegionType:AWSRegionYOUR_IDENTITY_POOL_REGION
                identityProvider:devAuth];
```

Si vous souhaitez prendre en charge les identités non authentifiées et les identités authentifiées par le développeur, remplacez la méthode `logins` dans votre implémentation de `AWSCognitoCredentialsProviderHelper`.

```
- (AWSTask<NSDictionary<NSString *, NSString *> * > *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else{
        return [super logins];
    }
}
```

Si vous souhaitez prendre en charge les identités authentifiées par le développeur et les fournisseurs sociaux, vous devez gérer qui est le fournisseur actuel dans votre implémentation `logins` de `AWSCognitoCredentialsProviderHelper`.

```
- (AWSTask<NSDictionary<NSString *, NSString *> * > *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else if (/*logic to determine if user is Facebook*/){
        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook :
[FBSDKAccessToken currentAccessToken] }];
    }else {
        return [super logins];
    }
}
```

iOS – swift

Pour utiliser des identités authentifiées par le développeur, implémentez votre propre classe de fournisseur d'identité qui étend [AWSCognitoCredentialsProviderHelper](#). Votre classe de fournisseur d'identité doit retourner un objet de réponse contenant le jeton comme attribut.

```
import AWSCore
/*
 * Use the token method to communicate with your backend to get an
 * identityId and token.
 */
class DeveloperAuthenticatedIdentityProvider : AWSCognitoCredentialsProviderHelper {
    override func token() -> AWSTask<NSString> {
        //Write code to call your backend:
        //pass username/password to backend or some sort of token to authenticate user, if
        successful,
        //from backend call getOpenIdTokenForDeveloperIdentity with logins map containing
        "your.provider.name":"enduser.username"
        //return the identity id and token to client
        //You can use AWSTaskCompletionSource to do this asynchronously

        // Set the identity id and return the token
        self.identityId = resultFromAbove.identityId
        return AWSTask(result: resultFromAbove.token)
    }
}
```

Pour utiliser ce fournisseur d'identité, transmettez-le dans `AWSCognitoCredentialsProvider` comme illustré dans l'exemple suivant :

```
let devAuth =
    DeveloperAuthenticatedIdentityProvider(regionType: .YOUR_IDENTITY_POOL_REGION,
        identityPoolId: "YOUR_IDENTITY_POOL_ID", useEnhancedFlow: true,
        identityProviderManager:nil)
let credentialsProvider =
    AWSCognitoCredentialsProvider(regionType: .YOUR_IDENTITY_POOL_REGION,
        identityProvider:devAuth)
let configuration = AWSServiceConfiguration(region: .YOUR_IDENTITY_POOL_REGION,
        credentialsProvider:credentialsProvider)
AWSServiceManager.default().defaultServiceConfiguration = configuration
```


Si vous souhaitez prendre en charge les identités non authentifiées et les identités authentifiées par le développeur, remplacez la méthode `logins` dans votre implémentation de `AWSCognitoCredentialsProviderHelper`.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else {
        return super.logins()
    }
}
```

Si vous souhaitez prendre en charge les identités authentifiées par le développeur et les fournisseurs sociaux, vous devez gérer qui est le fournisseur actuel dans votre implémentation `logins` de `AWSCognitoCredentialsProviderHelper`.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else if (/*logic to determine if user is Facebook*/){
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error:NSError(domain: "Facebook Login", code: -1 , userInfo:
["Facebook" : "No current Facebook access token"]))
    }else {
        return super.logins()
    }
}
```

JavaScript

Une fois que vous obtenez un ID d'identité et un jeton de session à partir de votre système backend, transmettez-les au fournisseur `AWS.CognitoIdentityCredentials`. Voici un exemple :

```
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
    IdentityPoolId: 'IDENTITY_POOL_ID',
    IdentityId: 'IDENTITY_ID_RETURNED_FROM_YOUR_PROVIDER',
    Logins: {
        'cognito-identity.amazonaws.com': 'TOKEN_RETURNED_FROM_YOUR_PROVIDER'
    }
})
```

```
});
```

Unity

Afin d'utiliser les identités authentifiées par le développeur, vous devez étendre `CognitoAWSCredentials` et remplacer la méthode `RefreshIdentity` pour récupérer l'ID d'identité utilisateur et le jeton à partir de votre système backend, puis les renvoyer. Voici un exemple simple de fournisseur d'identité qui contacte un système backend hypothétique à l'adresse « `example.com` » :

```
using UnityEngine;
using System.Collections;
using Amazon.CognitoIdentity;
using System.Collections.Generic;
using ThirdParty.Json.LitJson;
using System;
using System.Threading;

public class DeveloperAuthenticatedCredentials : CognitoAWSCredentials
{
    const string PROVIDER_NAME = "example.com";
    const string IDENTITY_POOL = "IDENTITY_POOL_ID";
    static readonly RegionEndpoint REGION = RegionEndpoint.USEast1;

    private string login = null;

    public DeveloperAuthenticatedCredentials(string loginAlias)
        : base(IDENTITY_POOL, REGION)
    {
        login = loginAlias;
    }

    protected override IdentityState RefreshIdentity()
    {
        IdentityState state = null;
        ManualResetEvent waitLock = new ManualResetEvent(false);
        MainThreadDispatcher.ExecuteCoroutineOnMainThread(ContactProvider((s) =>
        {
            state = s;
            waitLock.Set();
        })));
        waitLock.WaitOne();
        return state;
    }
}
```

```
}

IEnumerator ContactProvider(Action<IdentityState> callback)
{
    WWW www = new WWW("http://example.com/?username="+login);
    yield return www;
    string response = www.text;

    JsonData json = JsonMapper.ToObject(response);

    //The backend has to send us back an Identity and a OpenID token
    string identityId = json["IdentityId"].ToString();
    string token = json["Token"].ToString();

    IdentityState state = new IdentityState(identityId, PROVIDER_NAME, token,
false);
    callback(state);
}
}
```

Le code ci-dessus utilise un objet de répartiteur de thread pour appeler une co-routine. Si vous n'avez pas la possibilité de le faire dans votre projet, vous pouvez utiliser le script suivant dans vos scènes :

```
using System;
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class MainThreadDispatcher : MonoBehaviour
{
    static Queue<IEnumerator> _coroutineQueue = new Queue<IEnumerator>();
    static object _lock = new object();

    public void Update()
    {
        while (_coroutineQueue.Count > 0)
        {
            StartCoroutine(_coroutineQueue.Dequeue());
        }
    }

    public static void ExecuteCoroutineOnMainThread(IEnumerator coroutine)
    {
```

```
        lock (_lock) {
            _coroutineQueue.Enqueue(coroutine);
        }
    }
}
```

Xamarin

Afin d'utiliser les identités authentifiées par le développeur, vous devez étendre `CognitoAWSCredentials` et remplacer la méthode `RefreshIdentity` pour récupérer l'ID d'identité utilisateur et le jeton à partir de votre système backend, puis les renvoyer. Voici un exemple élémentaire de fournisseur d'identité qui contacte un système backend hypothétique à l'adresse « `example.com` » :

```
public class DeveloperAuthenticatedCredentials : CognitoAWSCredentials
{
    const string PROVIDER_NAME = "example.com";
    const string IDENTITY_POOL = "IDENTITY_POOL_ID";
    static readonly RegionEndpoint REGION = RegionEndpoint.USEast1;
    private string login = null;

    public DeveloperAuthenticatedCredentials(string loginAlias)
        : base(IDENTITY_POOL, REGION)
    {
        login = loginAlias;
    }

    protected override async Task<IdentityState> RefreshIdentityAsync()
    {
        IdentityState state = null;
        //get your identity and set the state
        return state;
    }
}
```

Mise à jour de la carte de connexions (Android et iOS uniquement)

Android

Une fois que l'utilisateur est authentifié avec votre système d'authentification, mettez à jour le mappage des connexions avec le nom du fournisseur de développement et un ID utilisateur de

développeur. Il s'agit d'une chaîne alphanumérique qui identifie de manière unique un utilisateur dans votre système d'authentification. Veillez à appeler la méthode `refresh` après la mise à jour de la carte de connexions, car il se peut que `identityId` ait changé :

```
HashMap<String, String> loginsMap = new HashMap<String, String>();
loginsMap.put(developerAuthenticationProvider.getProviderName(),
    developerUserIdentifier);

credentialsProvider.setLogins(loginsMap);
credentialsProvider.refresh();
```

iOS – objective-C

Le kit SDK iOS appelle uniquement votre méthode `logins` pour obtenir la dernière carte de connexions s'il n'existe pas d'informations d'identification ou si celles-ci ont expiré. Si vous voulez forcer le kit SDK à obtenir de nouvelles informations d'identification (par exemple, votre utilisateur final est passé de non authentifié à authentifié et vous souhaitez des informations d'identification pour l'utilisateur authentifié), appelez `clearCredentials` sur votre `credentialsProvider`.

```
[credentialsProvider clearCredentials];
```

iOS – swift

Le kit SDK iOS appelle uniquement votre méthode `logins` pour obtenir la dernière carte de connexions s'il n'existe pas d'informations d'identification ou si celles-ci ont expiré. Si vous voulez forcer le kit SDK à obtenir les nouvelles informations d'identification (par exemple, votre utilisateur final est passé de non authentifié à authentifié et vous souhaitez des informations d'identification pour l'utilisateur authentifié), appelez `clearCredentials` sur votre `credentialsProvider`.

```
credentialsProvider.clearCredentials()
```

Obtention d'un jeton (côté serveur)

Vous obtenez un jeton en appelant [GetOpenIdTokenForDeveloperIdentity](#). Cette API doit être invoquée depuis votre backend à l'aide des informations d'identification AWS du développeur. Elle ne doit pas être appelée depuis le kit SDK client. L'API reçoit l'ID de réserve d'identités Cognito, un mappage de connexions contenant votre nom de fournisseur d'identité comme clé et l'identificateur comme valeur, ainsi qu'éventuellement un ID d'identité Cognito (par exemple, vous transformez un utilisateur non authentifié en utilisateur authentifié). L'identificateur peut être le nom de l'utilisateur,

une adresse e-mail ou une valeur numérique. L'API répond à votre appel avec un ID Cognito unique et un jeton OpenID Connect pour l'utilisateur.

Voici quelques éléments à garder à l'esprit sur le jeton renvoyé par

`:GetOpenIdTokenForDeveloperIdentity`

- Vous pouvez spécifier une durée d'expiration personnalisée pour le jeton, afin de pouvoir le mettre en cache. Si vous ne fournissez pas de durée d'expiration personnalisée, le jeton est valide pendant 15 minutes.
- La durée maximale que vous pouvez définir pour un jeton est de 24 heures.
- Gardez à l'esprit les répercussions que l'augmentation de la durée de validité du jeton peut avoir sur la sécurité. Si un attaquant obtient ce jeton, il peut l'échanger contre des AWS informations d'identification pour l'utilisateur final pendant la durée du jeton.

L'extrait de code Java suivant montre comment initialiser un client Amazon Cognito et récupérer un jeton pour une identité authentifiée par le développeur.

```
// authenticate your end user as appropriate
// ....

// if authenticated, initialize a cognito client with your AWS developer credentials
AmazonCognitoIdentity identityClient = new AmazonCognitoIdentityClient(
    new BasicAWSCredentials("access_key_id", "secret_access_key")
);

// create a new request to retrieve the token for your end user
GetOpenIdTokenForDeveloperIdentityRequest request =
    new GetOpenIdTokenForDeveloperIdentityRequest();
request.setIdentityPoolId("YOUR_COGNITO_IDENTITY_POOL_ID");

request.setIdentityId("YOUR_COGNITO_IDENTITY_ID"); //optional, set this if your client
has an
                                                    //identity ID that you want to link
to this
                                                    //developer account

// set up your logins map with the username of your end user
HashMap<String,String> logins = new HashMap<>();
logins.put("YOUR_IDENTITY_PROVIDER_NAME", "YOUR_END_USER_IDENTIFIER");
request.setLogins(logins);
```

```
// optionally set token duration (in seconds)
request.setTokenDuration(60 * 151);
GetOpenIdTokenForDeveloperIdentityResult response =
    identityClient.getOpenIdTokenForDeveloperIdentity(request);

// obtain identity id and token to return to your client
String identityId = response.getIdentityId();
String token = response.getToken();

//code to return identity id and token to client
//...
```

Une fois que vous aurez suivi les étapes précédentes, vous devriez être en mesure d'intégrer les identités authentifiées par le développeur dans votre application. Si vous avez des problèmes ou des questions, n'hésitez pas à nous en faire part dans nos [forums](#).

Connexion à une identité sociale existante

Toutes les liaisons de fournisseurs lorsque vous utilisez des identités authentifiées par le développeur doivent être effectuées à partir de votre système backend. Pour associer une identité personnalisée à l'identité sociale d'un utilisateur (Login with Amazon, Sign in with Apple, Facebook ou Google), ajoutez le jeton du fournisseur d'identité à la carte des connexions lorsque vous appelez [GetOpenIdTokenForDeveloperIdentity](#). Pour rendre cela possible, lorsque vous appelez votre système backend depuis votre kit SDK client pour authentifier votre utilisateur final, transmettez également le jeton du fournisseur de réseaux sociaux de l'utilisateur final.

Par exemple, si vous essayez de lier une identité personnalisée à Facebook, vous ajoutez le jeton Facebook en plus de votre identificateur de fournisseur d'identité à la carte de connexions lorsque vous appelez `GetOpenIdTokenForDeveloperIdentity`.

```
logins.put("YOUR_IDENTITY_PROVIDER_NAME", "YOUR_END_USER_IDENTIFIER");
logins.put("graph.facebook.com", "END_USERS_FACEBOOK_ACCESSTOKEN");
```

Transition d'un fournisseur à un autre

Android

Votre application peut nécessiter de prendre en charge les identités authentifiées ou non authentifiées via les fournisseurs publics (Login with Amazon, Se connecter avec Apple, Facebook ou Google), ainsi que les identités authentifiées par le développeur. La principale différence entre les

identités authentifiées par le développeur et les autres (identités non authentifiées ou authentifiées via un fournisseur public) réside dans le mode de récupération de l'ID d'identité et du jeton. Pour les autres identités, l'application mobile interagit directement avec Amazon Cognito au lieu de communiquer avec votre système d'authentification. Par conséquent, l'application mobile doit être capable de prendre en charge deux flux distincts selon le choix de l'utilisateur de l'application. Pour cela, vous devez apporter des modifications au fournisseur d'identités personnalisées.

La méthode `refresh` vérifie le mappage des connexions. Si ce mappage n'est pas vide et qu'il contient une clé avec le nom du fournisseur du développement, appelez votre système backend. Sinon, appelez la `getIdentityId` méthode et renvoyez `null`.

```
public String refresh() {

    setToken(null);

    // If the logins map is not empty make a call to your backend
    // to get the token and identityId
    if (getProviderName() != null &&
        !this.loginsMap.isEmpty() &&
        this.loginsMap.containsKey(getProviderName())) {

        /**
         * This is where you would call your backend
         */

        // now set the returned identity id and token in the provider
        update(identityId, token);
        return token;

    } else {
        // Call getIdentityId method and return null
        this.getIdentityId();
        return null;
    }
}
```

De même, la méthode `getIdentityId` propose deux flux en fonction du contenu de la carte de connexions :

```
public String getIdentityId() {

    // Load the identityId from the cache
```



```

identityId = cachedIdentityId;

if (identityId == null) {

    // If the logins map is not empty make a call to your backend
    // to get the token and identityId

    if (getProviderName() != null && !this.loginsMap.isEmpty()
        && this.loginsMap.containsKey(getProviderName())) {

        /**
         * This is where you would call your backend
         */

        // now set the returned identity id and token in the provider
        update(identityId, token);
        return token;

    } else {
        // Otherwise call &COG; using getIdentityId of super class
        return super.getIdentityId();
    }

} else {
    return identityId;
}
}

```

iOS – objective-C

Votre application peut nécessiter de prendre en charge les identités authentifiées ou non authentifiées via les fournisseurs publics (Login with Amazon, Se connecter avec Apple, Facebook ou Google), ainsi que les identités authentifiées par le développeur. Pour ce faire, remplacez la [AWSIdentityProviderHelper](#) login méthode afin de pouvoir renvoyer la carte de connexion correcte en fonction du fournisseur d'identité actuel. Cet exemple montre comment vous pouvez alterner entre une identité non authentifiée, Facebook et authentifiée par le développeur.

```

- (AWSTask<NSDictionary<NSString *, NSString *> *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    } else if (/*logic to determine if user is Facebook*/){

```

```

        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook :
[FBSDKAccessToken currentAccessToken] }];
    }else {
        return [super logins];
    }
}

```

Lorsque vous passez d'une identité non authentifiée à une identité authentifiée, vous devez appeler `[credentialsProvider clearCredentials]`; pour forcer le kit SDK à obtenir les nouvelles informations d'identification authentifiées. Lorsque vous basculez entre deux fournisseurs authentifiés et que vous n'essayez pas de lier les deux fournisseurs (par exemple, si vous ne fournissez pas de jetons pour plusieurs fournisseurs dans votre dictionnaire de connexions), appelez `[credentialsProvider clearKeychain]`; Cela efface les informations d'identification et l'identité, et force le kit SDK à obtenir de nouvelles informations d'identification.

iOS – swift

Votre application peut nécessiter de prendre en charge les identités authentifiées ou non authentifiées via les fournisseurs publics (Login with Amazon, Se connecter avec Apple, Facebook ou Google), ainsi que les identités authentifiées par le développeur. Pour ce faire, remplacez la [AWSIdentityProviderHelper](#) `logins` méthode afin de pouvoir renvoyer la carte de connexion correcte en fonction du fournisseur d'identité actuel. Cet exemple montre comment vous pouvez alterner entre une identité non authentifiée, Facebook et authentifiée par le développeur.

```

override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else if (/*logic to determine if user is Facebook*/) {
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error: NSError(domain: "Facebook Login", code: -1 , userInfo:
["Facebook" : "No current Facebook access token"]))
    }else {
        return super.logins()
    }
}

```

Lorsque vous passez d'une identité non authentifiée à une identité authentifiée, vous devez appeler `credentialsProvider.clearCredentials()` pour forcer le kit SDK à obtenir les

nouvelles informations d'identification authentifiées. Lorsque vous basculez entre deux fournisseurs authentifiés et que vous n'essayez pas de lier les deux fournisseurs (vous ne fournissez pas de jetons pour plusieurs fournisseurs dans votre dictionnaire de connexions), vous devez appeler `credentialsProvider.clearKeychain()`. Cela efface les informations d'identification et l'identité, et force le kit SDK à obtenir de nouvelles informations d'identification.

Unity

Votre application peut nécessiter de prendre en charge les identités authentifiées ou non authentifiées via les fournisseurs publics (Login with Amazon, Se connecter avec Apple, Facebook ou Google), ainsi que les identités authentifiées par le développeur. La principale différence entre les identités authentifiées par le développeur et les autres (identités non authentifiées ou authentifiées via un fournisseur public) réside dans le mode de récupération de l'ID d'identité et du jeton. Pour les autres identités, l'application mobile interagit directement avec Amazon Cognito au lieu de communiquer avec votre système d'authentification. L'application mobile doit être capable de prendre en charge deux flux distincts selon le choix de l'utilisateur de l'application. Pour cela, vous devrez apporter des modifications au fournisseur d'identités personnalisées.

La méthode recommandée pour le faire dans Unity est d'étendre votre fournisseur d'identité à la `AmazonCognitoEnhancedIdentityProvider` place de `AbstractCognitoIdentityProvider` et d'appeler la `RefreshAsync` méthode parent au lieu de la vôtre au cas où l'utilisateur ne serait pas authentifié auprès de votre propre backend. Si l'utilisateur est authentifié, vous pouvez utiliser le flux expliqué précédemment.

Xamarin

Votre application peut nécessiter de prendre en charge les identités authentifiées ou non authentifiées via les fournisseurs publics (Login with Amazon, Se connecter avec Apple, Facebook ou Google), ainsi que les identités authentifiées par le développeur. La principale différence entre les identités authentifiées par le développeur et les autres (identités non authentifiées ou authentifiées via un fournisseur public) réside dans le mode de récupération de l'ID d'identité et du jeton. Pour les autres identités, l'application mobile interagit directement avec Amazon Cognito au lieu de communiquer avec votre système d'authentification. L'application mobile doit être capable de prendre en charge deux flux distincts selon le choix de l'utilisateur de l'application. Pour cela, vous devez apporter des modifications au fournisseur d'identités personnalisées.

Basculement d'utilisateurs non authentifiés à des utilisateurs authentifiés (groupes d'identités)

Les groupes d'identités Amazon Cognito prennent en charge les utilisateurs authentifiés et non authentifiés. Les utilisateurs non authentifiés bénéficient d'un accès à vos ressources AWS, même s'ils ne sont pas connectés avec l'un de vos fournisseurs d'identité (IdP). Ce degré d'accès est utile pour afficher du contenu aux utilisateurs avant qu'ils se connectent. Chaque utilisateur non authentifié comporte une identité unique dans le groupe d'identités, même s'il n'a pas été individuellement connecté et authentifié.

Cette section décrit le cas où votre utilisateur choisirait de basculer d'une connexion sous une identité non authentifiée à une connexion sous une identité authentifiée.

Android

Les utilisateurs peuvent se connecter à votre application en tant qu'invités non authentifiés. Ils peuvent finir par décider de se connecter via l'un des fournisseurs d'identité pris en charge. Amazon Cognito veille à ce qu'une ancienne identité conserve le même identifiant unique que la nouvelle, et que les données de profil sont fusionnées automatiquement.

Votre application est informée d'une fusion de profils via l'interface `IdentityChangedListener`. Mettez en œuvre la méthode `identityChanged` dans l'interface pour recevoir ces messages :

```
@override
public void identityChanged(String oldIdentityId, String newIdentityId) {
    // handle the change
}
```

iOS – objective-C

Les utilisateurs peuvent se connecter à votre application en tant qu'invités non authentifiés. Ils peuvent finir par décider de se connecter via l'un des fournisseurs d'identité pris en charge. Amazon Cognito veille à ce qu'une ancienne identité conserve le même identifiant unique que la nouvelle, et que les données de profil sont fusionnées automatiquement.

`NSNotificationCenter` informe votre application d'une fusion de profils :

```
[[NSNotificationCenter defaultCenter] addObserver:self
```

```
        selector:@selector(identityIdDidChange:)
        name:AWSCognitoIdentityIdChangedNotification
        object:nil];

-(void)identityDidChange:(NSNotification*)notification {
    NSDictionary *userInfo = notification.userInfo;
    NSLog(@"identity changed from %@ to %@",
        [userInfo objectForKey:AWSCognitoNotificationPreviousId],
        [userInfo objectForKey:AWSCognitoNotificationNewId]);
}
```

iOS – swift

Les utilisateurs peuvent se connecter à votre application en tant qu'invités non authentifiés. Ils peuvent finir par décider de se connecter via l'un des fournisseurs d'identité pris en charge. Amazon Cognito veille à ce qu'une ancienne identité conserve le même identifiant unique que la nouvelle, et que les données de profil sont fusionnées automatiquement.

NSNotificationCenter informe votre application d'une fusion de profils :

```
[NSNotificationCenter defaultCenter().addObserver(observer: self
    selector:"identityDidChange"
    name:AWSCognitoIdentityIdChangedNotification
    object:nil)

func identityDidChange(notification: NSNotification!) {
    if let userInfo = notification.userInfo as? [String: AnyObject] {
        print("identity changed from: \(userInfo[AWSCognitoNotificationPreviousId])
            to: \(userInfo[AWSCognitoNotificationNewId])")
    }
}
```

JavaScript

Utilisateur initialement non authentifié

Les utilisateurs commencent généralement avec le rôle non authentifié. Pour ce rôle, vous définissez la propriété des informations d'identification de votre objet de configuration sans propriété d'identifiants. Dans ce cas, votre configuration par défaut peut se présenter comme suit :

```
// set the default config object
var creds = new AWS.CognitoIdentityCredentials({
    IdentityPoolId: 'us-east-1:1699ebc0-7900-4099-b910-2df94f52a030'
});
AWS.config.credentials = creds;
```

Basculement vers un utilisateur authentifié

Lorsqu'un utilisateur non authentifié se connecte à un fournisseur d'identité (IdP) et que vous avez un jeton, vous pouvez faire basculer l'utilisateur non authentifié en utilisateur authentifié en appelant une fonction personnalisée qui met à jour l'objet des informations d'identification et ajoute le jeton d'identifiants :

```
// Called when an identity provider has a token for a logged in user
function userLoggedIn(providerName, token) {
    creds.params.Logins = creds.params.Logins || {};
    creds.params.Logins[providerName] = token;

    // Expire credentials to refresh them on the next request
    creds.expired = true;
}
```

Vous pouvez également créer un objet `CognitoIdentityCredentials`. Dans ce cas, vous devez réinitialiser les propriétés d'informations d'identification de n'importe quel objet de service existant pour refléter les informations de configuration des informations d'identification mises à jour. Consultez [Utilisation de l'objet de configuration globale](#).

Pour plus d'informations sur l'objet `CognitoIdentityCredentials`, consultez [AWS.CognitoIdentityCredentials](#) dans la référence d'API AWS SDK for JavaScript.

Unity

Les utilisateurs peuvent se connecter à votre application en tant qu'invités non authentifiés. Ils peuvent finir par décider de se connecter via l'un des fournisseurs d'identité pris en charge. Amazon Cognito veille à ce qu'une ancienne identité conserve le même identifiant unique que la nouvelle, et que les données de profil sont fusionnées automatiquement.

Pour être informé des fusions de profil, vous pouvez vous abonner à l'événement `IdentityChangedEvent` :

```
credentialsProvider.IdentityChangedEvent += delegate(object sender,
    CognitoAWSCredentials.IdentityChangedEventArgs e)
{
    // handle the change
    Debug.Log("Identity changed from " + e.OldIdentityId + " to " + e.NewIdentityId);
};
```

Xamarin

Les utilisateurs peuvent se connecter à votre application en tant qu'invités non authentifiés. Ils peuvent finir par décider de se connecter via l'un des fournisseurs d'identité pris en charge. Amazon Cognito veille à ce qu'une ancienne identité conserve le même identifiant unique que la nouvelle, et que les données de profil sont fusionnées automatiquement.

```
credentialsProvider.IdentityChangedEvent += delegate(object sender,
    CognitoAWSCredentials.IdentityChangedEventArgs e){
    // handle the change
    Console.WriteLine("Identity changed from " + e.OldIdentityId + " to " +
    e.NewIdentityId);
};
```

Amazon Cognito Sync

⚠ Si vous débutez avec Amazon Cognito Sync, utilisez [AWS AppSync](#). Comme Amazon Cognito Sync, AWS AppSync est un service permettant de synchroniser des données d'application sur différents appareils.

Il permet de synchroniser les données utilisateur telles que des préférences de l'application ou l'état d'un jeu. Il étend également ces capacités en permettant à plusieurs utilisateurs de se synchroniser et de collaborer en temps réel sur des données partagées.

Amazon Cognito Sync est un Service AWS et une bibliothèque client qui permet de synchroniser entre appareils des données utilisateur liées à une application. Amazon Cognito Sync peut synchroniser les données de profil utilisateur entre différents appareils mobiles et le web sans utiliser votre propre backend. Les bibliothèques client mettent en cache les données localement pour que votre application puisse lire et écrire des données quel que soit l'état de connectivité de l'appareil. Quand l'appareil est en ligne, vous pouvez synchroniser les données. Si vous configurez la synchronisation en mode push, vous pouvez avertir immédiatement les autres appareils qu'une mise à jour est disponible.

Pour plus d'informations sur la disponibilité régionale d'Amazon Cognito Identity, consultez [Disponibilité des services AWS par région](#).

Pour en savoir plus sur Amazon Cognito Sync, consultez les rubriques suivantes.

Rubriques

- [Démarrer avec Amazon Cognito Sync](#)
- [Synchronisation des données](#)
- [Gestion des rappels](#)
- [Synchronisation en mode Push](#)
- [Flux Amazon Cognito](#)
- [Événements Amazon Cognito](#)

Démarrer avec Amazon Cognito Sync

⚠ Si vous débutez avec Amazon Cognito Sync, utilisez [AWS AppSync](#). Comme Amazon Cognito Sync, AWS AppSync est un service permettant de synchroniser des données d'application sur différents appareils.

Il permet de synchroniser les données utilisateur telles que des préférences de l'application ou l'état d'un jeu. Il étend également ces capacités en permettant à plusieurs utilisateurs de se synchroniser et de collaborer en temps réel sur des données partagées.

Amazon Cognito Sync est un service AWS et une bibliothèque client qui permet la synchronisation entre appareils des données utilisateur liées à une application. Vous pouvez en tirer parti pour synchroniser les données de profil utilisateur sur les différents appareils mobiles et applications web. Les bibliothèques client mettent les données en cache localement pour que votre application puisse les lire et les modifier, quel que soit l'état de connectivité du dispositif. Lorsque l'appareil est en ligne, vous pouvez synchroniser les données et, si vous avez configuré la synchronisation en mode push, vous pouvez avertir immédiatement les autres appareils qu'une mise à jour est disponible.

Configurer un groupe d'identités dans Amazon Cognito

Amazon Cognito Sync nécessite un pool d'identités Amazon Cognito pour fournir des identités utilisateur. Avant d'utiliser la synchronisation Amazon Cognito, vous devez configurer une réserve d'identités. Pour créer un groupe d'identités et installer le kit SDK, consultez [Commencer à utiliser les pools d'identités Amazon Cognito](#).

Stocker et synchroniser les données

Une fois que vous avez configuré votre groupe d'identités et installé le kit SDK, vous pouvez commencer à stocker et à synchroniser les données entre les appareils. Pour de plus amples informations, veuillez consulter [Synchronisation des données](#).

Synchronisation des données

⚠ Si vous débutez avec Amazon Cognito Sync, utilisez [AWS AppSync](#). Comme Amazon Cognito Sync, AWS AppSync est un service permettant de synchroniser des données d'application sur différents appareils.

Il permet de synchroniser les données utilisateur telles que des préférences de l'application ou l'état d'un jeu. Il étend également ces capacités en permettant à plusieurs utilisateurs de se synchroniser et de collaborer en temps réel sur des données partagées.

Amazon Cognito vous permet d'enregistrer les données utilisateur dans des jeux de données qui contiennent des paires clé/valeur. Amazon Cognito associe ces données à une identité dans votre groupe d'identités, afin que votre application puisse y accéder indépendamment des connexions et des appareils. Pour synchroniser ces données entre le service Amazon Cognito et les appareils d'utilisateurs finaux, appelez la méthode de synchronisation. Chaque ensemble de données peut avoir une taille maximale de 1 Mo. Vous pouvez associer jusqu'à 20 ensembles de données à une identité.

Le client Amazon Cognito Sync crée un cache local pour les données d'identité. Quand votre application lit et écrit les clés, elle communique avec le cache local. Cette communication garantit la disponibilité immédiate sur l'appareil de toutes les modifications que vous y apportez, même lorsque vous êtes hors ligne. Lorsque la méthode de synchronisation est appelée, les modifications provenant du service sont envoyées au dispositif, tandis que toutes les modifications locales sont transmises au service. À ce stade, les modifications sont disponibles et peuvent être synchronisées sur d'autres appareils.

Initialisation du client Amazon Cognito Sync

Pour initialiser le client Amazon Cognito Sync, vous devez commencer par créer un fournisseur d'informations d'identification. Ce fournisseur d'informations d'identification acquiert des informations d'identification AWS temporaires pour permettre à votre application d'accéder à vos ressources AWS. Vous devez également importer les fichiers d'en-tête nécessaires. Pour initialiser le client Amazon Cognito Sync, procédez comme suit.

Android

1. Pour créer un fournisseur d'informations d'identification, suivez les instructions décrites dans [Obtention des informations d'identification](#).
2. Importez le package Amazon Cognito comme suit :

```
import com.amazonaws.mobileconnectors.cognito.*;
```

3. Initialisez Amazon Cognito Sync. Transmettez le contexte de l'application Android, l'ID du groupe d'identités, une Région AWS et un fournisseur d'informations d'identification Amazon Cognito initialisé comme suit :

```
CognitoSyncManager client = new CognitoSyncManager(
    getApplicationContext(),
    Regions.YOUR_REGION,
    credentialsProvider);
```

iOS : Objective-C

1. Pour créer un fournisseur d'informations d'identification, suivez les instructions décrites dans [Obtention des informations d'identification](#).
2. Importez AWSCore et Cognito, et initialisez AWSCognito comme suit :

```
#import <AWSiOSSDKv2/AWSCore.h>
#import <AWSCognitoSync/Cognito.h>

AWSCognito *syncClient = [AWSCognito defaultCognito];
```

3. Si vous utilisez CocoaPods, remplacez <AWSiOSSDKv2/AWSCore.h> par AWSCore.h. Suivez la même syntaxe pour l'importation Amazon Cognito.

iOS : Swift

1. Pour créer un fournisseur d'informations d'identification, suivez les instructions décrites dans [Obtention des informations d'identification](#).
2. Importez et initialisez AWSCognito comme suit :

```
import AWSCognito
let syncClient = AWSCognito.default()!
```

JavaScript

1. Télécharger le [gestionnaire Amazon Cognito Sync pour JavaScript](#).
2. Intégrez la bibliothèque du gestionnaire de synchronisation dans le projet.

3. Pour créer un fournisseur d'informations d'identification, suivez les instructions décrites dans [Obtention des informations d'identification](#).
4. Initialisez le gestionnaire de synchronisation comme suit :

```
var syncManager = new AWS.CognitoSyncManager();
```

Unity

1. Créez une instance de `CognitoAWSCredentials`, conformément aux instructions fournies dans [Obtention des informations d'identification](#).
2. Créez une instance de `CognitoSyncManager`. Transmettez l'objet `CognitoAwsCredentials` et `AmazonCognitoSyncConfig`, et incluez au moins la région définie, comme suit :

```
AmazonCognitoSyncConfig clientConfig = new AmazonCognitoSyncConfig { RegionEndpoint =  
    REGION };  
CognitoSyncManager syncManager = new CognitoSyncManager(credentials, clientConfig);
```

Xamarin

1. Créez une instance de `CognitoAWSCredentials`, conformément aux instructions fournies dans [Obtention des informations d'identification](#).
2. Créez une instance de `CognitoSyncManager`. Transmettez l'objet `CognitoAwsCredentials` et `AmazonCognitoSyncConfig`, et incluez au moins la région définie, comme suit :

```
AmazonCognitoSyncConfig clientConfig = new AmazonCognitoSyncConfig { RegionEndpoint =  
    REGION };  
CognitoSyncManager syncManager = new CognitoSyncManager(credentials, clientConfig);
```

Comprendre les jeux de données

Amazon Cognito organise les données de profil utilisateur en jeux de données. Chacun d'eux peut contenir jusqu'à 1 Mo de données sous la forme de paires clé/valeur. Un jeu de données représente l'entité la plus élémentaire que vous pouvez synchroniser. Les opérations de lecture et d'écriture sur un ensemble de données s'appliquent uniquement au stockage local tant que la méthode de

synchronisation n'est pas appelée. Amazon Cognito identifie un jeu de données à l'aide d'une chaîne unique. Vous pouvez créer un nouveau jeu de données ou en ouvrir un déjà existant, comme suit.

Android

```
Dataset dataset = client.openOrCreateDataset("datasetname");
```

Pour supprimer un jeu de données, commencez par appeler la méthode afin de le supprimer du stockage local, puis appelez la méthode `synchronize` pour supprimer le jeu de données d'Amazon Cognito, comme suit :

```
dataset.delete();  
dataset.synchronize(syncCallback);
```

iOS : Objective-C

```
AWSCognitoDataset *dataset = [syncClient openOrCreateDataset:@"myDataSet"];
```

Pour supprimer un jeu de données, commencez par appeler la méthode afin de le supprimer du stockage local, puis appelez la méthode `synchronize` pour supprimer le jeu de données d'Amazon Cognito, comme suit :

```
[dataset clear];  
[dataset synchronize];
```

iOS : Swift

```
let dataset = syncClient.openOrCreateDataset("myDataSet")!
```

Pour supprimer un jeu de données, commencez par appeler la méthode afin de le supprimer du stockage local, puis appelez la méthode `synchronize` comme suit, pour supprimer le jeu de données d'Amazon Cognito :

```
dataset.clear()  
dataset.synchronize()
```

JavaScript

```
syncManager.openOrCreateDataset('myDatasetName', function(err, dataset) {
```

```
// ...  
});
```

Unity

```
string myValue = dataset.Get("myKey");  
dataset.Put("myKey", "newValue");
```

Pour supprimer une clé d'un jeu de données, utilisez `Remove` comme suit :

```
dataset.Remove("myKey");
```

Xamarin

```
Dataset dataset = syncManager.OpenOrCreateDataset("myDatasetName");
```

Pour supprimer un jeu de données, commencez par appeler la méthode afin de le supprimer du stockage local, puis appelez la méthode `synchronize` pour supprimer le jeu de données d'Amazon Cognito, comme suit :

```
dataset.Delete();  
dataset.SynchronizeAsync();
```

Lecture et écriture de données dans les jeux de données

Les jeux de données Amazon Cognito fonctionnent comme des dictionnaires, avec des valeurs accessibles par clé. Vous pouvez lire, ajouter ou modifier les clés et les valeurs d'un jeu de données comme s'il s'agissait d'un dictionnaire, comme illustré dans les exemples suivants.

Notez que les valeurs que vous écrivez dans un jeu de données affectent uniquement la version locale mise en cache des données tant que vous n'appellez pas la méthode de synchronisation.

Android

```
String value = dataset.get("myKey");  
dataset.put("myKey", "my value");
```

iOS : Objective-C

```
[dataset setString:@"my value" forKey:@"myKey"];  
NSString *value = [dataset stringForKey:@"myKey"];
```

iOS : Swift

```
dataset.setString("my value", forKey:"myKey")  
let value = dataset.stringForKey("myKey")
```

JavaScript

```
dataset.get('myKey', function(err, value) {  
    console.log('myRecord: ' + value);  
});  
  
dataset.put('newKey', 'newValue', function(err, record) {  
    console.log(record);  
});  
  
dataset.remove('oldKey', function(err, record) {  
    console.log(success);  
});
```

Unity

```
string myValue = dataset.Get("myKey");  
dataset.Put("myKey", "newValue");
```

Xamarin

```
//obtain a value  
string myValue = dataset.Get("myKey");  
  
// Create a record in a dataset and synchronize with the server  
dataset.OnSyncSuccess += SyncSuccessCallback;  
dataset.Put("myKey", "myValue");  
dataset.SynchronizeAsync();
```

```
void SyncSuccessCallback(object sender, SyncSuccessEventArgs e) {  
    // Your handler code here  
}
```

Android

Pour supprimer des clés d'un jeu de données, utilisez la méthode `remove` comme suit :

```
dataset.remove("myKey");
```

iOS : Objective-C

Pour supprimer une clé d'un jeu de données, utilisez `removeObjectForKey` comme suit :

```
[dataset removeObjectForKey:@"myKey"];
```

iOS : Swift

Pour supprimer une clé d'un jeu de données, utilisez `removeObjectForKey` comme suit :

```
dataset.removeObjectForKey("myKey")
```

Unity

Pour supprimer une clé d'un jeu de données, utilisez `Remove` comme suit :

```
dataset.Remove("myKey");
```

Xamarin

Vous pouvez utiliser `Remove` pour supprimer une clé à partir d'un ensemble de données :

```
dataset.Remove("myKey");
```

Synchronisation des données locales avec le magasin de synchronisation

Android

La méthode `synchronize` compare les données locales mises en cache aux données stockées dans le magasin Amazon Cognito Sync. Les modifications à distance sont extraites du magasin Amazon Cognito Sync. Le cas échéant, la résolution des conflits est appelée, et les valeurs mises à jour sur l'appareil sont transférées au service. Pour synchroniser un ensemble de données, appelez sa méthode `synchronize` :

```
dataset.synchronize(syncCallback);
```

La méthode `synchronize` reçoit une implémentation de l'interface `SyncCallback`, comme décrit ci-après.

La méthode `synchronizeOnConnectivity()` tente d'effectuer la synchronisation lorsque la connectivité est disponible. Si la connectivité est disponible immédiatement, `synchronizeOnConnectivity()` se comporte comme `synchronize()`. Sinon, il est à l'affut des modifications de connectivité et effectue une synchronisation une fois que la connexion est disponible. Si `synchronizeOnConnectivity()` est appelé plusieurs fois, seule la dernière demande de synchronisation est conservée, et seul le dernier rappel est déclenché. Si l'ensemble de données ou le rappel est nettoyé de la mémoire, cette méthode n'effectue pas une synchronisation, et le rappel n'est pas déclenché.

Pour en savoir plus sur la synchronisation des ensembles de données et sur les différents rappels, consultez la section [Gestion des rappels](#).

iOS : Objective-C

La méthode `synchronize` compare les données locales mises en cache aux données stockées dans le magasin Amazon Cognito Sync. Les modifications à distance sont extraites du magasin Amazon Cognito Sync. Le cas échéant, la résolution des conflits est appelée, et les valeurs mises à jour sur l'appareil sont transférées au service. Pour synchroniser un ensemble de données, appelez sa méthode `synchronize` :

La méthode `synchronize` est asynchrone et renvoie un objet `AWSTask` pour traiter la réponse :

```
[[dataset synchronize] continueWithBlock:^id(AWSTask *task) {  
    if (task.isCancelled) {  
        // Task cancelled.  
    } else if (task.error) {  
        // Error while executing task.  
    }  
}
```

```
    } else {  
        // Task succeeded. The data was saved in the sync store.  
    }  
    return nil;  
}];
```

La méthode `synchronizeOnConnectivity` tente d'effectuer la synchronisation lorsque le dispositif est connecté. Tout d'abord, `synchronizeOnConnectivity` vérifie si le dispositif est en ligne. Si tel est le cas, il appelle immédiatement la synchronisation et renvoie l'objet `AWSTask` associé à la tentative.

Si le dispositif n'est pas connecté, `synchronizeOnConnectivity` 1) programme la synchronisation pour qu'elle ait lieu la prochaine fois qu'il sera en ligne et 2) renvoie un objet `AWSTask` avec un résultat nul. La synchronisation programmée est uniquement valide pour le cycle de vie de l'objet de l'ensemble de données. Les données ne seront pas synchronisées si l'application est arrêtée avant d'avoir récupéré la connectivité. Si vous souhaitez recevoir une notification lorsque des événements se produisent au cours de la synchronisation programmée, vous devez ajouter des observateurs des notifications trouvées dans `AWSCognito`.

Pour en savoir plus sur la synchronisation des ensembles de données et sur les différents rappels, consultez la section [Gestion des rappels](#).

iOS : Swift

La méthode `synchronize` compare les données locales mises en cache aux données stockées dans le magasin Amazon Cognito Sync. Les modifications à distance sont extraites du magasin Amazon Cognito Sync. Le cas échéant, la résolution des conflits est appelée, et les valeurs mises à jour sur l'appareil sont transférées au service. Pour synchroniser un ensemble de données, appelez sa méthode `synchronize` :

La méthode `synchronize` est asynchrone et renvoie un objet `AWSTask` pour traiter la réponse :

```
dataset.synchronize().continueWith(block: { (task) -> AnyObject? in  
  
    if task.isCancelled {  
        // Task cancelled.  
    } else if task.error != nil {  
        // Error while executing task  
    } else {  
        // Task succeeded. The data was saved in the sync store.  
    }  
}
```

```
        return task  
    })
```

La méthode `synchronizeOnConnectivity` tente d'effectuer la synchronisation lorsque le dispositif est connecté. Tout d'abord, `synchronizeOnConnectivity` vérifie si le dispositif est en ligne. Si tel est le cas, il appelle immédiatement `synchronize` et renvoie l'objet `AWSTask` associé à la tentative.

Si le dispositif n'est pas connecté, `synchronizeOnConnectivity` 1) programme la synchronisation pour qu'elle ait lieu la prochaine fois qu'il sera en ligne et 2) renvoie un objet `AWSTask` avec un résultat nul. La synchronisation programmée est uniquement valide pour le cycle de vie de l'objet de l'ensemble de données. Les données ne seront pas synchronisées si l'application est arrêtée avant d'avoir récupéré la connectivité. Si vous souhaitez recevoir une notification lorsque des événements se produisent au cours de la synchronisation programmée, vous devez ajouter des observateurs des notifications trouvées dans `AWSCognito`.

Pour en savoir plus sur la synchronisation des ensembles de données et sur les différents rappels, consultez la section [Gestion des rappels](#).

JavaScript

La méthode `synchronize` compare les données locales mises en cache aux données stockées dans le magasin Amazon Cognito Sync. Les modifications à distance sont extraites du magasin Amazon Cognito Sync. Le cas échéant, la résolution des conflits est appelée, et les valeurs mises à jour sur l'appareil sont transférées au service. Pour synchroniser un ensemble de données, appelez sa méthode `synchronize` :

```
dataset.synchronize();
```

Pour en savoir plus sur la synchronisation des ensembles de données et sur les différents rappels, consultez la section [Gestion des rappels](#).

Unity

La méthode de synchronisation compare les données locales mises en cache aux données stockées dans le magasin Amazon Cognito Sync. Les modifications à distance sont extraites du magasin Amazon Cognito Sync. Le cas échéant, la résolution des conflits est appelée, et les valeurs mises à jour sur l'appareil sont transférées au service. Pour synchroniser un ensemble de données, appelez sa méthode `synchronize` :

```
dataset.Synchronize();
```

La synchronisation est exécutée de façon asynchrone et finit par appeler l'un des différents rappels que vous pouvez spécifier dans l'ensemble de données.

Pour en savoir plus sur la synchronisation des ensembles de données et sur les différents rappels, consultez la section [Gestion des rappels](#).

Xamarin

La méthode `synchronize` compare les données locales mises en cache aux données stockées dans le magasin Amazon Cognito Sync. Les modifications à distance sont extraites du magasin Amazon Cognito Sync. Le cas échéant, la résolution des conflits est appelée, et les valeurs mises à jour sur l'appareil sont transférées au service. Pour synchroniser un ensemble de données, appelez sa méthode `synchronize` :

```
dataset.SynchronizeAsync();
```

Pour en savoir plus sur la synchronisation des ensembles de données et sur les différents rappels, consultez la section [Gestion des rappels](#).

Gestion des rappels

⚠ Si vous débutez avec Amazon Cognito Sync, utilisez [AWS AppSync](#). Comme Amazon Cognito Sync, AWS AppSync est un service permettant de synchroniser des données d'application sur différents appareils.

Il permet de synchroniser les données utilisateur telles que des préférences de l'application ou l'état d'un jeu. Il étend également ces capacités en permettant à plusieurs utilisateurs de se synchroniser et de collaborer en temps réel sur des données partagées.

Cette section décrit comment traiter les rappels.

Android

Interface SyncCallback

En mettant en œuvre l'interface `SyncCallback`, vous pouvez recevoir des notifications concernant la synchronisation de l'ensemble de données dans votre application. Cette dernière peut ensuite prendre des décisions actives sur la suppression des données locales, la fusion des profils non authentifiés et authentifiés et la résolution des conflits de synchronisation. Vous devez implémenter les méthodes suivantes, qui sont requises par l'interface :

- `onSuccess()`
- `onFailure()`
- `onConflict()`
- `onDatasetDeleted()`
- `onDatasetsMerged()`

Notez que, si vous ne voulez pas spécifier tous les rappels, vous pouvez également utiliser la classe `DefaultSyncCallback` qui fournit des implémentations vides par défaut pour chacun d'eux.

`onSuccess`

Le rappel `onSuccess()` est déclenché quand un ensemble de données est téléchargé avec succès depuis l'espace de synchronisation.

```
@Override
public void onSuccess(Dataset dataset, List<Record> newRecords) {
}
```

`onFailure`

`onFailure()` est appelé si une exception s'est produite lors de la synchronisation.

```
@Override
public void onFailure(DataStorageException dse) {
}
```

`onConflict`

Des conflits peuvent survenir si la même clé a été modifiée sur le stockage local et dans l'espace de synchronisation. La méthode `onConflict()` gère la résolution des conflits. Si vous n'implémentez pas cette méthode, le client Amazon Cognito Sync utilise par défaut la modification la plus récente.

```
@Override
```

```
public boolean onConflict(Dataset dataset, final List<SyncConflict> conflicts) {
    List<Record> resolvedRecords = new ArrayList<Record>();
    for (SyncConflict conflict : conflicts) {
        /* resolved by taking remote records */
        resolvedRecords.add(conflict.resolveWithRemoteRecord());

        /* alternately take the local records */
        // resolvedRecords.add(conflict.resolveWithLocalRecord());

        /* or customer logic, say concatenate strings */
        // String newValue = conflict.getRemoteRecord().getValue()
        //     + conflict.getLocalRecord().getValue();
        // resolvedRecords.add(conflict.resolveWithValue(newValue);
    }
    dataset.resolve(resolvedRecords);

    // return true so that synchronize() is retried after conflicts are resolved
    return true;
}
```

onDatasetDeleted

Lorsqu'un jeu de données est supprimé, le client Amazon Cognito utilise l'interface `SyncCallback` pour vérifier si la version locale mise en cache doit également être supprimée. Implémentez la méthode `onDatasetDeleted()` pour que le kit SDK du client sache quoi faire avec les données locales.

```
@Override
public boolean onDatasetDeleted(Dataset dataset, String datasetName) {
    // return true to delete the local copy of the dataset
    return true;
}
```

onDatasetMerged

Lorsque deux identités qui n'étaient pas connectées sont liées, tous les ensembles de données sont fusionnés. Les applications sont informées de la fusion par le biais de la méthode `onDatasetsMerged()` :

```
@Override
public boolean onDatasetsMerged(Dataset dataset, List<String> datasetNames) {
    // return false to handle Dataset merge outside the synchronization callback
}
```

```
    return false;
}
```

iOS : Objective-C

Notifications de synchronisation

Le client Amazon Cognito émet un certain nombre d'événements `NSNotification` lors d'un appel de synchronisation. Vous pouvez demander à surveiller ces notifications via le `NSNotificationCenter` standard :

```
[NSNotificationCenter defaultCenter]
    addObserver:self
    selector:@selector(myNotificationHandler:)
    name:NOTIFICATION_TYPE
    object:nil];
```

Amazon Cognito prend en charge les cinq types de notifications énumérés ci-dessous.

`AWSCognitoDidStartSynchronizeNotification`

Appelé lorsqu'une opération de synchronisation commence. L'objet `userInfo` contient l'ensemble de données de la clé, qui correspond au nom de l'ensemble de données en cours de synchronisation.

`AWSCognitoDidEndSynchronizeNotification`

Appelé lorsqu'une opération de synchronisation se termine (avec succès ou non). L'objet `userInfo` contient l'ensemble de données de la clé, qui correspond au nom de l'ensemble de données en cours de synchronisation.

`AWSCognitoDidFailToSynchronizeNotification`

Appelé lorsqu'une opération de synchronisation échoue. L'objet `userInfo` contient l'ensemble de données de la clé, qui correspond au nom de l'ensemble de données en cours de synchronisation, et l'erreur de clé qui contient l'erreur qui a provoqué l'échec.

`AWSCognitoDidChangeRemoteValueNotification`

Appelé lorsque les modifications locales sont correctement transmises à Amazon Cognito. L'objet `userInfo` contient l'ensemble de données des clés, qui correspond au nom de l'ensemble de données en cours de synchronisation, et les clés qui contiennent un `NSArray` des clés d'enregistrement qui ont été transmises.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Appelé lorsqu'une valeur locale change en raison d'une opération de synchronisation. L'objet `userInfo` contient l'ensemble de données des clés, qui correspond au nom de l'ensemble de données en cours de synchronisation, et les clés qui contiennent un `NSArray` des clés d'enregistrement qui ont été modifiées.

Gestionnaire de résolution des conflits

Lors d'une opération de synchronisation, des conflits peuvent survenir si la même clé a été modifiée sur le stockage local et dans l'espace de synchronisation. Si vous n'avez pas défini un gestionnaire de résolution des conflits, Amazon Cognito choisit par défaut la mise à jour la plus récente.

L'implémentation et l'attribution d'un gestionnaire `AWSCognitoRecordConflictHandler` vous permet de modifier la résolution des conflits par défaut. Le conflit du paramètre d'entrée `AWSCognitoConflict` contient un objet `AWSCognitoRecord` pour les données locales mises en cache, ainsi que pour l'enregistrement conflictuel dans l'espace de synchronisation. Avec `AWSCognitoConflict`, vous pouvez résoudre le conflit avec l'enregistrement local [`conflict resolveWithLocalRecord`], l'enregistrement distant [`conflict resolveWithRemoteRecord`] ou une nouvelle valeur [`conflict resolveWithValue:value`]. Le renvoi de la valeur nulle à partir de cette méthode empêche la synchronisation de se poursuivre et les conflits seront présentés à nouveau la prochaine fois que le processus de synchronisation démarrera.

Vous pouvez définir le gestionnaire de résolution des conflits au niveau du client :

```
client.conflictHandler = ^AWSCognitoResolvedConflict* (NSString *datasetName,
    AWSCognitoConflict *conflict) {
    // always choose local changes
    return [conflict resolveWithLocalRecord];
};
```

Ou au niveau de l'ensemble de données :

```
dataset.conflictHandler = ^AWSCognitoResolvedConflict* (NSString *datasetName,
    AWSCognitoConflict *conflict) {
    // override and always choose remote changes
    return [conflict resolveWithRemoteRecord];
};
```

Gestionnaire de suppression de l'ensemble de données

Lorsqu'un jeu de données est supprimé, le client Amazon Cognito utilise l'`AWSCognitoDatasetDeletedHandler` pour vérifier si la version locale mise en cache doit également être supprimée. Si aucun gestionnaire `AWSCognitoDatasetDeletedHandler` n'est mis en œuvre, les données locales seront purgées automatiquement. Implémentez un gestionnaire `AWSCognitoDatasetDeletedHandler` si vous souhaitez conserver une copie des données locales avant l'effacement ou si vous souhaitez conserver les données locales.

Vous pouvez définir le gestionnaire de suppression de l'ensemble de données au niveau du client :

```
client.datasetDeletedHandler = ^BOOL (NSString *datasetName) {
    // make a backup of the data if you choose
    ...
    // delete the local data (default behavior)
    return YES;
};
```

Ou au niveau de l'ensemble de données :

```
dataset.datasetDeletedHandler = ^BOOL (NSString *datasetName) {
    // override default and keep the local data
    return NO;
};
```

Gestionnaire de fusion de l'ensemble de données

Lorsque deux identités qui n'étaient pas connectées sont liées, tous les ensembles de données sont fusionnés. Les applications sont informées de la fusion par le biais du gestionnaire `DatasetMergeHandler`. Le gestionnaire reçoit le nom de l'ensemble de données racine, ainsi qu'un tableau de noms d'ensembles de données qui sont marqués comme étant des fusions de l'ensemble de données racine.

Si aucun gestionnaire `DatasetMergeHandler` n'est mis en œuvre, ces ensembles de données seront ignorés, mais continueront à utiliser l'espace autorisant un maximum de 20 ensembles de données.

Vous pouvez définir le gestionnaire de fusion de l'ensemble de données au niveau du client :

```
client.datasetMergedHandler = ^(NSString *datasetName, NSArray *datasets) {
    // Blindly delete the datasets
    for (NSString *name in datasets) {
```

```

        AWSCognitoDataset *merged = [[AWSCognito defaultCognito]
openOrCreateDataset:name];
        [merged clear];
        [merged synchronize];
    }
};

```

Ou au niveau de l'ensemble de données :

```

dataset.datasetMergedHandler = ^(NSString *datasetName, NSArray *datasets) {
    // Blindly delete the datasets
    for (NSString *name in datasets) {
        AWSCognitoDataset *merged = [[AWSCognito defaultCognito]
openOrCreateDataset:name];
        // do something with the data if it differs from existing dataset
        ...
        // now delete it
        [merged clear];
        [merged synchronize];
    }
};

```

iOS : Swift

Notifications de synchronisation

Le client Amazon Cognito émet un certain nombre d'événements `NSNotification` lors d'un appel de synchronisation. Vous pouvez demander à surveiller ces notifications via le `NSNotificationCenter` standard :

```

NSNotificationCenter.defaultCenter().addObserver(observer: self,
selector: "myNotificationHandler",
name:NOTIFICATION_TYPE,
object:nil)

```

Amazon Cognito prend en charge les cinq types de notifications énumérés ci-dessous.

`AWSCognitoDidStartSynchronizeNotification`

Appelé lorsqu'une opération de synchronisation commence. L'objet `userInfo` contient l'ensemble de données de la clé, qui correspond au nom de l'ensemble de données en cours de synchronisation.

AWSCognitoDidEndSynchronizeNotification

Appelé lorsqu'une opération de synchronisation se termine (avec succès ou non). L'objet `userInfo` contient l'ensemble de données de la clé, qui correspond au nom de l'ensemble de données en cours de synchronisation.

AWSCognitoDidFailToSynchronizeNotification

Appelé lorsqu'une opération de synchronisation échoue. L'objet `userInfo` contient l'ensemble de données de la clé, qui correspond au nom de l'ensemble de données en cours de synchronisation, et l'erreur de clé qui contient l'erreur qui a provoqué l'échec.

AWSCognitoDidChangeRemoteValueNotification

Appelé lorsque les modifications locales sont correctement transmises à Amazon Cognito. L'objet `userInfo` contient l'ensemble de données des clés, qui correspond au nom de l'ensemble de données en cours de synchronisation, et les clés qui contiennent un `NSArray` des clés d'enregistrement qui ont été transmises.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Appelé lorsqu'une valeur locale change en raison d'une opération de synchronisation. L'objet `userInfo` contient l'ensemble de données des clés, qui correspond au nom de l'ensemble de données en cours de synchronisation, et les clés qui contiennent un `NSArray` des clés d'enregistrement qui ont été modifiées.

Gestionnaire de résolution des conflits

Lors d'une opération de synchronisation, des conflits peuvent survenir si la même clé a été modifiée sur le stockage local et dans l'espace de synchronisation. Si vous n'avez pas défini un gestionnaire de résolution des conflits, Amazon Cognito choisit par défaut la mise à jour la plus récente.

L'implémentation et l'attribution d'un gestionnaire `AWSCognitoRecordConflictHandler` vous permet de modifier la résolution des conflits par défaut. Le conflit du paramètre d'entrée `AWSCognitoConflict` contient un objet `AWSCognitoRecord` pour les données locales mises en cache, ainsi que pour l'enregistrement conflictuel dans l'espace de synchronisation. Avec `AWSCognitoConflict`, vous pouvez résoudre le conflit avec l'enregistrement local [`conflict.resolveWithLocalRecord`], l'enregistrement distant [`conflict.resolveWithRemoteRecord`] ou une nouvelle valeur [`conflict.resolveWithValue:value`]. Le renvoi de la valeur nulle à partir de cette méthode empêche la synchronisation de se poursuivre et les conflits seront présentés à nouveau la prochaine fois que le processus de synchronisation démarrera.

Vous pouvez définir le gestionnaire de résolution des conflits au niveau du client :

```
client.conflictHandler = {
    (datasetName: String?, conflict: AWSCognitoConflict?) ->
    AWSCognitoResolvedConflict? in
    return conflict.resolveWithLocalRecord()
}
```

Ou au niveau de l'ensemble de données :

```
dataset.conflictHandler = {
    (datasetName: String?, conflict: AWSCognitoConflict?) ->
    AWSCognitoResolvedConflict? in
    return conflict.resolveWithLocalRecord()
}
```

Gestionnaire de suppression de l'ensemble de données

Lorsqu'un jeu de données est supprimé, le client Amazon Cognito utilise l'`AWSCognitoDatasetDeletedHandler` pour vérifier si la version locale mise en cache doit également être supprimée. Si aucun gestionnaire `AWSCognitoDatasetDeletedHandler` n'est mis en œuvre, les données locales seront purgées automatiquement. Implémentez un gestionnaire `AWSCognitoDatasetDeletedHandler` si vous souhaitez conserver une copie des données locales avant l'effacement ou si vous souhaitez conserver les données locales.

Vous pouvez définir le gestionnaire de suppression de l'ensemble de données au niveau du client :

```
client.datasetDeletedHandler = {
    (datasetName: String!) -> Bool in
    // make a backup of the data if you choose
    ...
    // delete the local data (default behaviour)
    return true
}
```

Ou au niveau de l'ensemble de données :

```
dataset.datasetDeletedHandler = {
    (datasetName: String!) -> Bool in
    // make a backup of the data if you choose
    ...
}
```

```
// delete the local data (default behaviour)
return true
}
```

Gestionnaire de fusion des jeux de données

Lorsque deux identités qui n'étaient pas connectées sont liées, tous les ensembles de données sont fusionnés. Les applications sont informées de la fusion par le biais du gestionnaire `DatasetMergeHandler`. Le gestionnaire reçoit le nom de l'ensemble de données racine, ainsi qu'un tableau de noms d'ensembles de données qui sont marqués comme étant des fusions de l'ensemble de données racine.

Si aucun gestionnaire `DatasetMergeHandler` n'est mis en œuvre, ces ensembles de données seront ignorés, mais continueront à utiliser l'espace autorisant un maximum de 20 ensembles de données.

Vous pouvez définir le gestionnaire de fusion de l'ensemble de données au niveau du client :

```
client.datasetMergedHandler = {
    (datasetName: String!, datasets: [AnyObject]!) -> Void in
    for nameObject in datasets {
        if let name = nameObject as? String {
            let merged = AWSCognito.defaultCognito().openOrCreateDataset(name)
            merged.clear()
            merged.synchronize()
        }
    }
}
```

Ou au niveau de l'ensemble de données :

```
dataset.datasetMergedHandler = {
    (datasetName: String!, datasets: [AnyObject]!) -> Void in
    for nameObject in datasets {
        if let name = nameObject as? String {
            let merged = AWSCognito.defaultCognito().openOrCreateDataset(name)
            // do something with the data if it differs from existing dataset
            ...
            // now delete it
            merged.clear()
            merged.synchronize()
        }
    }
}
```

```
}  
}
```

JavaScript

Rappels de synchronisation

Lorsque vous effectuez une opération `synchronize()` au niveau d'un ensemble de données, vous pouvez spécifier des rappels pour gérer chacun des états suivants :

```
dataset.synchronize({  
  
  onSuccess: function(dataset, newRecords) {  
    //...  
  },  
  
  onFailure: function(err) {  
    //...  
  },  
  
  onConflict: function(dataset, conflicts, callback) {  
    //...  
  },  
  
  onDatasetDeleted: function(dataset, datasetName, callback) {  
    //...  
  },  
  
  onDatasetMerged: function(dataset, datasetNames, callback) {  
    //...  
  }  
  
});
```

onSuccess()

Le rappel `onSuccess()` est déclenché quand un ensemble de données est mis à jour avec succès depuis l'espace de synchronisation. Si vous ne définissez pas un rappel, la synchronisation aboutit en mode silencieux.

```
onSuccess: function(dataset, newRecords) {  
  console.log('Successfully synchronized ' + newRecords.length + ' new records.');
```

```
}
```

onFailure()

`onFailure()` est appelé si une exception s'est produite lors de la synchronisation. Si vous ne définissez pas un rappel, la synchronisation échoue en mode silencieux.

```
onFailure: function(err) {
  console.log('Synchronization failed.');
```

```
  console.log(err);
}
```

onConflict()

Des conflits peuvent survenir si la même clé a été modifiée sur le stockage local et dans l'espace de synchronisation. La méthode `onConflict()` gère la résolution des conflits. Si vous n'implémentez cette méthode, la synchronisation est interrompue quand il y a un conflit.

```
onConflict: function(dataset, conflicts, callback) {

  var resolved = [];

  for (var i=0; i<conflicts.length; i++) {

    // Take remote version.
    resolved.push(conflicts[i].resolveWithRemoteRecord());

    // Or... take local version.
    // resolved.push(conflicts[i].resolveWithLocalRecord());

    // Or... use custom logic.
    // var newValue = conflicts[i].getRemoteRecord().getValue() +
conflicts[i].getLocalRecord().getValue();
    // resolved.push(conflicts[i].resovleWithValue(newValue);

  }

  dataset.resolve(resolved, function() {
    return callback(true);
  });

  // Or... callback false to stop the synchronization process.
```

```
// return callback(false);  
  
}
```

onDatasetDeleted()

Lorsqu'un jeu de données est supprimé, le client Amazon Cognito utilise le rappel `onDatasetDeleted()` pour décider si la version locale mise en cache doit également être supprimée. Par défaut, l'ensemble de données n'est pas supprimé.

```
onDatasetDeleted: function(dataset, datasetName, callback) {  
  
    // Return true to delete the local copy of the dataset.  
    // Return false to handle deleted datasets outside the synchronization callback.  
  
    return callback(true);  
  
}
```

onDatasetMerged()

Lorsque deux identités qui n'étaient pas connectées sont liées, tous les ensembles de données sont fusionnés. Les applications sont informées de la fusion par le biais du rappel `onDatasetsMerged()`.

```
onDatasetMerged: function(dataset, datasetNames, callback) {  
  
    // Return true to continue the synchronization process.  
    // Return false to handle dataset merges outside the synchronization callback.  
  
    return callback(false);  
  
}
```

Unity

Une fois que vous ouvrez ou créez un ensemble de données, vous pouvez définir différents rappels qui se déclencheront lorsque vous utilisez la méthode de synchronisation. Voici comment enregistrer ces rappels :

```
dataset.OnSyncSuccess += this.HandleSyncSuccess;
```



```
dataset.OnSyncFailure += this.HandleSyncFailure;
dataset.OnSyncConflict = this.HandleSyncConflict;
dataset.OnDatasetMerged = this.HandleDatasetMerged;
dataset.OnDatasetDeleted = this.HandleDatasetDeleted;
```

Notez que `SyncSuccess` et `SyncFailure` utilisent `+=` au lieu de `=`. Vous pouvez donc vous abonner à plusieurs rappels.

OnSyncSuccess

Le rappel `OnSyncSuccess` est déclenché quand un ensemble de données est mis à jour avec succès depuis le cloud. Si vous ne définissez pas un rappel, la synchronisation aboutit en mode silencieux.

```
private void HandleSyncSuccess(object sender, SyncSuccessEvent e)
{
    // Continue with your game flow, display the loaded data, etc.
}
```

OnSyncFailure

`OnSyncFailure` est appelé si une exception s'est produite lors de la synchronisation. Si vous ne définissez pas un rappel, la synchronisation échoue en mode silencieux.

```
private void HandleSyncFailure(object sender, SyncFailureEvent e)
{
    Dataset dataset = sender as Dataset;
    if (dataset.Metadata != null) {
        Debug.Log("Sync failed for dataset : " + dataset.Metadata.DatasetName);
    } else {
        Debug.Log("Sync failed");
    }
    // Handle the error
    Debug.LogException(e.Exception);
}
```

OnSyncConflict

Des conflits peuvent survenir si la même clé a été modifiée sur le stockage local et dans l'espace de synchronisation. Le rappel `OnSyncConflict` gère la résolution des conflits. Si vous n'implémentez cette méthode, la synchronisation est interrompue quand il y a un conflit.

```
private bool HandleSyncConflict(Dataset dataset, List < SyncConflict > conflicts)
{
    if (dataset.Metadata != null) {
        Debug.LogWarning("Sync conflict " + dataset.Metadata.DatasetName);
    } else {
        Debug.LogWarning("Sync conflict");
    }
    List < Amazon.CognitoSync.SyncManager.Record > resolvedRecords = new List <
Amazon.CognitoSync.SyncManager.Record > ();
    foreach(SyncConflict conflictRecord in conflicts) {
        // SyncManager provides the following default conflict resolution methods:
        //     ResolveWithRemoteRecord - overwrites the local with remote records
        //     ResolveWithLocalRecord - overwrites the remote with local records
        //     ResolveWithValue - to implement your own logic
        resolvedRecords.Add(conflictRecord.ResolveWithRemoteRecord());
    }
    // resolves the conflicts in local storage
    dataset.Resolve(resolvedRecords);
    // on return true the synchronize operation continues where it left,
    //     returning false cancels the synchronize operation
    return true;
}
```

OnDatasetDeleted

Lorsqu'un jeu de données est supprimé, le client Amazon Cognito utilise le rappel

`OnDatasetDeleted` pour décider si la version locale mise en cache doit également être supprimée.

Par défaut, l'ensemble de données n'est pas supprimé.

```
private bool HandleDatasetDeleted(Dataset dataset)
{
    Debug.Log(dataset.Metadata.DatasetName + " Dataset has been deleted");
    // Do clean up if necessary
    // returning true informs the corresponding dataset can be purged in the local
storage and return false retains the local dataset
    return true;
}
```

OnDatasetMerged

Lorsque deux identités qui n'étaient pas connectées sont liées, tous les ensembles de données sont fusionnés. Les applications sont informées de la fusion par le biais du rappel `OnDatasetsMerged`.

```
public bool HandleDatasetMerged(Dataset localDataset, List<string> mergedDatasetNames)
{
    foreach (string name in mergedDatasetNames)
    {
        Dataset mergedDataset = syncManager.OpenOrCreateDataset(name);
        //Lambda function to delete the dataset after fetching it
        EventHandler<SyncSuccessEvent> lambda;
        lambda = (object sender, SyncSuccessEvent e) => {
            ICollection<string> existingValues = localDataset.GetAll().Values;
            ICollection<string> newValues = mergedDataset.GetAll().Values;

            //Implement your merge logic here

            mergedDataset.Delete(); //Delete the dataset locally
            mergedDataset.OnSyncSuccess -= lambda; //We don't want this callback to be
            fired again
            mergedDataset.OnSyncSuccess += (object s2, SyncSuccessEvent e2) => {
                localDataset.Synchronize(); //Continue the sync operation that was
                interrupted by the merge
            };
            mergedDataset.Synchronize(); //Synchronize it as deleted, failing to do so
            will leave us in an inconsistent state
        };
        mergedDataset.OnSyncSuccess += lambda;
        mergedDataset.Synchronize(); //Asnchronously fetch the dataset
    }

    // returning true allows the Synchronize to continue and false stops it
    return false;
}
```

Xamarin

Une fois que vous ouvrez ou créez un ensemble de données, vous pouvez définir différents rappels qui se déclencheront lorsque vous utilisez la méthode de synchronisation. Voici comment enregistrer ces rappels :

```
dataset.OnSyncSuccess += this.HandleSyncSuccess;
dataset.OnSyncFailure += this.HandleSyncFailure;
dataset.OnSyncConflict = this.HandleSyncConflict;
dataset.OnDatasetMerged = this.HandleDatasetMerged;
dataset.OnDatasetDeleted = this.HandleDatasetDeleted;
```

Notez que `SyncSuccess` et `SyncFailure` utilisent `+=` au lieu de `=`. Vous pouvez donc vous abonner à plusieurs rappels.

OnSyncSuccess

Le rappel `OnSyncSuccess` est déclenché quand un ensemble de données est mis à jour avec succès depuis le cloud. Si vous ne définissez pas un rappel, la synchronisation aboutit en mode silencieux.

```
private void HandleSyncSuccess(object sender, SyncSuccessEventArgs e)
{
    // Continue with your game flow, display the loaded data, etc.
}
```

OnSyncFailure

`OnSyncFailure` est appelé si une exception s'est produite lors de la synchronisation. Si vous ne définissez pas un rappel, la synchronisation échoue en mode silencieux.

```
private void HandleSyncFailure(object sender, SyncFailureEventArgs e)
{
    Dataset dataset = sender as Dataset;
    if (dataset.Metadata != null) {
        Console.WriteLine("Sync failed for dataset : " + dataset.Metadata.DatasetName);
    } else {
        Console.WriteLine("Sync failed");
    }
}
```

OnSyncConflict

Des conflits peuvent survenir si la même clé a été modifiée sur le stockage local et dans l'espace de synchronisation. Le rappel `OnSyncConflict` gère la résolution des conflits. Si vous n'implémentez cette méthode, la synchronisation est interrompue quand il y a un conflit.

```
private bool HandleSyncConflict(Dataset dataset, List < SyncConflict > conflicts)
{
    if (dataset.Metadata != null) {
        Console.WriteLine("Sync conflict " + dataset.Metadata.DatasetName);
    } else {
        Console.WriteLine("Sync conflict");
    }
}
```

```
List < Amazon.CognitoSync.SyncManager.Record > resolvedRecords = new List <
Amazon.CognitoSync.SyncManager.Record > ();
foreach(SyncConflict conflictRecord in conflicts) {
    // SyncManager provides the following default conflict resolution methods:
    //     ResolveWithRemoteRecord - overwrites the local with remote records
    //     ResolveWithLocalRecord - overwrites the remote with local records
    //     ResolveWithValue - to implement your own logic
    resolvedRecords.Add(conflictRecord.ResolveWithRemoteRecord());
}
// resolves the conflicts in local storage
dataset.Resolve(resolvedRecords);
// on return true the synchronize operation continues where it left,
//     returning false cancels the synchronize operation
return true;
}
```

OnDatasetDeleted

Lorsqu'un jeu de données est supprimé, le client Amazon Cognito utilise le rappel `OnDatasetDeleted` pour décider si la version locale mise en cache doit également être supprimée. Par défaut, l'ensemble de données n'est pas supprimé.

```
private bool HandleDatasetDeleted(Dataset dataset)
{
    Console.WriteLine(dataset.Metadata.DatasetName + " Dataset has been deleted");
    // Do clean up if necessary
    // returning true informs the corresponding dataset can be purged in the local
    storage and return false retains the local dataset
    return true;
}
```

OnDatasetMerged

Lorsque deux identités qui n'étaient pas connectées sont liées, tous les ensembles de données sont fusionnés. Les applications sont informées de la fusion par le biais du rappel `OnDatasetsMerged`.

```
public bool HandleDatasetMerged(Dataset localDataset, List<string> mergedDatasetNames)
{
    foreach (string name in mergedDatasetNames)
    {
        Dataset mergedDataset = syncManager.OpenOrCreateDataset(name);

        //Implement your merge logic here
    }
}
```

```
mergedDataset.OnSyncSuccess += lambda;  
mergedDataset.SynchronizeAsync(); //Asnchronously fetch the dataset  
}  
  
// returning true allows the Synchronize to continue and false stops it  
return false;  
}
```

Synchronisation en mode Push

⚠ Si vous débutez avec Amazon Cognito Sync, utilisez [AWS AppSync](#). Comme Amazon Cognito Sync, AWS AppSync est un service permettant de synchroniser des données d'application sur différents appareils.

Il permet de synchroniser les données utilisateur telles que des préférences de l'application ou l'état d'un jeu. Il étend également ces capacités en permettant à plusieurs utilisateurs de se synchroniser et de collaborer en temps réel sur des données partagées.

Amazon Cognito suit automatiquement l'association entre l'identité et les appareils. La fonctionnalité de synchronisation en mode push vous permet de vous assurer que chaque instance d'une identité donnée est informée en cas de modification de ses données. Tous les dispositifs associés à cette identité reçoivent une notification push silencieuse chaque fois que ses données changent dans l'espace de synchronisation.

i Note

La synchronisation en mode Push n'est pas prise en charge par JavaScript, Unity ou Xamarin.

Pour pouvoir utiliser la synchronisation en mode Push, vous devez au préalable configurer votre compte en ce sens et activer la synchronisation en mode Push dans la console Amazon Cognito.

Créer une application Amazon Simple Notification Service (Amazon SNS)

Créez et configurez une application Amazon SNS pour les plateformes prises en charge, comme décrit dans le [Guide du développeur SNS](#).

Activer la synchronisation en mode Push via la console Amazon Cognito.

Vous pouvez activer la synchronisation en mode Push via la console Amazon Cognito. A partir de la [page d'accueil de la console](#) :

1. Cliquez sur le nom du groupe d'identités pour lequel vous souhaitez activer la synchronisation en mode Push. La page Dashboard (Tableau de bord) correspondant à votre groupe d'identités s'affiche.
2. Dans l'angle supérieur droit de la page Dashboard (Tableau de bord), cliquez sur Manage Identity Pools (Gérer les groupes d'identités). La page Identités fédérées s'ouvre.
3. Faites défiler l'écran vers le bas et cliquez sur Push synchronization (Synchronisation Push) pour développer cette option.
4. Dans le menu déroulant Service role (Rôle de service), sélectionnez le rôle IAM qui autorise Cognito à envoyer une notification SNS. Cliquez sur Create role (Créer un rôle) afin de créer ou de modifier les rôles associés à votre groupe d'identités dans la [console AWS IAM](#).
5. Sélectionnez une application de plateforme, puis cliquez sur Save Changes (Enregistrer les modifications).
6. Autorisez SNS à accéder à votre application.

Dans la console AWS Identity and Access Management, configurez les rôles IAM pour qu'ils aient un accès Amazon SNS complet ou créez un rôle qui dispose d'un accès SNS complet. L'exemple suivant de politique d'approbation de rôle permet à Amazon Cognito Sync d'assumer un rôle IAM limité. Amazon Cognito Sync ne peut assumer le rôle que lorsqu'il le fait au nom du groupe d'identités dans la condition `aws:SourceArn` et du compte dans la condition `aws:SourceAccount`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cognito-sync.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

```
        "ArnLike": {
            "AWS:SourceArn": "arn:aws:cognito-identity:us-
east-1:123456789012:identitypool/us-east-1:177a950c-2c08-43f0-9983-28727EXAMPLE"
        }
    }
}
```

Pour en savoir plus sur les rôles IAM, consultez la section [Rôles \(délégation et fédération\)](#).

Utilisation de la synchronisation en mode Push dans votre application : Android

Votre application doit importer les services Google Play. Vous pouvez télécharger la dernière version du kit SDK Google Play via le [gestionnaire Android SDK](#). Pour enregistrer votre application et recevoir un ID d'enregistrement de GCM, reportez-vous à la documentation Android relative à [l'implémentation Android](#). Une fois que vous avez l'ID d'enregistrement, enregistrez l'appareil auprès d'Amazon Cognito, comme illustré dans l'extrait de code ci-dessous :

```
String registrationId = "MY_GCM_REGISTRATION_ID";
try {
    client.registerDevice("GCM", registrationId);
} catch (RegistrationFailedException rfe) {
    Log.e(TAG, "Failed to register device for silent sync", rfe);
} catch (AmazonClientException ace) {
    Log.e(TAG, "An unknown error caused registration for silent sync to fail", ace);
}
```

Vous pouvez désormais abonner un dispositif pour recevoir les mises à jour à partir d'un ensemble de données particulier :

```
Dataset trackedDataset = client.openOrCreateDataset("myDataset");
if (client.isDeviceRegistered()) {
    try {
        trackedDataset.subscribe();
    } catch (SubscribeFailedException sfe) {
        Log.e(TAG, "Failed to subscribe to datasets", sfe);
    } catch (AmazonClientException ace) {
        Log.e(TAG, "An unknown error caused the subscription to fail", ace);
    }
}
```



```
}
```

Pour arrêter de recevoir des notifications push à partir d'un ensemble de données, il vous suffit d'appeler la méthode de désinscription. Pour vous abonner à tous les ensembles de données (ou à un sous-ensemble spécifique) dans l'objet `CognitoSyncManager`, utilisez `subscribeAll()` :

```
if (client.isDeviceRegistered()) {
    try {
        client.subscribeAll();
    } catch (SubscribeFailedException sfe) {
        Log.e(TAG, "Failed to subscribe to datasets", sfe);
    } catch (AmazonClientException ace) {
        Log.e(TAG, "An unknown error caused the subscription to fail", ace);
    }
}
```

Dans votre implémentation de l'objet [BroadcastReceiver Android](#), vous pouvez vérifier la dernière version de l'ensemble de données modifié et décider si votre application a besoin de procéder à une nouvelle synchronisation :

```
@Override
public void onReceive(Context context, Intent intent) {

    PushSyncUpdate update = client.getPushSyncUpdate(intent);

    // The update has the source (cognito-sync here), identityId of the
    // user, identityPoolId in question, the non-local sync count of the
    // data set and the name of the dataset. All are accessible through
    // relevant getters.

    String source = update.getSource();
    String identityPoolId = update.getIdentityPoolId();
    String identityId = update.getIdentityId();
    String datasetName = update.getDatasetName();
    long syncCount = update.getSyncCount();

    Dataset dataset = client.openOrCreateDataset(datasetName);

    // need to access last sync count. If sync count is less or equal to
    // last sync count of the dataset, no sync is required.

    long lastSyncCount = dataset.getLastSyncCount();
```

```
    if (lastSyncCount < syncCount) {
        dataset.synchronize(new SyncCallback() {
            // ...
        });
    }
}
```

Les clés suivantes sont disponibles dans la charge utile des notifications push :

- `source`: synchronisation Cognito. Cette clé peut servir de facteur de différenciation entre les notifications.
- `identityPoolId` : ID du groupe d'identités. Cette clé peut être utilisée pour la validation ou pour plus d'informations, bien qu'elle ne soit pas intégrale du point de vue du destinataire.
- `identityId` : ID d'identité dans le groupe.
- `datasetName` : nom de l'ensemble de données qui a été mis à jour. Cette clé est disponible pour l'appel `openOrCreateDataset`.
- `syncCount` : nombre de synchronisations pour l'ensemble de données distant. Vous pouvez utiliser cette clé comme méthode pour vous assurer que l'ensemble de données local est obsolète et que la synchronisation entrante est nouvelle.

Utilisation de la synchronisation en mode Push dans votre application : iOS - Objective-C

Pour obtenir un jeton de dispositif pour votre application, reportez-vous à la documentation Apple sur les demandes de notifications à distance. Une fois que vous recevez le jeton de l'appareil comme objet `NSData` à partir de l'APN, vous devez enregistrer l'appareil auprès d'Amazon Cognito à l'aide de la méthode `registerDevice` : du client de synchronisation, comme ci-dessous :

```
AWSCognito *syncClient = [AWSCognito defaultCognito];
[[syncClient registerDevice: devToken] continueWithBlock:^id(AWSTask *task) {
    if(task.error){
        NSLog(@"Unable to registerDevice: %@", task.error);
    } else {
        NSLog(@"Successfully registered device with id: %@", task.result);
    }
    return nil;
}
```

```
];
```

En mode débogage, le dispositif s'enregistre avec le sandbox APN. En mode release, il s'enregistrera avec des APN. Pour recevoir les mises à jour à partir d'un ensemble de données particulier, utilisez la méthode `subscribe` :

```
[[[syncClient openOrCreateDataset:@"MyDataset"] subscribe]
  continueWithBlock:^id(AWSTask *task) {
    if(task.error){
      NSLog(@"Unable to subscribe to dataset: %@", task.error);
    } else {
      NSLog(@"Successfully subscribed to dataset: %@", task.result);
    }
    return nil;
  }
];
```

Pour arrêter de recevoir des notifications push à partir d'un ensemble de données, il vous suffit d'appeler la méthode `unsubscribe` :

```
[[[syncClient openOrCreateDataset:@"MyDataset"] unsubscribe]
  continueWithBlock:^id(AWSTask *task) {
    if(task.error){
      NSLog(@"Unable to unsubscribe from dataset: %@", task.error);
    } else {
      NSLog(@"Successfully unsubscribed from dataset: %@", task.result);
    }
    return nil;
  }
];
```

Pour vous abonner à tous les ensembles de données dans l'objet `AWSCognito`, appelez `subscribeAll` :

```
[[syncClient subscribeAll] continueWithBlock:^id(AWSTask *task) {
  if(task.error){
    NSLog(@"Unable to subscribe to all datasets: %@", task.error);
  } else {
    NSLog(@"Successfully subscribed to all datasets: %@", task.result);
  }
  return nil;
};
```

```
    }  
];
```

Avant d'appeler `subscribeAll`, veillez à effectuer une synchronisation au moins une fois au niveau de chaque ensemble de données pour qu'ils soient tous sur le serveur.

Pour répondre aux notifications push, vous devez implémenter la méthode `didReceiveRemoteNotification` dans le délégué d'application :

```
- (void)application:(UIApplication *)application didReceiveRemoteNotification:  
(NSDictionary *)userInfo  
{  
    [[NSNotificationCenter defaultCenter]  
    postNotificationName:@"CognitoPushNotification" object:userInfo];  
}
```

Si vous publiez une notification à l'aide d'un gestionnaire de notification, vous pouvez y répondre ailleurs dans l'application où se trouve ce gestionnaire. Si vous vous abonnez à la notification comme ceci...

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(didReceivePushSync:)  
name: :@"CognitoPushNotification" object:nil];
```

... vous pouvez répondre à la notification comme suit :

```
- (void)didReceivePushSync:(NSNotification*)notification  
{  
    NSDictionary * data = [(NSDictionary *)[notification object]  
objectForKey:@"data"];  
    NSString * identityId = [data objectForKey:@"identityId"];  
    NSString * datasetName = [data objectForKey:@"datasetName"];  
    if([self.dataset.name isEqualToString:datasetName] && [self.identityId  
isEqualToString:identityId]){  
        [[self.dataset synchronize] continueWithBlock:^id(AWSTask *task) {  
            if(!task.error){  
                NSLog(@"Successfully synced dataset");  
            }  
            return nil;  
        }];  
    }  
}
```

```
}
```

Les clés suivantes sont disponibles dans la charge utile des notifications push :

- `source`: synchronisation Cognito. Cette clé peut servir de facteur de différenciation entre les notifications.
- `identityPoolId` : ID du groupe d'identités. Cette clé peut être utilisée pour la validation ou pour plus d'informations, bien qu'elle ne soit pas intégrale du point de vue du destinataire.
- `identityId` : ID d'identité dans le groupe.
- `datasetName` : nom de l'ensemble de données qui a été mis à jour. Cette clé est disponible pour l'appel `openOrCreateDataset`.
- `syncCount` : nombre de synchronisations pour l'ensemble de données distant. Vous pouvez utiliser cette clé comme méthode pour vous assurer que l'ensemble de données local est obsolète et que la synchronisation entrante est nouvelle.

Utilisation de la synchronisation en mode Push dans votre application : iOS - Swift

Pour obtenir un jeton de dispositif pour votre application, reportez-vous à la documentation Apple sur les demandes de notifications à distance. Une fois que vous recevez le jeton de l'appareil comme objet `NSData` à partir de l'APN, vous devez enregistrer l'appareil auprès d'Amazon Cognito à l'aide de la méthode `registerDevice`: du client de synchronisation, comme ci-dessous :

```
let syncClient = AWSCognito.default()
syncClient.registerDevice(devToken).continueWith(block: { (task: AWSTask!) ->
    AnyObject! in
    if (task.error != nil) {
        print("Unable to register device: " + task.error.localizedDescription)
    } else {
        print("Successfully registered device with id: \(task.result)")
    }
    return task
})
```

En mode débogage, le dispositif s'enregistre avec le sandbox APN. En mode release, il s'enregistrera avec des APN. Pour recevoir les mises à jour à partir d'un ensemble de données particulier, utilisez la méthode `subscribe` :

```
syncClient.openOrCreateDataset("MyDataset").subscribe().continueWith(block: { (task:
  AWSTask!) -> AnyObject! in
  if (task.error != nil) {
    print("Unable to subscribe to dataset: " + task.error.localizedDescription)

  } else {
    print("Successfully subscribed to dataset: \(task.result)")
  }
  return task
})
```

Pour arrêter de recevoir des notifications push à partir d'un ensemble de données, appelez la méthode `unsubscribe` :

```
syncClient.openOrCreateDataset("MyDataset").unsubscribe().continueWith(block: { (task:
  AWSTask!) -> AnyObject! in
  if (task.error != nil) {
    print("Unable to unsubscribe to dataset: " + task.error.localizedDescription)

  } else {
    print("Successfully unsubscribed to dataset: \(task.result)")
  }
  return task
})
```

Pour vous abonner à tous les ensembles de données dans l'objet `AWSCognito`, appelez `subscribeAll` :

```
syncClient.openOrCreateDataset("MyDataset").subscribeAll().continueWith(block: { (task:
  AWSTask!) -> AnyObject! in
  if (task.error != nil) {
    print("Unable to subscribe to all datasets: " + task.error.localizedDescription)

  } else {
    print("Successfully subscribed to all datasets: \(task.result)")
  }
  return task
})
```

Avant d'appeler `subscribeAll`, veillez à effectuer une synchronisation au moins une fois au niveau de chaque ensemble de données pour qu'ils soient tous sur le serveur.

Pour répondre aux notifications push, vous devez implémenter la méthode `didReceiveRemoteNotification` dans le délégué d'application :

```
func application(application: UIApplication, didReceiveRemoteNotification userInfo:
  [NSObject : AnyObject],
  fetchCompletionHandler completionHandler: (UIBackgroundFetchResult) -> Void) {

  NotificationCenter.defaultCenter().postNotificationName("CognitoPushNotification",
    object: userInfo)
}
```

Si vous publiez une notification à l'aide d'un gestionnaire de notification, vous pouvez y répondre ailleurs dans l'application où se trouve ce gestionnaire. Si vous vous abonnez à la notification comme ceci...

```
NotificationCenter.defaultCenter().addObserver(observer:self,
  selector:"didReceivePushSync:",
  name:"CognitoPushNotification",
  object:nil)
```

... vous pouvez répondre à la notification comme suit :


```
func didReceivePushSync(notification: NSNotification) {
  if let data = (notification.object as! [String: AnyObject])["data"] as? [String:
  AnyObject] {
    let identityId = data["identityId"] as! String
    let datasetName = data["datasetName"] as! String

    if self.dataset.name == datasetName && self.identityId == identityId {
      dataset.synchronize().continueWithBlock {(task) -> AnyObject! in
        if task.error == nil {
          print("Successfully synced dataset")
        }
        return nil
      }
    }
  }
}
```

Les clés suivantes sont disponibles dans la charge utile des notifications push :

- `source`: synchronisation Cognito. Cette clé peut servir de facteur de différenciation entre les notifications.
- `identityPoolId` : ID du groupe d'identités. Cette clé peut être utilisée pour la validation ou pour plus d'informations, bien qu'elle ne soit pas intégrale du point de vue du destinataire.
- `identityId` : ID d'identité dans le groupe.
- `datasetName` : nom de l'ensemble de données qui a été mis à jour. Cette clé est disponible pour l'appel `openOrCreateDataset`.
- `syncCount` : nombre de synchronisations pour l'ensemble de données distant. Vous pouvez utiliser cette clé comme méthode pour vous assurer que l'ensemble de données local est obsolète et que la synchronisation entrante est nouvelle.

Flux Amazon Cognito

 Si vous débutez avec Amazon Cognito Sync, utilisez [AWS AppSync](#). Comme Amazon Cognito Sync, AWS AppSync est un service permettant de synchroniser des données d'application sur différents appareils.

Il permet de synchroniser les données utilisateur telles que des préférences de l'application ou l'état d'un jeu. Il étend également ces capacités en permettant à plusieurs utilisateurs de se synchroniser et de collaborer en temps réel sur des données partagées.

Les flux Amazon Cognito permettent aux développeurs de voir et de contrôler les données stockées dans Amazon Cognito. Les développeurs peuvent désormais configurer un flux Kinesis en vue de recevoir des événements lorsque les données sont mises à jour et synchronisées. Amazon Cognito peut transmettre en temps réel une modification de jeu de données à un flux Kinesis que vous possédez.

Les flux Amazon Cognito vous permettent de déplacer vers Kinesis toutes vos données de synchronisation, qui peuvent ensuite être envoyées à un outil d'entrepôt de données comme Amazon Redshift à des fins d'analyse complémentaire. Pour en savoir plus sur Kinesis, consultez [Mise en route avec Amazon Kinesis](#).

Configuration des flux

Vous pouvez configurer les flux Amazon Cognito dans la console Amazon Cognito. Pour activer les flux Amazon Cognito dans la console Amazon Cognito, vous devez sélectionner le flux Kinesis

dans lequel effectuer la publication, ainsi qu'un rôle IAM qui autorise Amazon Cognito à placer les événements dans le flux sélectionné.

A partir de la [page d'accueil de la console](#) :

1. Cliquez sur le nom du groupe d'identités pour lequel vous souhaitez configurer des flux Amazon Cognito. La page Dashboard (Tableau de bord) correspondant à votre groupe d'identités s'affiche.
2. Dans l'angle supérieur droit de la page Dashboard (Tableau de bord), cliquez sur Manage Identity Pools (Gérer les groupes d'identités). La page Gérer les identités fédérées s'ouvre.
3. Faites défiler l'écran vers le bas et cliquez sur Cognito Streams (Flux Cognito) pour développer cette option.
4. Dans le menu déroulant Nom de flux, sélectionnez le nom d'un flux Kinesis existant. Vous pouvez également cliquer sur Créer un flux pour en créer un, en saisissant un nom de flux et le nombre de partitions. Pour en savoir sur les partitions et déterminer le nombre de partitions requis pour votre flux, consultez le [Manuel du développeur Kinesis](#).
5. Dans le menu déroulant Publish role (Publier un rôle), sélectionnez le rôle IAM qui autorise Amazon Cognito à publier votre flux. Cliquez sur Create role (Créer un rôle) afin de créer ou de modifier les rôles associés à votre groupe d'identités dans la [console AWS IAM](#).
6. Dans le menu déroulant État du flux, sélectionnez Activé pour activer les mises à jour du flux. Cliquez sur Save Changes (Enregistrer les modifications).

Une fois que vous avez configuré les flux Amazon Cognito, toutes les mises à jour ultérieures apportées aux jeux de données de ce groupe d'identités sont envoyées au flux Kinesis.

Contenu du flux

Chaque enregistrement envoyé au flux représente une seule synchronisation. Voici un exemple d'enregistrement envoyé au flux :

```
{
  "identityPoolId": "Pool Id",
  "identityId": "Identity Id",
  "dataSetName": "Dataset Name",
  "operation": "(replace|remove)",
  "kinesisSyncRecords": [
    {
      "key": "Key",
      "value": "Value",
```

```
        "syncCount": 1,
        "lastModifiedDate": 1424801824343,
        "deviceLastModifiedDate": 1424801824343,
        "op": "(replace|remove)"
    },
    ...
],
"lastModifiedDate": 1424801824343,
"kinesisSyncRecordsURL": "S3Url",
"payloadType": "(S3Url|Inline)",
"syncCount": 1
}
```

Pour les mises à jour qui dépassent la charge utile maximale de Kinesis de 1 Mo, Amazon Cognito inclut une URL Amazon S3 pré-signée comprenant le contenu total de la mise à jour.

Une fois que vous avez configuré les flux Amazon Cognito, si vous supprimez le flux Kinesis ou modifiez l'autorisation d'approbation de rôle pour qu'Amazon Cognito Sync ne puisse plus endosser ce rôle, vous désactivez les flux Amazon Cognito. Vous devez recréer le flux Kinesis ou corriger le rôle, puis réactiver le flux.


Publication en bloc

Une fois que vous avez configuré les flux Amazon Cognito, vous pouvez exécuter une publication en bloc des données existantes du groupe d'identités. Une fois que vous lancez ce type d'opération via la console ou directement via l'API, Amazon Cognito commence à publier ces données dans le même flux que celui qui reçoit les mises à jour.

Amazon Cognito ne garantit pas l'unicité des données envoyées au flux lors de l'opération de publication en bloc. Vous pouvez recevoir la même mise à jour en tant que mise à jour, ainsi que dans le cadre d'une publication en bloc. Gardez cette information à l'esprit lorsque vous traiterez les enregistrements de votre flux.

Pour publier en bloc tous vos flux, suivez les étapes 1 à 6 décrites dans la section Configuration des flux, puis cliquez sur Start bulk publish. Vous êtes limité à une publication en bloc à tout moment et à une demande de publication en bloc réussie toutes les 24 heures.

Événements Amazon Cognito

 Si vous débutez avec Amazon Cognito Sync, utilisez [AWS AppSync](#). Comme Amazon Cognito Sync, AWS AppSync est un service permettant de synchroniser des données d'application sur différents appareils.

Il permet de synchroniser les données utilisateur telles que des préférences de l'application ou l'état d'un jeu. Il étend également ces capacités en permettant à plusieurs utilisateurs de se synchroniser et de collaborer en temps réel sur des données partagées.

Les événements Amazon Cognito vous permettent d'exécuter une fonction AWS Lambda en réponse à des événements importants dans Amazon Cognito. Amazon Cognito déclenche l'événement Sync Trigger lors de la synchronisation d'un jeu de données. Vous pouvez utiliser cet événement pour effectuer une action lorsqu'un utilisateur met à jour des données. Cette fonction peut évaluer et éventuellement manipuler les données avant d'être stockées dans le cloud et synchronisées avec les autres dispositifs de l'utilisateur. Elle est particulièrement utile pour valider des données provenant d'un dispositif avant qu'elles ne soient synchronisées avec les autres appareils de l'utilisateur, ou pour mettre à jour d'autres valeurs dans l'ensemble de données en fonction des données entrantes (comme l'attribution d'un prix quand un joueur atteint un nouveau niveau).

Les étapes ci-dessous illustrent comment configurer une fonction Lambda qui s'exécute chaque fois qu'un jeu de données Amazon Cognito est synchronisé.

Note

Lorsque vous utilisez des événements Amazon Cognito, vous pouvez uniquement utiliser les informations d'identification obtenues d'Amazon Cognito Identity. Si vous avez une fonction Lambda associée, mais appelez `UpdateRecords` avec des informations d'identification de compte AWS (informations d'identification de développeur), votre fonction Lambda n'est pas appelée.

Création d'une fonction dans AWS Lambda

Pour intégrer Lambda avec Amazon Cognito, vous devez commencer par créer une fonction dans Lambda. Pour ce faire :

Sélection de la fonction Lambda dans Amazon Cognito

1. Ouvrez la console Lambda.
2. Cliquez sur Créer une fonction Lambda.
3. Sur l'écran Select blueprint, recherchez « cognito-sync-trigger » et sélectionnez cette option.
4. Sur l'écran Configure event sources, laissez le type de source d'événement sur « Cognito Sync Trigger » dans Event source type, puis sélectionnez votre groupe d'identités. Cliquez sur Next (Suivant).

Note

Lorsque vous configurez un déclencheur Amazon Cognito Sync en dehors de la console, vous devez ajouter des autorisations basées sur les ressources Lambda pour autoriser Amazon Cognito à appeler la fonction. Vous pouvez ajouter cette autorisation à partir de la console Lambda (voir [Utilisation de politiques basées sur les ressources pour AWS Lambda](#)) ou en utilisant l'opération Lambda [AddPermission](#).

Exemple de politique basée sur les ressources Lambda

La politique AWS Lambda suivante basée sur les ressources permet à Amazon Cognito d'appeler d'une manière limitée une fonction Lambda. Amazon Cognito ne peut appeler la fonction que pour le compte du groupe d'identités dans la condition `aws:SourceArn` et du compte dans la condition `aws:SourceAccount`.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "lambda-allow-cognito-my-function",
      "Effect": "Allow",
      "Principal": {
        "Service": "cognito-sync.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "<your Lambda function ARN>",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "<your account number>"
        },
        "ArnLike": {
          "AWS:SourceArn": "<your identity pool ARN>"
        }
      }
    }
  ]
}
```

```
}  
  }  
} ]  
}
```

5. Sur l'écran Configurer la fonction, saisissez un nom et une description pour votre fonction. Laissez « Node.js » pour Runtime. Ne modifiez pas le code pour notre exemple. L'exemple par défaut ne modifie pas les données qui sont en cours de synchronisation. Il enregistre uniquement le fait que l'événement Sync Trigger d'Amazon Cognito a eu lieu. Conservez le nom de gestionnaire « index.handler ». Pour l'option Rôle, sélectionnez un rôle IAM qui autorise votre code à accéder à AWS Lambda. Pour modifier les rôles, consultez la console IAM. Ne modifiez pas les paramètres avancés. Cliquez sur Next (Suivant).
6. Sur l'écran Révision, passez en revue les détails et cliquez sur Create fonction. La page suivante affiche votre nouvelle fonction Lambda.

Maintenant qu'une fonction appropriée est écrite dans Lambda, vous devez choisir cette fonction en tant que gestionnaire de l'événement Sync Trigger d'Amazon Cognito. Les étapes indiquées ci-dessous vous guident tout au long de ce processus.

A partir de la page d'accueil de la console :

1. Cliquez sur le nom du groupe d'identités pour lequel vous souhaitez configurer des événements Amazon Cognito. La page Dashboard (Tableau de bord) correspondant à votre groupe d'identités s'affiche.
2. Dans l'angle supérieur droit de la page Tableau de bord, cliquez sur Gérer les identités fédérées. La page Gérer les identités fédérées s'ouvre.
3. Faites défiler l'écran vers le bas et cliquez sur Cognito Events pour développer cette option.
4. Dans le menu déroulant Sync Trigger, sélectionnez la fonction Lambda que vous souhaitez déclencher lorsqu'un événement de synchronisation se produit.
5. Cliquez sur Save Changes (Enregistrer les modifications).

Désormais, la fonction Lambda sera exécutée à chaque synchronisation d'un jeu de données. La section suivante explique comment lire et modifier les données dans votre fonction lorsqu'elles sont en cours de synchronisation.

Écriture d'une fonction Lambda pour des déclencheurs de synchronisation

Les déclencheurs de synchronisation suivent le modèle de programmation suivi par les interfaces du fournisseur de services. Amazon Cognito fournit une entrée à votre fonction Lambda au format JSON suivant.

```
{
  "version": 2,
  "eventType": "SyncTrigger",
  "region": "us-east-1",
  "identityPoolId": "identityPoolId",
  "identityId": "identityId",
  "datasetName": "datasetName",
  "datasetRecords": {
    "SampleKey1": {
      "oldValue": "oldValue1",
      "newValue": "newValue1",
      "op": "replace"
    },
    "SampleKey2": {
      "oldValue": "oldValue2",
      "newValue": "newValue2",
      "op": "replace"
    },
    ...
  }
}
```

Amazon Cognito attend la valeur de retour de la fonction au même format que l'entrée.

Lorsque vous écrivez des fonctions pour l'événement de déclencheur de synchronisation, observez les points suivants :

- Quand Amazon Cognito appelle votre fonction Lambda au cours d'une opération `UpdateRecords`, la fonction doit répondre dans un délai de 5 secondes. Sinon, le service Amazon Cognito Sync génère une exception `LambdaSocketTimeoutException`. Vous ne pouvez pas augmenter ce délai d'attente.
- Si vous obtenez une exception `LambdaThrottledException`, réessayez l'opération de synchronisation pour mettre à jour les enregistrements.
- Amazon Cognito fournit tous les enregistrements présents dans le jeu de données en tant qu'entrée pour la fonction.
- Les enregistrements mis à jour par l'utilisateur de l'application ont le champ `op` défini sur `replace`. Les enregistrements supprimés ont le champ `op` défini sur `remove`.

- Vous pouvez modifier n'importe quel enregistrement, même si l'utilisateur de l'application ne met pas à jour l'enregistrement.
- Tous les champs sauf `datasetRecords` sont en lecture seule. Ne les modifiez pas. Si vous modifiez ces champs, vous ne pouvez pas mettre à jour les enregistrements.
- Pour modifier la valeur d'un enregistrement, mettez-la à jour et définissez le champ `op` sur `replace`.
- Pour supprimer un enregistrement, définissez le champ `op` sur `remove`, ou définissez la valeur sur `null`.
- Pour ajouter un enregistrement, ajoutez un nouvel enregistrement dans le tableau `datasetRecords`.
- Amazon Cognito ignore tout enregistrement omis dans la réponse quand Amazon Cognito met à jour l'enregistrement.

Exemple de fonction Lambda

L'exemple de fonction Lambda suivant montre comment accéder aux données, les modifier et les supprimer.

```
console.log('Loading function');

exports.handler = function(event, context) {
    console.log(JSON.stringify(event, null, 2));

    //Check for the event type
    if (event.eventType === 'SyncTrigger') {

        //Modify value for a key
        if('SampleKey1' in event.datasetRecords){
            event.datasetRecords.SampleKey1.newValue = 'ModifyValue1';
            event.datasetRecords.SampleKey1.op = 'replace';
        }

        //Remove a key
        if('SampleKey2' in event.datasetRecords){
            event.datasetRecords.SampleKey2.op = 'remove';
        }

        //Add a key
        if(!('SampleKey3' in event.datasetRecords)){
            event.datasetRecords.SampleKey3={'newValue':'ModifyValue3', 'op' :
            'replace'};
        }
    }
}
```

```
    }  
  }  
  context.done(null, event);  
};
```


Utilisation de la console Amazon Cognito

La [console Amazon Cognito](#) vous permet de créer et gérer les groupes d'utilisateurs et d'identités.

Ce guide fournit des instructions step-by-step détaillées sur les tâches courantes du groupe d'utilisateurs Amazon Cognito dans la console Amazon Cognito.

Pour utiliser la console Amazon Cognito

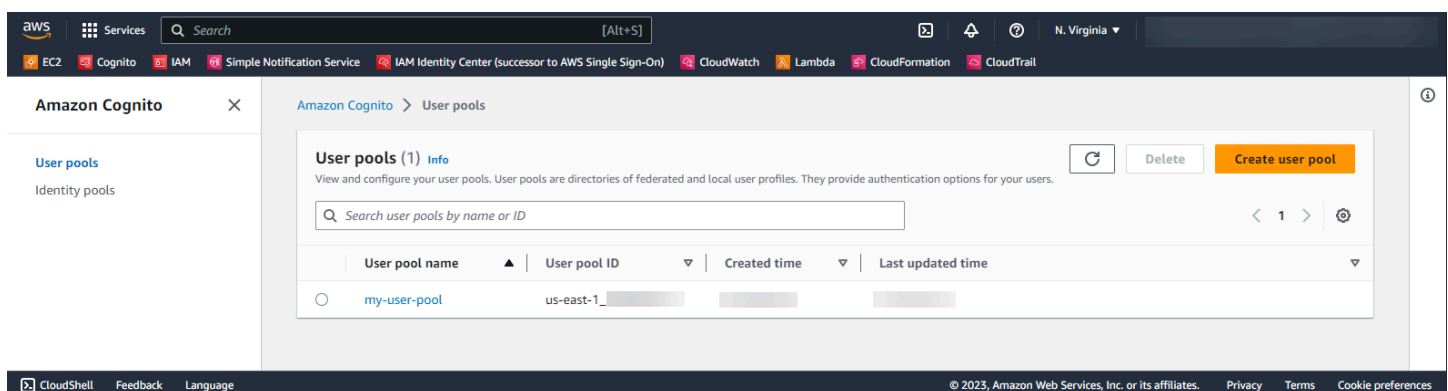
1. Pour utiliser Amazon Cognito, vous [devez créer un AWS compte](#).
2. Accédez à la [console Amazon Cognito](#). Il se peut que vous soyez invité à saisir vos AWS informations d'identification.
3. Pour créer ou modifier un groupe d'utilisateurs, choisissez Groupes d'utilisateurs dans le panneau de navigation de gauche.

Pour plus d'informations, consultez [Démarrage avec les groupes d'utilisateurs](#).

4. Pour créer ou modifier une réserve d'identités, choisissez Groupes d'identités. Vous serez dirigé vers la console d'origine des groupes d'identités Amazon Cognito.

Pour plus d'informations, consultez [Commencer à utiliser les pools d'identités Amazon Cognito](#).

La console Amazon Cognito fait partie de AWS Management Console, qui fournit des informations sur votre compte et votre facturation. Pour plus d'informations, consultez [Utilisation de AWS Management Console](#).



Rubriques

- [Console des groupes d'utilisateurs](#)
- [Console des réserves d'identités](#)

Console des groupes d'utilisateurs

Dans la vue Groupes d'utilisateurs de la console Amazon Cognito, choisissez un groupe d'utilisateurs dans la liste pour en afficher les détails. Dans la vue détaillée, la Présentation du groupe d'utilisateurs en haut de la console contient des informations de base sur votre groupe d'utilisateurs. Les onglets suivants organisent la configuration de votre groupe d'utilisateurs en fonctions connexes.

Users

L'onglet Utilisateurs contient des informations sur les utilisateurs et les importations d'utilisateurs à partir de fichiers CSV. Vous pouvez ajouter, supprimer et modifier des utilisateurs dans cet onglet.

Références

- [Gestion des utilisateurs dans votre groupe d'utilisateurs](#)
- [Importation d'utilisateurs dans des groupes d'utilisateurs depuis un fichier CSV](#)

Groups

L'onglet Groupes contient des informations sur les groupes d'utilisateurs. Vous pouvez ajouter, modifier et remplacer l'appartenance à des groupes, ainsi que modifier les rôles IAM associés aux groupes pour l'intégration de la réserve d'identités.

Références

- [Ajout de groupes à un groupe d'utilisateurs](#)

Expérience de connexion

L'onglet Expérience de connexion contient des informations sur la manière dont les utilisateurs se connectent à votre groupe d'utilisateurs. Cet onglet inclut les fournisseurs d'identité tiers, les options de nom d'utilisateur, la politique de mot de passe, la configuration de l'authentification multifactorielle (MFA), le comportement de mot de passe oublié et la fonction de mémorisation de l'appareil. Vous pouvez ajouter et modifier des fournisseurs d'identité, et modifier le comportement de connexion global de votre groupe d'utilisateurs.

Références

- [Ajout de la connexion du groupe d'utilisateurs via un tiers](#)
- [Personnalisation des attributs de connexion](#)
- [Ajout d'exigences de mot de passe pour un groupe d'utilisateurs](#)

- [Ajout de l'authentification MFA à un groupe d'utilisateurs](#)
- [Récupération de comptes d'utilisateurs](#)
- [Utilisation d'appareils utilisateur dans votre groupe d'utilisateurs](#)

Expérience d'inscription

L'onglet Expérience d'inscription contient des informations sur l'inscription en libre-service, les attributs requis, la vérification des numéros de téléphone et des adresses e-mail, ainsi que des attributs personnalisés.

Références

- [Inscription et confirmation des comptes d'utilisateur](#)
- [Attributs de groupe d'utilisateurs](#)
- [Vérification des coordonnées à l'inscription](#)

Messagerie

L'onglet Messagerie contient des informations sur les Services AWS que vous souhaitez utiliser pour envoyer des e-mails et des SMS à vos utilisateurs, ainsi que le format des messages que vous souhaitez leur envoyer.

Références

- [Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito](#)
- [Paramètres des SMS pour les groupes d'utilisateurs Amazon Cognito](#)
- [Configuration des messages SMS, e-mails de vérification et messages d'invitation des utilisateurs](#)

Intégration d'application

L'onglet Intégration d'application contient des informations sur les clients d'application du groupe d'utilisateurs, le domaine que vous attribuez aux points de terminaison de service de votre groupe d'utilisateurs, les serveurs de ressources d'API, l'interface utilisateur hébergée et la sécurité avancée. Vous pouvez explorer en détail chaque client d'application pour configurer les éléments suivants.

1. Paramètres de jeton
2. Adresses URL de rappel
3. Flux d'authentification
4. Autorisations d'attribut

5. Sécurité avancée et paramètres d'interface utilisateur hébergés spécifiques à l'application
6. Analytique Amazon Pinpoint

Références

- [Clients de l'application de groupe d'utilisateurs](#)
- [Configuration et utilisation de l'interface utilisateur hébergée par Amazon Cognito et des points de terminaison de fédération](#)
- [Configuration d'un domaine de groupe d'utilisateurs](#)
- [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#)
- [Ajout de la sécurité avancée à un groupe d'utilisateurs](#)
- [Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito.](#)

Propriétés du groupe d'utilisateurs

L'onglet Propriétés du groupe d'utilisateurs contient des informations sur la configuration du groupe d'utilisateurs qui ne sont pas directement liées aux utilisateurs : déclencheurs Lambda, protection ACL AWS WAF Web, protection contre les suppressions et balises de ressources.

Références

- [Personnalisation des flux de travail de groupe d'utilisateurs avec des déclencheurs Lambda](#)
- [Associer une ACL AWS WAF Web à un groupe d'utilisateurs](#)
- [Protection contre la suppression du groupe d'utilisateurs](#)
- [Marquer vos ressources AWS](#)

Console des réserves d'identités

Dans la vue Groupes d'identités de la console Amazon Cognito, choisissez une réserve d'identités dans la liste pour en afficher les détails. Dans la vue détaillée, l'Aperçu du groupe d'identités en haut de la console contient des informations de base sur votre groupe d'utilisateurs. Les onglets suivants organisent la configuration de votre groupe d'utilisateurs en fonctions connexes.

Statistiques utilisateur

L'onglet Statistiques utilisateur affiche des informations statistiques sur les utilisateurs qui ont généré des identités dans votre réserve d'identités. Vous ne pouvez configurer aucun paramètre de réserve d'identités dans cet onglet.

Navigateur d'identité

L'onglet Navigateur d'identité contient des informations sur les identités individuelles que les utilisateurs ont générées dans votre réserve d'identités. Vous pouvez consulter et supprimer des identités.

Références

- [Commencer à utiliser les pools d'identités Amazon Cognito](#)

Accès utilisateur

L'onglet Accès utilisateur contient des informations sur les fournisseurs d'identité que vous avez liés à votre réserve d'identités, les fournisseurs de développeurs, les rôles IAM par défaut attribués aux identités, et la configuration d'accès invités non authentifiés. Vous pouvez explorer en détail chaque fournisseur d'identité pour configurer les éléments suivants.

1. Contrôle d'accès basé sur les rôles avec Sélection du rôle IAM
2. Contrôle d'accès basé sur les attributs avec Attributs de contrôle d'accès

Références

- [Fournisseurs d'identité externes aux groupes d'identités](#)
- [Rôles IAM](#)
- [Identités authentifiées et non authentifiées](#)
- [Identités authentifiées par le développeur \(réserves d'identités\)](#)
- [Utilisation du contrôle d'accès basé sur les rôles](#)
- [Utilisation d'attributs pour le contrôle d'accès](#)

Propriétés du groupe d'identités

L'onglet Propriétés du groupe d'identités contient des informations sur les diverses configurations de réserve d'identités : authentification de base (classique) et balises de ressources.

- [Flux d'authentification de groupes d'identités \(identités fédérées\)](#)
- [Marquer vos ressources AWS](#)

Sécurité dans Amazon Cognito

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture 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 ceci 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 à Amazon Cognito, consultez la section [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 la sensibilité de vos données, les exigences de votre entreprise et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Amazon Cognito. Elle montre comment configurer Amazon Cognito pour atteindre vos objectifs en matière de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources Amazon Cognito.

Table des matières

- [Protection des données dans Amazon Cognito](#)
- [Gestion des identités et des accès pour Amazon Cognito](#)
- [Journalisation et surveillance dans Amazon Cognito](#)
- [Validation de conformité pour Amazon Cognito](#)
- [Résilience dans Amazon Cognito](#)
- [Sécurité de l'infrastructure dans Amazon Cognito](#)
- [Configuration et analyse des vulnérabilités dans les groupes d'utilisateurs Amazon Cognito](#)
- [AWS politiques gérées pour Amazon Cognito](#)

Protection des données dans Amazon Cognito

Le modèle de [responsabilité AWS partagée Le modèle](#) s'applique à la protection des données dans Amazon Cognito (Amazon Cognito). Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure mondiale qui gère l'ensemble du AWS cloud. 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 les AWS services que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez les [FAQ sur la confidentialité des données](#).

Pour des raisons de protection des données, nous vous recommandons de protéger les informations d'identification des AWS comptes et de configurer des comptes utilisateur individuels avec AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS
- 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 au sein AWS des services.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données personnelles stockées dans Amazon S3.

Nous vous recommandons vivement de ne jamais placer d'informations identifiables sensibles, telles que les numéros de compte de vos clients, dans des champs de formulaire comme Nom. Cela inclut lorsque vous travaillez avec Amazon Cognito ou d'autres AWS services à l'aide de la console, de l'API ou AWS des AWS CLI SDK. Toutes les données que vous saisissez dans Amazon Cognito ou d'autres services peuvent être récupérées, afin d'être insérées dans des journaux de diagnostic. Lorsque vous fournissez une URL à un serveur externe, n'incluez pas les informations d'identification non chiffrées dans l'URL pour valider votre demande adressée au serveur.

Chiffrement des données

Le chiffrement des données se divise généralement en deux catégories : le chiffrement au repos et le chiffrement en transit.

Chiffrement au repos

Les données au sein d'Amazon Cognito sont chiffrées au repos conformément aux normes du secteur.

Chiffrement en transit

En tant que service géré, Amazon Cognito est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Amazon Cognito via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Les groupes d'utilisateurs et les réserves d'identités Amazon Cognito utilisent des opérations d'API authentifiées par IAM, non authentifiées et autorisées par des jetons. Les opérations d'API non authentifiées et autorisées par des jetons sont destinées à être utilisées par vos clients, les utilisateurs finaux de votre application. Les opérations d'API non authentifiées et autorisées par des jetons sont chiffrées au repos et en transit. Pour plus d'informations, consultez [Opérations d'API authentifiées et non authentifiées des groupes d'utilisateurs Amazon Cognito](#).

Note

Amazon Cognito chiffre le votre contenu en interne et ne prend pas en charge les clés fournies par le client.

Gestion des identités et des accès pour Amazon Cognito

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Des administrateurs IAM contrôlent les personnes qui peuvent être authentifiées (connectées) et autorisées (dotées d'autorisations) à utiliser des ressources Amazon Cognito. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion des accès à l'aide de politiques](#)
- [Fonctionnement d'Amazon Cognito avec IAM](#)
- [Exemples de politiques basées sur une identité pour Amazon Cognito](#)
- [Résolution de problèmes d'identité et d'accès dans Amazon Cognito](#)
- [Utilisation de rôles liés à un service pour Amazon Cognito](#)

Public ciblé

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

Utilisateur du service – Si vous utilisez le service Amazon Cognito pour accomplir votre tâche, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Plus vous utilisez de fonctions Amazon Cognito pour accomplir votre travail, plus vous risquez d'avoir besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans Amazon Cognito, veuillez consulter [Résolution de problèmes d'identité et d'accès dans Amazon Cognito](#).

Administrateur du service – Si vous êtes le responsable des ressources Amazon Cognito dans votre entreprise, vous bénéficiez probablement d'un accès total à Amazon Cognito. C'est à vous de déterminer les fonctions et les ressources Amazon Cognito auxquelles vos utilisateurs des services

pourront accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec Amazon Cognito, consultez [Fonctionnement d'Amazon Cognito avec IAM](#).

Administrateur IAM – Si vous êtes un administrateur IAM, vous souhaitez probablement en savoir plus sur la façon d'écrire des politiques pour gérer l'accès à Amazon Cognito. Pour voir des exemples de politiques basées sur une identité pour Amazon Cognito que vous pouvez utiliser dans IAM, consultez [Exemples de politiques basées sur une identité pour Amazon Cognito](#).

Authentification par des identités

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

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

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

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

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

plus, consultez [Authentification multifactorielle](#) dans le Guide de l'utilisateur AWS IAM Identity Center et [Utilisation de l'authentification multifactorielle \(MFA\) dans l'interface AWS](#) dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

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

Identité fédérée

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

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

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

Utilisateurs et groupes IAM

Un [utilisateur IAM](#) est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous

vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification](#) dans le Guide de l'utilisateur IAM.

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

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

Rôles IAM

Un [rôle IAM](#) est une identité au sein de votre Compte AWS 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 rôle IAM 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 [Utilisation de rôles IAM](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- 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 obtenir des informations sur les rôles pour la fédération, consultez [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (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 en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez [Différence entre les rôles IAM et les politiques basées sur les ressources](#) dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
 - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez [Sessions de transmission d'accès](#).
 - Rôle de service : il s'agit d'un [rôle IAM](#) attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.
 - Rôle lié à un service — 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 administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS 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 [Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2](#) dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez [Quand créer un rôle IAM \(au lieu d'un utilisateur\)](#) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

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

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

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

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

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces

politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

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

politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de 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. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

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

Listes de contrôle d'accès (ACL)

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

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

Autres types de politique

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

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

Fonctionnement d'Amazon Cognito avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon Cognito, découvrez les fonctionnalités IAM que vous pouvez utiliser avec Amazon Cognito.

Fonctions IAM que vous pouvez utiliser avec Amazon Cognito

Fonction IAM	Assistance Amazon Cognito
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	Oui
ACL	Non
ABAC (identifications dans les politiques)	Partielle
Informations d'identification temporaires	Oui
Autorisations de principaux	Non
Fonctions de service	Oui
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble de la façon dont Amazon Cognito et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les [AWS services compatibles avec IAM dans le guide de l'utilisateur IAM](#).

Politiques basées sur l'identité pour Amazon Cognito

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 découvrir comment créer une politique basée sur l'identité, consultez [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur une identité pour Amazon Cognito

Pour voir des exemples de politiques basées sur une identité pour Amazon Cognito, consultez [Exemples de politiques basées sur une identité pour Amazon Cognito](#).

Politiques basées sur une ressource dans Amazon Cognito

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. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de 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. 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 permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez [Différence entre les rôles IAM et les politiques basées sur une ressource](#) dans le Guide de l'utilisateur IAM.

Actions de politique pour Amazon Cognito

Prend en charge les actions de politique Oui

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

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

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

Pour afficher la liste des actions Amazon Cognito, consultez [Actions définies par Amazon Cognito](#) dans Référence de l'autorisation de service.

Les actions de politique dans Amazon Cognito utilisent le préfixe suivant avant l'action :

```
cognito-identity
```

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

```
"Action": [  
  "cognito-identity:action1",  
  "cognito-identity:action2"  
]
```

API signées et non signées

Lorsque vous signez des demandes d'API Amazon Cognito avec des AWS informations d'identification, vous pouvez les restreindre dans une politique AWS Identity and Access Management (IAM). Les demandes d'API avec lesquelles vous devez signer les informations d'identification AWS incluent la connexion côté serveur avec `AdminInitiateAuth`, ainsi que des actions permettant de créer, d'afficher ou de modifier vos ressources Amazon Cognito, telles que `UpdateUserPool`. Pour plus d'informations sur les demandes d'API signées, consultez [la section Signature des demandes AWS d'API](#).

Amazon Cognito étant un produit d'identité client destiné aux applications que vous souhaitez mettre à la disposition du public, vous avez accès aux API non signées suivantes. Votre application envoie ces requêtes d'API à vos utilisateurs et à vos utilisateurs potentiels. Certaines API ne nécessitent aucune autorisation préalable, comme `InitiateAuth` pour démarrer une nouvelle session d'authentification. Certaines API utilisent des jetons d'accès ou des clés de session pour l'autorisation, comme `VerifySoftwareToken` pour terminer la configuration MFA pour un utilisateur disposant déjà d'une session authentifiée. Une API des groupes d'utilisateurs Amazon Cognito autorisée et non signée prend en charge un paramètre `Session` ou `AccessToken` dans la syntaxe de la requête, telle qu'elle est affichée dans [Référence d'API Amazon Cognito](#). Une API Amazon Cognito Identity non signée prend en charge un paramètre `IdentityId` tel qu'affiché dans la [Référence d'API des identités fédérées d'Amazon Cognito](#).

Pour plus d'informations sur les rôles et modèles d'autorisation des opérations d'API des groupes d'utilisateurs Amazon Cognito, consultez [Opérations d'API authentifiées et non authentifiées des groupes d'utilisateurs Amazon Cognito](#).

Opérations d'API des réserves d'identités Amazon Cognito

- `GetId`
- `GetOpenIdToken`
- `GetCredentialsForIdentity`

- `UnlinkIdentity`

Opérations d'API des groupes d'utilisateurs Amazon Cognito

- `AssociateSoftwareToken`
- `ChangePassword`
- `ConfirmDevice`
- `ConfirmForgotPassword`
- `ConfirmSignUp`
- `DeleteUser`
- `DeleteUserAttributes`
- `ForgetDevice`
- `ForgotPassword`
- `GetDevice`
- `GetUser`
- `GetUserAttributeVerificationCode`
- `GlobalSignOut`
- `InitiateAuth`
- `ListDevices`
- `ResendConfirmationCode`
- `RespondToAuthChallenge`
- `RevokeToken`
- `SetUserMFAPreference`
- `SetUserSettings`
- `SignUp`
- `UpdateAuthEventFeedback`
- `UpdateDeviceStatus`
- `UpdateUserAttributes`
- `VerifySoftwareToken`
- `VerifyUserAttribute`

Pour voir des exemples de politiques basées sur une identité pour Amazon Cognito, consultez [Exemples de politiques basées sur une identité pour Amazon Cognito](#).

Ressources de politique pour Amazon Cognito

Prend en charge les ressources de politique Oui

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

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

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

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

Amazon Resource Names (ARN)

ARN pour les identités fédérées Amazon Cognito

Dans les groupes d'identités (identités fédérées) Amazon Cognito, il est possible de limiter l'accès d'un utilisateur IAM à un groupe d'identités spécifiques, en utilisant le format Amazon Resource Name (ARN), comme dans l'exemple suivant. Pour plus d'informations sur les ARN, consultez [Identifiants IAM](#).

```
arn:aws:cognito-identity:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID
```

ARN pour Amazon Cognito Sync

Dans Amazon Cognito Sync, les clients peuvent également restreindre l'accès en fonction de l'ID de groupe d'identités, de l'ID d'identité et du nom de jeu de données.

Pour les API qui opèrent sur un groupe d'identités, le format d'ARN de groupe d'identités est le même que pour les identités fédérées Amazon Cognito, sauf que le nom du service est cognito-sync au lieu de cognito-identity :

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID
```

Pour les API qui fonctionnent au niveau d'une seule identité, par exemple, RegisterDevice, vous pouvez faire référence à l'identité individuelle avec le format d'ARN suivant :

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID/  
identity/IDENTITY_ID
```

Pour les API qui exploitent des ensembles de données (par exemple, UpdateRecords et ListRecords), vous pouvez faire référence à l'ensemble de données avec le format d'ARN suivant :

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID/  
identity/IDENTITY_ID/dataset/DATASET_NAME
```

ARN pour les groupes d'utilisateurs d'Amazon Cognito

Pour Amazon Cognito Your User Pools, il est possible de limiter l'accès d'un utilisateur à un groupe d'utilisateurs spécifique, en utilisant le format d'ARN suivant :

```
arn:aws:cognito-idp:REGION:ACCOUNT_ID:userpool/USER_POOL_ID
```

Pour afficher la liste des types de ressources Amazon Cognito, consultez [Ressources définies par Amazon Cognito](#) dans la Référence de l'autorisation de service. Pour savoir les actions avec lesquelles vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par Amazon Cognito](#).

Pour voir des exemples de politiques basées sur une identité pour Amazon Cognito, consultez [Exemples de politiques basées sur une identité pour Amazon Cognito](#).

Clés de condition de politique pour Amazon Cognito

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

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

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

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de [contexte de condition AWS globales](#) dans le guide de l'utilisateur IAM.

Pour afficher la liste des clés de condition Amazon Cognito, consultez [Clés de condition pour Amazon Cognito](#) dans la Référence de l'autorisation de service. Pour savoir les actions et ressources avec lesquelles vous pouvez utiliser une clé de condition, consultez [Actions définies par Amazon Cognito](#).

Pour voir des exemples de politiques basées sur une identité pour Amazon Cognito, consultez [Exemples de politiques basées sur une identité pour Amazon Cognito](#).

Listes de contrôle d'accès (ACL) dans Amazon Cognito

Prend en charge les listes ACL

Non

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

Contrôle d'accès par attributs (ABAC) avec Amazon Cognito

Prise en charge d'ABAC (identifications dans les politiques) Partielle

Le contrôle d'accès basé sur les attributs (ABAC) est une politique d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'[élément de condition](#) d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, consultez [Qu'est-ce que le contrôle d'accès basé sur les attributs \(ABAC\) ?](#) dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez [Utilisation du contrôle d'accès par attributs \(ABAC\)](#) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec Amazon Cognito

Prend en charge les informations d'identification temporaires Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation [d'IAM](#) dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez [Changement de rôle \(console\)](#) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez [Informations d'identification de sécurité temporaires dans IAM](#).

Autorisations de principaux entre services pour Amazon Cognito

Prend en charge les sessions d'accès direct (FAS)	Non
---	-----

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez [Transmission des sessions d'accès](#).

Fonctions du service pour Amazon Cognito

Prend en charge les fonctions du service	Oui
--	-----

Une fonction de service est un [rôle IAM](#) qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM.

Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les fonctions du service Amazon Cognito, consultez [Activation de la synchronisation push](#) et [Synchronisation en mode Push](#).

Warning

La modification des autorisations d'une fonction du service peut altérer la fonctionnalité d'Amazon Cognito. Ne modifiez des fonctions du service que quand Amazon Cognito vous le conseille.

Utilisation de rôles liés à un service pour Amazon Cognito

Prend en charge les rôles liés à un service. Oui

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 administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion de rôles liés à un service dans Amazon Cognito, consultez [Utilisation de rôles liés à un service pour Amazon Cognito](#).

Exemples de politiques basées sur une identité pour Amazon Cognito

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ni à modifier des ressources Amazon Cognito. 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. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez [Création de politiques dans l'onglet JSON](#) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par Amazon Cognito, y compris le format des ARN pour chacun des types de ressources, consultez [Actions, ressources et clés de condition pour Amazon Cognito](#) dans la Référence de l'autorisation de service.

Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la console Amazon Cognito](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)
- [Restriction de l'accès à la console à un groupe d'identités spécifique](#)
- [Autoriser l'accès à un jeu de données spécifique pour toutes les identités d'un groupe](#)

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si une personne peut créer, consulter ou supprimer des ressources Amazon Cognito 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 [politiques gérées par AWS](#) ou [politiques gérées par AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule 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 de IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder

l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes 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 [Validation de politique IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Configuration de l'accès aux API protégé par MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

Note

La version d'origine et la nouvelle version de la console Amazon Cognito ont un comportement sous-jacent différent lorsque vous consultez et modifiez vos ressources Amazon Cognito. Si vous aviez autorisé des actions selon le préfixe de service `cognito-idp` uniquement lorsque la condition `aws:ViaAWSService` est vraie, le principal IAM concerné aurait pu être effectif pour les ressources Amazon Cognito dans la console d'origine, mais pas dans la nouvelle console. Pour travailler dans la console Amazon Cognito, ne définissez pas de condition `aws:ViaAWSService` sur les autorisations Amazon Cognito dans votre politique IAM.

Utilisation de la console Amazon Cognito

Pour accéder à la console Amazon Cognito, vous devez disposer d'un ensemble minimum d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources Amazon Cognito présentes dans votre. Compte AWS Si vous créez une stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum

d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la console Amazon Cognito, associez également Amazon ConsoleAccess Cognito ReadOnly AWS ou la politique gérée aux entités. Pour plus d'informations, consultez [Ajout d'autorisations à un utilisateur](#) dans le Guide de l'utilisateur IAM.

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": "*"
}
]
}

```

Restriction de l'accès à la console à un groupe d'identités spécifique

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cognito-identity:ListIdentityPools"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cognito-identity:*"
      ],
      "Resource": "arn:aws:cognito-identity:us-east-1:0123456789:identitypool/us-east-1:1a1a1a1a-ffff-1111-9999-12345678"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cognito-sync:*"
      ],
      "Resource": "arn:aws:cognito-sync:us-east-1:0123456789:identitypool/us-east-1:1a1a1a1a-ffff-1111-9999-12345678"
    }
  ]
}

```

Autoriser l'accès à un jeu de données spécifique pour toutes les identités d'un groupe

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cognito-sync:ListRecords",
        "cognito-sync:UpdateRecords"
      ],
      "Resource": "arn:aws:cognito-sync:us-east-1:0123456789:identitypool/us-east-1:1a1a1a1a-ffff-1111-9999-12345678/identity/*/dataset/UserProfile"
    }
  ]
}
```

Résolution de problèmes d'identité et d'accès dans Amazon Cognito

Pour identifier et résoudre des problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec Amazon Cognito et IAM, utilisez les informations ci-après.

Rubriques

- [Je ne suis pas autorisé à effectuer une action dans Amazon Cognito](#)
- [Je ne suis pas autorisé à effectuer iam : PassRole](#)
- [Je suis un administrateur et je veux autoriser d'autres utilisateurs à accéder à Amazon Cognito](#)
- [Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources Amazon Cognito](#)

Je ne suis pas autorisé à effectuer une action dans Amazon Cognito

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 `cognito-identity:GetWidget` fictives.


```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
cognito-identity: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 *cognito-identity: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 selon lequel vous n'êtes pas autorisé à exécuter l'action *iam:PassRole*, vos stratégies doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon Cognito.

Certains vos Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'erreur suivante se produit quand un utilisateur IAM nommé marymajor tente d'utiliser la console pour exécuter une action dans Amazon Cognito. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action *iam:PassRole*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je suis un administrateur et je veux autoriser d'autres utilisateurs à accéder à Amazon Cognito

Pour permettre à d'autres utilisateurs d'accéder à Amazon Cognito, vous devez créer une entité IAM (utilisateur ou rôle) pour la personne ou l'application qui a besoin d'accéder. Ils utiliseront les

informations d'identification de cette entité pour accéder à AWS. Vous devez ensuite attacher à l'entité une politique qui lui accorde les autorisations appropriées dans Amazon Cognito.

Pour démarrer immédiatement, consultez [Création de votre premier groupe et utilisateur délégué IAM](#) dans le Guide de l'utilisateur IAM.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources Amazon Cognito

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si Amazon Cognito est compatible avec ces fonctionnalités, consultez [Fonctionnement d'Amazon Cognito 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'utilisateur IAM.
- 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'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez [Fournir un accès à des utilisateurs authentifiés en externe \(fédération d'identité\)](#) dans le Guide de l'utilisateur IAM.
- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez [Différence entre les rôles IAM et les politiques basées sur les ressources](#) dans le Guide de l'utilisateur IAM.

Utilisation de rôles liés à un service pour Amazon Cognito

[Amazon Cognito utilise des rôles liés à un AWS Identity and Access Management service \(IAM\)](#). Un rôle lié à un service est un type unique de rôle IAM doté d'une politique de confiance qui permet à un homme d' Service AWS assumer ce rôle. Les rôles liés à un service sont prédéfinis par Amazon

Cognito et incluent toutes les autorisations dont le service a besoin pour appeler AWS d'autres services en votre nom.

Un rôle lié à un service simplifie la configuration d'Amazon Cognito, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. Amazon Cognito définissant les autorisations de ses rôles liés à un service, sauf définition contraire, seul Amazon Cognito peut endosser ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources Amazon Cognito sont ainsi protégées, car vous ne pouvez pas supprimer involontairement l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez [Services AWS qui fonctionnent avec IAM](#) et recherchez les services où Oui figure dans la colonne Rôle lié à un service. Choisissez un Oui ayant un lien permettant de consulter la documentation du rôle lié à un service, pour ce service.

Autorisations de rôle lié à un service pour Amazon Cognito

Amazon Cognito utilise les rôles liés à un service suivants :

- `AWSServiceRoleForAmazonCognitoIdpEmailService`— Permet au service de groupes d'utilisateurs Amazon Cognito d'utiliser vos identités Amazon SES pour envoyer des e-mails.
- `AWSServiceRoleForAmazonCognitoIdp`— Permet aux groupes d'utilisateurs Amazon Cognito de publier des événements et de configurer des points de terminaison pour vos projets Amazon Pinpoint.

`AWSServiceRoleForAmazonCognitoIdpEmailService`

Le rôle lié à un service `AWSServiceRoleForAmazonCognitoIdpEmailService` approuve les services suivants pour endosser le rôle :

- `email.cognito-idp.amazonaws.com`

La politique d'autorisations de rôle permet à Amazon Cognito d'effectuer les actions suivantes sur les ressources spécifiées :

Actions autorisées pour `AWSServiceRoleForAmazonCognitoIdpEmailService` :

- Action : `ses:SendEmail` et `ses:SendRawEmail`
- Ressource : *

La politique refuse à Amazon Cognito la possibilité d'effectuer les actions suivantes sur les ressources spécifiées :

Actions refusées

- Action : `ses:List*`
- Ressource : *

Avec ces autorisations, Amazon Cognito peut utiliser vos adresses électroniques vérifiées dans Amazon SES uniquement pour envoyer des courriels à vos utilisateurs. Amazon Cognito envoie des courriels à vos utilisateurs quand ceux-ci effectuent certaines actions dans l'appli cliente pour un groupe d'utilisateurs, comme une inscription ou une réinitialisation de mot de passe.

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez [Autorisations de rôles liés à un service](#) dans le Guide de l'utilisateur IAM.

`AWSServiceRoleForAmazonCognitoIdp`

Le rôle `AWSServiceRoleForAmazonCognitoIdp` lié à un service fait confiance aux services suivants pour assumer le rôle :

- `email.cognito-idp.amazonaws.com`

La politique d'autorisations de rôle permet à Amazon Cognito d'effectuer les actions suivantes sur les ressources indiquées :

Actions autorisées pour `AWSServiceRoleForAmazonCognitoIdp`

- Action : `cognito-idp:Describe`
- Ressource : *

Avec cette autorisation, Amazon Cognito peut appeler les opérations d'API Amazon Cognito `Describe` pour vous.

Note

Lorsque vous intégrez Amazon Cognito avec Amazon Pinpoint en utilisant `createUserPoolClient` et `updateUserPoolClient`, des autorisations d'accès aux ressources sont ajoutées au rôle lié à un service (SLR) en tant que politique incluse. La politique incluse fournit des autorisations `mobiletargeting:UpdateEndpoint` et `mobiletargeting:PutEvents`. Celles-ci permettent à Amazon Cognito de publier des événements et de configurer des points de terminaison pour les projets Pinpoint que vous intégrez avec Cognito.

Création d'un rôle lié à un service pour Amazon Cognito

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous configurez un groupe d'utilisateurs afin qu'il utilise votre configuration Amazon SES pour gérer la livraison des e-mails dans l' AWS Management Console API Amazon Cognito ou dans l'API Amazon Cognito, Amazon Cognito crée le rôle lié au service pour vous. AWS CLI

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous configurez un groupe d'utilisateurs pour utiliser votre configuration Amazon SES pour gérer la livraison des courriels, Amazon Cognito crée à nouveau le rôle lié à un service pour vous.

Pour permettre à Amazon Cognito de créer ce rôle, les autorisations IAM que vous utilisez pour configurer votre groupe d'utilisateurs doivent inclure l'action `iam:CreateServiceLinkedRole`. Pour plus d'informations sur la mise à jour des autorisations dans IAM, consultez [Modification des autorisations pour un utilisateur IAM](#) dans le Guide de l'utilisateur IAM.

Modification d'un rôle lié à un service pour Amazon Cognito

Vous ne pouvez pas modifier les rôles `AmazonCognitoIdp` ou les rôles `AmazonCognitoIdpEmailService` liés à un service dans AWS Identity and Access Management. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence au rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez [Modification d'un rôle lié à un service](#) dans le guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour Amazon Cognito

Si vous n'avez plus besoin d'utiliser une fonctionnalité ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. Si vous supprimez ce rôle, vous ne retenez que les entités qu'Amazon Cognito surveille ou gère activement. Avant de supprimer des rôles AmazonCognitoIam ou des rôles AmazonCognitoIamEmailService liés à un service, vous devez effectuer l'une des opérations suivantes pour chaque groupe d'utilisateurs qui utilise le rôle :

- Supprimer le groupe d'utilisateurs.
- Mettre à jour les paramètres de messagerie dans le groupe d'utilisateurs afin d'utiliser la fonctionnalité de messagerie par défaut. Le paramètre par défaut n'utilise pas le rôle lié à un service.

N'oubliez pas d'exécuter l'action dans chacune d'elles Région AWS avec un groupe d'utilisateurs utilisant le rôle.

Note

Si le service Amazon Cognito utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer un groupe d'utilisateurs Amazon Cognito

1. Connectez-vous à la console Amazon Cognito AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/cognito>
2. Sélectionnez Gérer les groupes d'utilisateurs.
3. Sur la page Vos groupes d'utilisateurs, choisissez le groupe d'utilisateurs que vous souhaitez supprimer.
4. Sélectionnez Supprimer le groupe.
5. Dans la fenêtre Supprimer un groupe d'utilisateurs, saisissez **delete**, puis choisissez Supprimer le groupe.

Pour mettre à jour un groupe d'utilisateurs Amazon Cognito afin qu'il utilise la fonctionnalité de courriel par défaut

1. Connectez-vous à la console Amazon Cognito AWS Management Console et ouvrez-la à l'adresse. <https://console.aws.amazon.com/cognito>
2. Sélectionnez Gérer les groupes d'utilisateurs.
3. Sur la page Vos groupes d'utilisateurs, choisissez le groupe d'utilisateurs que vous souhaitez mettre à jour.
4. Dans le menu de navigation de gauche, choisissez Personnalisation des messages.
5. Sous Do you want to send emails through your Amazon SES Configuration? (Voulez-vous à envoyer les courriels via votre configuration Amazon SES ?), choisissez No - Use Cognito (Default) [Non - Utiliser Cognito (par défaut)].
6. Une fois que vous avez défini les options de votre compte de messagerie, choisissez Enregistrer les modifications.

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM AWS CLI, l'API ou l' AWS API pour supprimer des rôles AmazonCognitoIdp ou liés à un AmazonCognitoIdpEmailService service. Pour plus d'informations, consultez [Suppression d'un rôle lié à un service](#) dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés à un service Amazon Cognito

Amazon Cognito prend en charge les rôles liés au service partout Régions AWS où le service est disponible. Pour plus d'informations, consultez [Régions AWS and Endpoints](#).

Journalisation et surveillance dans Amazon Cognito

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'Amazon Cognito et de vos autres AWS solutions. Amazon Cognito prend actuellement en charge les Services AWS suivants, qui vous permettent de surveiller votre organisation et l'activité en son sein.

- AWS CloudTrail — CloudTrail Vous pouvez ainsi capturer des appels d'API depuis la console Amazon Cognito et des appels de code vers les opérations de l'API Amazon Cognito. Par exemple, lorsqu'un utilisateur s'authentifie, il CloudTrail peut enregistrer des informations telles que l'adresse IP de la demande, l'auteur de la demande et la date à laquelle elle a été faite.
- Amazon CloudWatch Logs — Avec CloudWatch Logs, vous pouvez envoyer des journaux détaillés de l'activité des utilisateurs à un groupe de journaux. Par exemple, vous pouvez consulter les

journaux d'activité détaillés des utilisateurs pour résoudre les problèmes liés à l'envoi d'e-mails et de SMS à vos utilisateurs.

- Amazon CloudWatch Metrics — Grâce aux CloudWatch métriques, vous pouvez surveiller, signaler et prendre des mesures automatiques en cas d'événement en temps quasi réel. Par exemple, vous pouvez créer des CloudWatch tableaux de bord sur les indicateurs fournis afin de surveiller vos groupes d'utilisateurs Amazon Cognito, ou vous pouvez CloudWatch créer des alarmes sur les indicateurs fournis pour vous avertir en cas de dépassement d'un seuil défini.
- Amazon CloudWatch Logs Insights — Avec CloudWatch Logs Insights, vous pouvez configurer l'envoi CloudTrail d'événements à des CloudWatch fins de surveillance des fichiers CloudTrail journaux Amazon Cognito.

Rubriques

- [Coûts de surveillance](#)
- [Suivi des quotas, de l'utilisation CloudWatch et des Quotas de Service](#)
- [Journalisation des appels d'API Amazon Cognito avec AWS CloudTrail](#)

Coûts de surveillance

Amazon Cognito facture les dimensions suivantes de votre utilisation.

- Groupe d'utilisateurs utilisateurs actifs mensuels (MAU)
- Groupe d'utilisateurs MAU connectés avec la fédération OIDC ou SAML
- MAU dans un groupe d'utilisateurs dotés de fonctionnalités de sécurité avancées
- Groupe d'utilisateurs actifs, clients de l'application et volume de demandes d'autorisation machine à machine (M2M) avec octroi d'informations d'identification client
- Utilisation achetée supérieure aux quotas par défaut pour certaines catégories d'API de pool d'utilisateurs

En outre, les fonctionnalités de votre groupe d'utilisateurs, telles que les e-mails, les SMS et les déclencheurs Lambda, peuvent entraîner des coûts pour les services dépendants. Pour un aperçu complet, consultez la section Tarification [d'Amazon Cognito](#).

Visualisation et anticipation des coûts

Vous pouvez consulter et générer des rapports sur vos AWS coûts dans la [AWS Billing and Cost Management console](#). Vous trouverez les derniers frais que vous avez facturés pour Amazon Cognito dans la section Facturation et paiements. Sous Factures, Frais par service, filtrez Cognito pour afficher votre consommation. Pour plus d'informations, consultez la section [Viewing your bill](#) (Affichage d'une facture) dans le Guide de l'utilisateur AWS Billing .

Pour surveiller les taux de demandes d'API, consultez la métrique d'utilisation dans la console Service Quotas. Par exemple, les demandes d'informations d'identification des clients s'affichent sous la forme « Taux de ClientAuthentication demandes ». Dans votre facture, ces demandes sont associées à l'application cliente qui les a produites. Grâce à ces informations, vous pouvez répartir équitablement les coûts entre les locataires dans une [architecture multi-locataires](#).

Pour connaître le nombre de demandes M2M sur une période donnée, vous pouvez également envoyer des [AWS CloudTrail événements à CloudWatch Logs à des fins d'analyse](#). Recherchez dans vos CloudTrail événements les Token_POST événements bénéficiant d'une autorisation d'identification client. La requête CloudWatch Insights suivante renvoie ce nombre.

```
filter eventName = "Token_POST" and @message like '"grant_type":["client_credentials"]'
| stats count(*)
```

Gestion des coûts

Amazon Cognito facture en fonction du nombre d'utilisateurs, de l'utilisation des fonctionnalités et du volume de demandes. Voici quelques conseils pour gérer les coûts dans Amazon Cognito,

N'activez pas les utilisateurs inactifs

Les opérations typiques qui rendent un utilisateur actif sont la connexion, l'inscription et la réinitialisation du mot de passe. Pour une liste plus complète, voir [Monthly active users \(Utilisateurs actifs mensuels\)](#). Amazon Cognito ne prend pas en compte les utilisateurs inactifs dans le calcul de votre facture. Évitez toute opération qui active un utilisateur. Au lieu de l'opération [AdminGetUser](#) API, interrogez les utilisateurs avec l'[ListUsers](#) opération. N'effectuez pas de tests administratifs volumineux sur les opérations du groupe d'utilisateurs avec des utilisateurs inactifs.

Lier les utilisateurs fédérés

Les utilisateurs qui se connectent avec un fournisseur d'identité SAML 2.0 ou OpenID Connect (OIDC) ont un [coût](#) plus élevé que les utilisateurs locaux. Vous pouvez [associer ces utilisateurs à un](#)

[profil utilisateur local](#). Un utilisateur lié peut se connecter en tant qu'utilisateur local avec les attributs et les accès fournis avec son utilisateur fédéré. Les utilisateurs de SAML ou OIDC IdPs qui, au cours d'un mois, ne se connectent qu'avec un compte local associé sont facturés comme des utilisateurs locaux.

Gérez les taux de demandes

Si votre groupe d'utilisateurs approche de la limite supérieure de votre quota, vous pouvez envisager d'acheter de la capacité supplémentaire pour gérer le volume. Vous pourriez être en mesure de réduire le volume de demandes dans votre application. Pour plus d'informations, consultez [Optimisation des taux de demandes pour les limites de quotas](#).

Demandez un nouveau jeton uniquement lorsque vous en avez besoin

L'autorisation de machine à machine (M2M) avec l'octroi d'informations d'identification aux clients peut atteindre un volume élevé de demandes de jetons. Chaque nouvelle demande de jeton a un effet sur votre quota de taux de demandes et sur le montant de votre facture. Pour optimiser les coûts, incluez les paramètres d'expiration des jetons et la gestion des jetons dans la conception de vos applications.

- [Jetons d'accès au cache](#) afin que, lorsque votre application demande un nouveau jeton, elle reçoive une version mise en cache d'un jeton émis précédemment. Lorsque vous implémentez cette méthode, votre proxy de mise en cache agit comme une protection contre les applications qui demandent des jetons d'accès sans se rendre compte de l'expiration des jetons précédemment acquis. La mise en cache des jetons est idéale pour les microservices de courte durée tels que les fonctions Lambda et les conteneurs Docker.
- Mettez en œuvre des mécanismes de gestion des jetons dans vos applications qui tiennent compte de l'expiration des jetons. Ne demandez pas de nouveau jeton tant que les jetons précédents n'ont pas expiré. Évaluez les besoins de confidentialité et de disponibilité de chaque application et configurez le client de l'application du pool d'utilisateurs pour émettre des jetons d'accès avec une période de validité appropriée. La durée du jeton personnalisé fonctionne mieux avec des API et des serveurs à plus longue durée de vie capables de gérer de manière permanente la fréquence des demandes d'informations d'identification.

Supprimer les informations d'identification client non utilisées (clients d'applications)

Les factures d'autorisation M2M sont basées sur deux facteurs : le taux de demandes de jetons et le nombre de clients de l'application qui accordent des identifiants aux clients. Lorsque les clients de

l'application pour l'autorisation M2M ne sont pas utilisés, supprimez-les ou retirez leur autorisation d'émettre des informations d'identification client. Pour plus d'informations sur la gestion de la configuration du client d'application, consultez [Clients de l'application de groupe d'utilisateurs](#).

Gérez la sécurité avancée

Lorsque vous configurez des [fonctionnalités de sécurité avancées](#) dans un groupe d'utilisateurs, le taux de facturation de sécurité avancé s'applique à toutes les MAU du groupe d'utilisateurs. Si certains de vos utilisateurs n'ont pas besoin de fonctionnalités de sécurité avancées, séparez-les dans un autre groupe d'utilisateurs.

Suivi des quotas, de l'utilisation CloudWatch et des Quotas de Service

Vous pouvez surveiller les groupes d'utilisateurs d'Amazon Cognito à l'aide d'Amazon CloudWatch ou de Service Quotas. Vous pouvez également surveiller l'utilisation des groupes d'identités dans Service Quotas. CloudWatch collecte des données brutes et les transforme en indicateurs lisibles en temps quasi réel. Dans CloudWatch, vous pouvez définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour créer une CloudWatch alarme pour un quota de service, voir [Créer une CloudWatch alarme](#). Les métriques Amazon Cognito sont disponibles à intervalles de cinq minutes. Pour plus d'informations sur les périodes de conservation CloudWatch, consultez la [page CloudWatch FAQ d'Amazon](#).

Vous pouvez utiliser Service Quotas pour afficher et gérer l'utilisation des quotas de vos groupes d'utilisateurs et réserves d'identités Amazon Cognito. La console Service Quotas comporte trois fonctionnalités : afficher les quotas de service, demander une augmentation de quota de service et afficher l'utilisation actuelle. Vous pouvez utiliser la première fonction pour afficher les quotas et vérifier s'ils sont ajustables. Vous pouvez utiliser la deuxième fonction pour demander une augmentation de Service Quotas. Vous pouvez utiliser la dernière fonction pour afficher l'utilisation des quotas. Cette fonctionnalité n'est disponible qu'après que votre compte a été actif pendant un certain temps. Pour plus d'informations sur l'affichage des quotas dans la console Service Quotas, consultez [Affichage de Service Quotas](#).

Note

Les métriques Amazon Cognito sont disponibles à intervalles de 5 minutes. Pour plus d'informations sur les périodes de conservation CloudWatch, consultez la [page CloudWatch FAQ d'Amazon](#).

Si vous êtes connecté à un compte configuré en tant que Compte AWS et compte de surveillance dans le cadre de l'observabilité CloudWatch entre comptes, vous pouvez utiliser ce compte de surveillance pour visualiser les quotas de service et définir des alarmes pour les métriques des comptes sources liés à ce compte de surveillance. Pour plus d'informations, consultez la [CloudWatch section Observabilité entre comptes](#).

Rubriques

- [Journalisation d'activités supplémentaires à partir des groupes d'utilisateurs Amazon Cognito](#)
- [Métriques pour les groupes d'utilisateurs Amazon Cognito](#)
- [Dimensions pour les groupes d'utilisateurs Amazon Cognito](#)
- [Utiliser la console Service Quotas pour suivre les métriques](#)
- [Utiliser la CloudWatch console pour suivre les métriques](#)
- [Création d'une CloudWatch alarme pour un quota](#)

Journalisation d'activités supplémentaires à partir des groupes d'utilisateurs Amazon Cognito

Vous pouvez configurer votre groupe d'utilisateurs pour envoyer des journaux détaillés de certaines activités supplémentaires à un groupe de CloudWatch journaux. Ces journaux sont d'une granularité plus fine que ceux contenus dans AWS CloudTrail le fichier et peuvent être utiles pour résoudre les problèmes de votre groupe d'utilisateurs. Lorsque vous activez cette fonctionnalité, vous pouvez choisir le groupe de journaux auquel vous souhaitez qu'Amazon Cognito envoie les journaux. La journalisation des activités des utilisateurs est utile lorsque vous souhaitez connaître l'état des e-mails et des SMS envoyés par votre groupe d'utilisateurs via Amazon SNS et Amazon SES.

Actuellement, vous pouvez uniquement livrer les journaux de notification utilisateur au niveau de l'erreur provenant de votre groupe d'utilisateurs.

La journalisation détaillée ne remplace ni ne modifie les fonctions de journalisation suivantes des groupes d'utilisateurs.

1. CloudTrail journaux des activités courantes des utilisateurs, telles que l'inscription et la connexion.
2. Analyse de l'activité des utilisateurs à grande échelle à l'aide de CloudWatch métriques.

Séparément, vous pouvez également trouver les journaux des [tâches d'importation des utilisateurs et des déclencheurs Lambda](#) dans CloudWatch Logs. Amazon Cognito et Lambda stockent ces

journaux dans des groupes de journaux différents de ceux que vous avez définis pour les journaux d'activité détaillés.

Vous pouvez configurer des journaux d'activité détaillés avec l'API des groupes d'utilisateurs Amazon Cognito dans une demande d'[SetLogDeliveryConfiguration](#) API. Vous pouvez consulter la configuration de journalisation d'un groupe d'utilisateurs dans une demande d'[GetLogDeliveryConfiguration](#) API.

Vous devez autoriser ces demandes avec des AWS informations d'identification disposant des autorisations suivantes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ManageUserPoolLogs",
      "Action": [
        "cognito-idp:SetLogDeliveryConfiguration",
        "cognito-idp:GetLogDeliveryConfiguration",
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid": "CognitoLog",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid": "CognitoLoggingCWL",
      "Action": [
        "logs:PutResourcePolicy",

```

```
        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow"
}
]
```

Voici un exemple d'événement dans un groupe d'utilisateurs. Ce schéma de journal est soumis à modification. Certains champs peuvent être consignés avec des valeurs nulles.

```
{
  "eventTimestamp": "1687297330677",
  "eventSource": "USER_NOTIFICATION",
  "logLevel": "ERROR",
  "message": {
    "details": "String"
  },
  "logSourceId": {
    "userPoolId": "String"
  }
}
```

L'envoi des journaux depuis Amazon Cognito est la meilleure solution. Le volume de journaux fourni par votre groupe d'utilisateurs et vos quotas de service pour les CloudWatch journaux peuvent avoir une incidence sur la diffusion des journaux.

CloudWatch Les frais de journalisation s'appliquent lorsque la livraison des journaux est activée. Pour plus d'informations, consultez la section [Vended Logs](#) sur Amazon CloudWatch Pricing.

Pour envoyer des journaux à des groupes de journaux dont la taille de la politique de ressources est supérieure à 5 120 caractères, configurez un groupe de journaux avec un chemin commençant par /aws/vendedlogs. Pour plus d'informations, consultez la section [Activation de la journalisation à partir de certains AWS services](#).

Métriques pour les groupes d'utilisateurs Amazon Cognito

Le tableau suivant répertorie les métriques disponibles pour les groupes d'utilisateurs Amazon Cognito. L'espace de noms des métriques Amazon CloudWatch pour Amazon Cognito est AWS/

Cognito. Pour plus d'informations, consultez [Namespaces](#) dans le guide de CloudWatch l'utilisateur Amazon.

Note

Les métriques qui n'ont pas eu de nouveaux points de données au cours des deux dernières semaines ne s'affichent pas dans la console. Elles ne s'affichent pas non plus lorsque vous saisissez leur nom de métrique ou de dimension dans la zone de recherche sous l'onglet Toutes les métriques dans la console. En outre, elles ne sont pas renvoyées dans les résultats d'une commande `list-metrics`. La meilleure façon de récupérer ces métriques est d'utiliser les `get-metric-statistics` commandes `get-metric-data` or de la AWS CLI.

Métrique	Description
SignUpSuccesses	<p>Indique le nombre total de demandes d'enregistrement utilisateur adressées au groupe d'utilisateurs Amazon Cognito qui ont abouti. Une demande d'enregistrement d'utilisateur réussie génère la valeur 1, tandis qu'une demande infructueuse indique la valeur 0. Une demande limitée étant également considérée comme inaboutie, elle produit le résultat 0.</p> <p>Pour obtenir le pourcentage de demandes d'enregistrement d'utilisateurs ayant abouti, utilisez la statistique <code>Average</code> pour cette métrique. Pour compter le nombre total de demandes d'enregistrement d'utilisateurs, utilisez la statistique <code>Sample Count</code> pour cette métrique. Pour compter le nombre total de demandes d'enregistrement d'utilisateurs ayant abouti, utilisez la statistique <code>Sum</code> pour cette métrique. Pour compter le nombre total de demandes d'enregistrement d'utilisateurs ayant échoué, utilisez l' <code>CloudWatch Math</code> expressio</p>

Métrique	Description
	<p>n et soustrayez la Sum statistique de la Sample Count statistique.</p> <p>Cette métrique est publiée pour chaque groupe d'utilisateurs pour chaque client de groupe d'utilisateurs. Si l'enregistrement utilisateur est effectué par un administrateur, la métrique est publiée avec le client de groupe d'utilisateurs en tant que Admin.</p> <p>Notez que cette métrique n'est pas émise pour les cas d'importation d'utilisateurs et de migration d'utilisateurs.</p> <p>Dimension de métrique : UserPool, UserPoolClient</p> <p>Unités : nombre</p>

Métrique	Description
SignUpThrottles	<p>Indique le nombre total de demandes d'enregistrement utilisateur envoyées au groupe d'utilisateurs Amazon Cognito qui ont été limitées. La valeur 1 est générée chaque fois qu'une demande d'enregistrement d'utilisateur est restreinte.</p> <p>Pour compter le nombre total de demandes d'enregistrement d'utilisateurs restreintes, utilisez la statistique Sum pour cette métrique.</p> <p>Cette métrique est publiée pour chaque groupe d'utilisateurs pour chaque client. Si la demande limitée a été effectuée par un administrateur, la métrique est publiée avec le client de groupe d'utilisateurs en tant que Admin.</p> <p>Dimension de métrique : <code>UserPool</code>, <code>UserPoolClient</code></p> <p>Unités : nombre</p>

Métrique	Description
SignInSuccesses	<p>Indique le nombre total de demandes d'authentification d'utilisateur envoyées au groupe d'utilisateurs Amazon Cognito qui ont abouti. Une authentification utilisateur est considéré e comme réussie lorsque le jeton d'authentification est transmis à l'utilisateur. Une authentification réussie génère la valeur 1, tandis qu'une demande infructueuse indique la valeur 0. Une demande limitée étant également considérée comme inaboutie, elle produit le résultat 0.</p> <p>Pour obtenir le pourcentage de demandes d'authentification ayant abouti, utilisez la statistique <code>Average</code> pour cette métrique. Pour compter le nombre total de demandes d'authentification d'utilisateurs, utilisez la statistique <code>Sample Count</code> pour cette métrique. Pour compter le nombre total de demandes d'authentification d'utilisateurs ayant abouti, utilisez la statistique <code>Sum</code> pour cette métrique. Pour compter le nombre total de demandes d'authentification utilisateur ayant échoué, utilisez l' <code>CloudWatch Mathexpression</code> et soustrayez la <code>Sum</code> statistique de la <code>Sample Count</code> statistique.</p> <p>Cette métrique est publiée pour chaque groupe d'utilisateurs pour chaque client. Si un client de groupe d'utilisateurs non valide est fourni avec une demande, la valeur du client de groupe d'utilisateurs correspondante dans la métrique contient la valeur fixe <code>Invalid</code> au lieu de la valeur non valide réelle envoyée dans cette demande.</p>

Métrique	Description
	<p>Notez que les demandes d'actualisation du jeton Amazon Cognito ne sont pas incluses dans cette métrique. Une métrique distincte permet de fournir les statistiques du jeton Refresh.</p> <p>Dimension de métrique : <code>UserPool</code>, <code>UserPoolClient</code></p> <p>Unités : nombre</p>

Métrique	Description
SignInThrottles	<p>Indique le nombre total de demandes d'authentification d'utilisateurs envoyées au groupe d'utilisateurs Amazon Cognito qui ont été limitées. La valeur 1 est générée chaque fois qu'une demande d'authentification est restreinte.</p> <p>Pour compter le nombre total de demandes d'authentification d'utilisateurs restreintes, utilisez la statistique <code>Sum</code> pour cette métrique.</p> <p>Cette métrique est publiée pour chaque groupe d'utilisateurs pour chaque client. Si un client de groupe d'utilisateurs non valide est fourni avec une demande, la valeur du client de groupe d'utilisateurs correspondante dans la métrique contient la valeur fixe <code>Invalid</code> au lieu de la valeur non valide réelle envoyée dans cette demande.</p> <p>Les demandes d'actualisation de jeton Amazon Cognito ne sont pas incluses dans cette métrique. Une métrique distincte permet de fournir les statistiques du jeton <code>Refresh</code>.</p> <p>Dimension de métrique : <code>UserPool</code>, <code>UserPoolClient</code></p> <p>Unités : nombre</p>

Métrique	Description
TokenRefreshSuccesses	<p>Indique le nombre total de demandes d'actualisation de jeton Amazon Cognito envoyées au groupe d'utilisateurs Amazon Cognito qui ont abouti. Une demande d'actualisation de jeton Amazon Cognito aboutie génère la valeur 1, tandis qu'une demande non aboutie génère la valeur 0. Une demande limitée étant également considérée comme inaboutie, elle produit le résultat 0.</p> <p>Pour obtenir le pourcentage de demandes d'actualisation de jeton Amazon Cognito ayant abouti, utilisez la statistique <code>Average</code> sur cette métrique. Pour compter le nombre total de demandes d'actualisation de jeton Amazon Cognito, utilisez la statistique <code>Sample Count</code> sur cette métrique. Pour compter le nombre total de demandes d'actualisation de jeton Amazon Cognito ayant abouti, utilisez la statistique <code>Sum</code> sur cette métrique. Pour compter le nombre total de demandes d'actualisation d'un jeton Amazon Cognito ayant échoué, utilisez l'expression <code>CloudWatchMathexpression</code> et soustrayez la <code>Sum</code> statistique de la statistique <code>Sample Count</code>.</p> <p>Cette métrique est publiée par client de groupe d'utilisateurs. Si un client de groupe d'utilisateurs non valide figure dans une demande, la valeur de ce client contient une valeur fixe <code>Invalid</code>.</p> <p>Dimension de métrique : <code>UserPool</code>, <code>UserPoolClient</code></p>

Métrique	Description
	Unités : nombre
TokenRefreshThrottles	<p>Indique le nombre total de demandes d'actualisation de jeton Amazon Cognito envoyées au groupe d'utilisateurs Amazon Cognito qui ont échoué. La valeur 1 est publiée chaque fois qu'une demande d'actualisation de jeton Amazon Cognito est limitée.</p> <p>Pour compter le nombre total de demandes d'actualisation de jeton Amazon Cognito limitées, utilisez la statistique Sum pour cette métrique.</p> <p>Cette métrique est publiée pour chaque groupe d'utilisateurs pour chaque client. Si un client de groupe d'utilisateurs non valide est fourni avec une demande, la valeur du client de groupe d'utilisateurs correspondante dans la métrique contient la valeur fixe Invalid au lieu de la valeur non valide réelle envoyée dans cette demande.</p> <p>Dimension de métrique : UserPool, UserPoolClient</p> <p>Unités : nombre</p>

Métrique	Description
FederationSuccesses	<p>Indique le nombre total de demandes de fédération d'identité envoyées au groupe d'utilisateurs Amazon Cognito qui ont abouti. Une fédération d'identité est considérée comme réussie quand Amazon Cognito émet des jetons d'authentification pour l'utilisateur. Une demande de fédération d'identité réussie génère la valeur 1, tandis qu'une demande infructueuse indique la valeur 0. Les demandes limitées et les demandes qui génèrent un code d'autorisation mais aucun jeton produisent une valeur de 0.</p> <p>Pour obtenir le pourcentage de demandes de fédération d'identité ayant abouti, utilisez la statistique <code>Average</code> pour cette métrique. Pour compter le nombre total de demandes de fédération d'identité, utilisez la statistique <code>Sample Count</code> pour cette métrique. Pour compter le nombre total de demandes de fédération d'identité ayant abouti, utilisez la statistique <code>Sum</code> pour cette métrique. Pour compter le nombre total de demandes de fédération d'identité ayant échoué, utilisez l' <code>CloudWatch Mathexpression</code> et soustrayez la <code>Sum</code> statistique de la <code>Sample Count</code> statistique.</p> <p>Dimension de métrique : <code>UserPool</code>, <code>UserPoolClient</code> , <code>IdentityProvider</code></p> <p>Unités : nombre</p>

Métrique	Description
FederationThrottles	<p>Indique le nombre total de demandes de fédération d'identité envoyées au groupe d'utilisateurs Amazon Cognito qui ont été limitées. La valeur 1 est publiée chaque fois qu'une demande de fédération d'identité est limitée.</p> <p>Pour compter le nombre total de demandes de fédération d'identité restreintes, utilisez la statistique Sum pour cette métrique.</p> <p>Dimension de métrique : <code>UserPool</code>, <code>UserPoolClient</code> , <code>IdentityProvider</code></p> <p>Unités : nombre</p>
CallCount	<p>Indique le nombre total d'appels effectués par des clients en rapport avec une catégorie. Cette métrique inclut tous les appels, tels que les appels limités, les appels échoués et les appels réussis.</p> <p>Cette métrique est disponible dans le namespace Utilisation.</p> <p>Le quota de catégorie est appliqué pour chaque AWS compte dans tous les groupes d'utilisateurs d'un compte et d'une région.</p> <p>Vous pouvez compter le nombre total d'appels dans une catégorie à l'aide de la statistique Sum pour cette métrique.</p> <p>Dimension de métrique : <code>Service</code>, <code>Type</code>, <code>Ressource</code>, <code>Classe</code></p> <p>Unités : nombre</p>

Métrique	Description
ThrottleCount	<p>Indique le nombre total d'appels limités en rapport avec une catégorie.</p> <p>Cette métrique est disponible dans le namespace Utilisation.</p> <p>Cette métrique est publiée au niveau du compte.</p> <p>Vous pouvez compter le nombre total d'appels dans une catégorie à l'aide de la statistique Sum pour cette métrique.</p> <p>Dimension de métrique : Service, Type, Ressource, Classe</p> <p>Unités : nombre</p>

Dimensions pour les groupes d'utilisateurs Amazon Cognito

Les dimensions suivantes permettent d'affiner les métriques d'utilisation publiées par Amazon Cognito. Les dimensions s'appliquent uniquement aux métriques CallCount et ThrottleCount .

Dimension	Description
Service	Nom du AWS service contenant la ressource . Pour les métriques d'utilisation d'Amazon Cognito, la valeur de cette dimension est Cognito user pool.
Type	Type d'entité faisant l'objet d'un rapport. La seule valeur valide pour les métriques d'utilisation d'Amazon Cognito est l'API.
Ressource	Type de ressource en cours d'exécution. La seule valeur valide est un nom de catégorie.

Dimension	Description
Classe	Classe de ressource suivie. Amazon Cognito n'utilise pas la dimension Classe.

Utiliser la console Service Quotas pour suivre les métriques

Vous pouvez afficher et gérer vos quotas de groupes d'utilisateurs et de réserves d'identités Amazon Cognito à partir d'un emplacement central avec Service Quotas. La console Service Quotas vous permet de voir les détails d'un quota spécifique, de surveiller son utilisation et d'en demander une augmentation. Pour certains types de quotas, vous pouvez créer une CloudWatch alarme pour suivre l'utilisation de vos quotas. Pour en savoir plus sur les métriques Amazon Cognito que vous pouvez suivre, consultez [Suivre l'usage des quotas](#).

Pour afficher l'utilisation des quotas de service des groupes d'identité et des groupes d'utilisateurs Amazon Cognito, procédez comme suit.

1. Ouvrez la console [Service Quotas](#).
2. Dans le panneau de navigation, choisissez Services AWS .
3. Dans la liste des services AWS , recherchez et choisissez les groupes d'utilisateurs Amazon Cognito ou les Identités fédérées Amazon Cognito. La page des quota de service s'affiche.
4. Sélectionnez un quota qui prend en charge CloudWatch la surveillance. Par exemple, choisissez `Rate of UserAuthentication requests` dans des groupes d'utilisateurs Amazon Cognito.
5. Faites défiler jusqu'à Surveillance. Cette section n'apparaît que pour les quotas qui prennent en charge le CloudWatch suivi.
6. Dans Surveillance, vous pouvez voir l'utilisation actuelle du quota de service dans le graphique.
7. Dans Surveillance, sélectionnez une période d'une heure, de trois heures, de douze heures, d'un jour, de trois jours ou d'une semaine.
8. Sélectionnez une zone quelconque à l'intérieur du graphique pour afficher le pourcentage d'utilisation du quota de service. À partir de là, vous pouvez ajouter le graphique à votre tableau de bord ou utiliser le menu d'action pour sélectionner Afficher dans les métriques, ce qui vous permettra d'accéder aux métriques associées dans la CloudWatch console.

Utiliser la CloudWatch console pour suivre les métriques

Vous pouvez suivre et collecter les statistiques des groupes d'utilisateurs Amazon Cognito à l'aide de CloudWatch. Le CloudWatch tableau de bord affichera les statistiques relatives à chaque AWS service que vous utilisez. Vous pouvez l'utiliser CloudWatch pour créer des alarmes métriques. Vous pouvez configurer les alarmes pour vous envoyer des notifications ou apporter une modification à une ressource spécifique que vous surveillez. Pour consulter les métriques de quota de service dans CloudWatch, procédez comme suit.

1. Ouvrez la [CloudWatch console](#).
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Dans Toutes les métriques, sélectionnez une métrique et une dimension.
4. Sélectionnez la case à cocher en regard d'une métrique. Les métriques apparaissent dans le graphique.

Note

Les métriques qui n'ont pas eu de nouveaux points de données au cours des deux dernières semaines ne s'affichent pas dans la console. Elles ne s'affichent pas non plus lorsque vous saisissez leur nom de métrique ou de dimension dans la zone de recherche sous l'onglet Toutes les métriques dans la console, et ne sont pas renvoyées dans les résultats d'une commande `list-metrics`. La meilleure façon de récupérer ces métriques est d'utiliser les commandes `get-metric-data` ou `get-metric-statistics` dans la CLI de AWS.

Création d'une CloudWatch alarme pour un quota

Amazon Cognito fournit des statistiques CloudWatch d'utilisation qui correspondent aux quotas de AWS service `CallCount` et `ThrottleCount` aux API. Pour plus d'informations sur le suivi de l'utilisation dans CloudWatch, consultez [Suivre l'usage des quotas](#).

Dans la console Service Quotas, vous pouvez créer des alarmes qui vous alertent quand votre utilisation d'un service approche du quota défini pour celui-ci. Pour savoir comment configurer une CloudWatch alarme à l'aide de la console Service Quotas, voir [Quotas de service et CloudWatch alarmes](#).

Journalisation des appels d'API Amazon Cognito avec AWS CloudTrail

Amazon Cognito est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou un AWS service dans Amazon Cognito. CloudTrail capture un sous-ensemble d'appels d'API pour Amazon Cognito sous forme d'événements, y compris les appels depuis la console Amazon Cognito et les appels de code vers les opérations de l'API Amazon Cognito. Si vous créez un suivi, vous pouvez choisir de transférer des CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour Amazon Cognito. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon Cognito, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le [guide de AWS CloudTrail l'utilisateur](#).

Vous pouvez également créer des CloudWatch alarmes Amazon pour des CloudTrail événements spécifiques. Par exemple, vous pouvez configurer CloudWatch pour déclencher une alarme si la configuration d'un pool d'identités est modifiée. Pour plus d'informations, voir [Création d' CloudWatch alarmes pour CloudTrail des événements : exemples](#).

Rubriques

- [Informations sur Amazon Cognito dans CloudTrail](#)
- [Présentation des événements de connexion Amazon Cognito](#)
- [Analyse des CloudTrail événements Amazon Cognito avec Amazon CloudWatch Logs Insights](#)

Informations sur Amazon Cognito dans CloudTrail

CloudTrail est activé lorsque vous créez votre Compte AWS. Lorsqu'une activité événementielle prise en charge se produit dans Amazon Cognito, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section [Affichage des événements avec l'historique des CloudTrail événements](#).

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements relatifs à Amazon Cognito, créez un suivi. Un CloudTrail suivi fournit des fichiers journaux à un

compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions . Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 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. section withinPour plus d'informations, consultez :

- [Présentation de la création d'un journal d'activité](#)
- [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](#)

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez l'élément [CloudTrail UserIdentity](#).

Données confidentielles dans AWS CloudTrail

Dans la mesure où les groupes d'utilisateurs et les groupes d'identités traitent les données utilisateur, Amazon Cognito masque certains champs privés de vos CloudTrail événements avec cette valeur. `HIDDEN_FOR_SECURITY_REASONS` Pour des exemples de champs qu'Amazon Cognito ne renseigne pas pour les événements, consultez [Présentation des événements de connexion Amazon Cognito](#). Amazon Cognito masque uniquement certains champs qui contiennent généralement des informations utilisateur, tels que les mots de passe et les jetons. Amazon Cognito n'effectue aucune détection ni aucun masquage automatiques des informations d'identification personnelle que vous renseignez dans des champs non privés, dans vos demandes d'API.

Groupes d'utilisateurs Amazon Cognito

Amazon Cognito prend en charge la journalisation de toutes les actions répertoriées sur la page des [actions du groupe d'utilisateurs](#) sous forme d'événements dans des fichiers CloudTrail journaux. Amazon Cognito enregistre les événements du groupe d'utilisateurs en CloudTrail tant qu'événements de gestion.

Le eventTypeId champ d'une CloudTrail entrée relative aux groupes d'utilisateurs Amazon Cognito indique si votre application a envoyé la demande à l'API des [groupes d'utilisateurs Amazon Cognito](#) ou à [un point de terminaison qui fournit des ressources pour OpenID Connect, SAML 2.0](#) ou l'interface utilisateur hébergée. Les demandes d'API ont un eventTypeId de AwsApiCall et les demandes de point de terminaison ont un eventTypeId de AwsServiceEvent.

Amazon Cognito enregistre les demandes d'interface utilisateur hébergée suivantes dans votre interface utilisateur hébergée sous forme d'événements dans CloudTrail

Opérations de l'interface utilisateur hébergée dans CloudTrail

Opération	Description
Login_GET , CognitoAuthentication	Un utilisateur consulte ou soumet des informations d'identification à votre Point de terminaison de connexion .
OAuth2_Authorize_GET , Beta_Authorize_GET	Un utilisateur consulte votre Point de terminaison d'autorisation .
OAuth2Response_GET , OAuth2Response_POST	Un utilisateur soumet un jeton du fournisseur d'identité à votre point de terminaison /oauth2/idpresponse .
SAML2Response_POST , Beta_SAML2Response_POST	Un utilisateur soumet une assertion SAML du fournisseur d'identité à votre point de terminaison /saml2/idpresponse .
Login_OIDC_SAML_POST	Un utilisateur saisit un nom d'utilisateur dans votre Point de terminaison de connexion et met en relation à un Identifiant IdP .

Opération	Description
Token_POST , Beta-Token_POST	Un utilisateur soumet un code d'autorisation à votre Point de terminaison de jeton .
Signup_GET , Signup_POST	Un utilisateur soumet des informations d'inscription à votre point de terminaison /signup.
Confirm_GET , Confirm_POST	Un utilisateur soumet un code de confirmation dans l'interface utilisateur hébergée.
ResendCode_POST	Un utilisateur envoie une demande pour renvoyer un code de confirmation dans l'interface utilisateur hébergée.
ForgotPassword_GET , ForgotPassword_POST	Un utilisateur envoie une demande pour réinitialiser son mot de passe à votre point de terminaison /forgotPassword .
ConfirmForgotPassword_GET , ConfirmForgotPassword_POST	Un utilisateur soumet un code à votre point de terminaison /confirmForgotPassword qui confirme sa demande ForgotPassword .
ResetPassword_GET , ResetPassword_POST	Un utilisateur soumet un nouveau mot de passe dans l'interface utilisateur hébergée.
Mfa_GET, Mfa_POST	Un utilisateur soumet un code d'authentification multifactorielle (MFA) dans l'interface utilisateur hébergée.
MfaOption_GET , MfaOption_POST	L'utilisateur choisit sa méthode préférée pour MFA dans l'interface utilisateur hébergée.
MfaRegister_GET , MfaRegister_POST	Un utilisateur soumet un code d'authentification multifactorielle (MFA) dans l'interface utilisateur hébergée lors de l'enregistrement de la MFA.
Logout	Un utilisateur se déconnecte sur votre point de terminaison /logout.

Opération	Description
SAML2Logout_POST	Un utilisateur se déconnecte sur votre point de terminaison <code>/saml2/logout</code> .
Error_GET	Un utilisateur affiche une page d'erreur dans l'interface utilisateur hébergée.
UserInfo_GET , UserInfo_POST	Un utilisateur ou un fournisseur d'identité échange des informations avec votre Point de terminaison UserInfo .
Confirm_With_Link_GET	Un utilisateur soumet une confirmation basée sur un lien envoyé par Amazon Cognito dans un message électronique.
Event_Feedback_GET	Un utilisateur envoie des commentaires à Amazon Cognito à propos d'un événement fonctions de sécurité avancée .

Note

Amazon Cognito enregistre les demandes spécifiques `UserName` à un utilisateur, `UserSub` mais pas dans les CloudTrail journaux. Vous pouvez trouver un utilisateur pour un `UserSub` donné en appelant l'API `ListUsers`, et en utilisant un filtre pour `sub`.

Groupes d'identités Amazon Cognito

Événements de données

Amazon Cognito enregistre les événements Amazon Cognito Identity suivants en tant qu'événements CloudTrail de données. Les [événements de données](#) sont des opérations d'API de plan de données à volume élevé qui CloudTrail ne sont pas enregistrées par défaut. Des frais supplémentaires s'appliquent pour les événements de données.

- [GetCredentialsForIdentity](#)
- [GetId](#)

- [GetOpenIdToken](#)
- [GetOpenIdTokenForDeveloperIdentity](#)
- [UnlinkIdentity](#)

Pour générer des CloudTrail journaux pour ces opérations d'API, vous devez activer les événements de données dans votre historique et choisir des sélecteurs d'événements pour les pools d'identités Cognito. Pour plus d'informations, veuillez consulter [Consignation d'événements de données pour les journaux d'activité](#) dans le Guide de l'utilisateur AWS CloudTrail .

Vous pouvez également ajouter des sélecteurs d'événements de groupes d'identités dans votre journal de suivi à l'aide de la commande CLI suivante.

```
aws cloudtrail put-event-selectors --trail-name <trail name> --advanced-event-selectors
\
"{\
  \"Name\": \"Cognito Selector\", \
  \"FieldSelectors\": [\
    {\
      \"Field\": \"eventCategory\", \
      \"Equals\": [\
        \"Data\" \
      ] \
    }, \
    {\
      \"Field\": \"resources.type\", \
      \"Equals\": [\
        \"AWS::Cognito::IdentityPool\" \
      ] \
    } \
  ] \
}"
```

Événements de gestion

Amazon Cognito enregistre le reste des opérations de l'API des groupes d'identités Amazon Cognito sous forme d'événements de gestion. CloudTrail enregistre les opérations de l'API des événements de gestion par défaut.

Pour obtenir la liste des opérations d'API des groupes d'identités Amazon Cognito auxquelles Amazon Cognito se connecte CloudTrail, consultez le manuel de référence des API des groupes d'identités Amazon [Cognito](#).

Amazon Cognito Sync

Amazon Cognito consigne toutes les opérations d'API de synchronisation Amazon Cognito en tant qu'événements de gestion. Pour obtenir la liste des opérations de l'API Amazon Cognito Sync auxquelles Amazon Cognito se connecte CloudTrail, consultez le manuel Amazon [Cognito Sync API Reference](#).

Présentation des événements de connexion Amazon Cognito

Un suivi peut transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

Rubriques

- [Exemples d' CloudTrail événements pour l'inscription à une interface utilisateur hébergée](#)
- [Exemple CloudTrail d'événement pour une demande SAML](#)
- [Exemples d' CloudTrail événements pour les demandes adressées au point de terminaison du jeton](#)
- [Exemple CloudTrail d'événement pour CreateIdentityPool](#)
- [Exemple CloudTrail d'événement pour GetCredentialsForIdentity](#)
- [Exemple CloudTrail d'événement pour GetId](#)
- [Exemple CloudTrail d'événement pour GetOpenIdToken](#)
- [Exemple CloudTrail d'événement pour GetOpenIdTokenForDeveloperIdentity](#)
- [Exemple CloudTrail d'événement pour UnlinkIdentity](#)

Exemples d' CloudTrail événements pour l'inscription à une interface utilisateur hébergée

Les exemples d' CloudTrail événements suivants illustrent les informations enregistrées par Amazon Cognito lorsqu'un utilisateur s'inscrit via l'interface utilisateur hébergée.

Amazon Cognito enregistre l'événement suivant lorsqu'un nouvel utilisateur accède à la page de connexion de votre application.

```
{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-04-06T05:38:12Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "Login_GET",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
  "errorCode": "",
  "errorMessage": "",
  "additionalEventData":
  {
    "responseParameters":
    {
      "status": 200.0
    },
    "requestParameters":
    {
      "redirect_uri":
      [
        "https://www.amazon.com"
      ],
      "response_type":
      [
        "token"
      ],
      "client_id":
      [
        "1example23456789"
      ]
    }
  },
  "eventID": "382ae09a-151d-4116-8f2b-6ac0a804a38c",
  "readOnly": true,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
```

```
"recipientAccountId": "123456789012",
"serviceEventDetails":
{
  "serviceAccountId": "111122223333"
},
"eventCategory": "Management"
}
```

Amazon Cognito enregistre l'événement suivant lorsqu'un nouvel utilisateur choisit Inscrivez-vous à la page de connexion de votre application.

```
{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-05-05T23:21:43Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "Signup_GET",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData":
  {
    "responseParameters":
    {
      "status": 200
    },
    "requestParameters":
    {
      "response_type":
      [
        "code"
      ],
      "redirect_uri":
      [
        "https://www.amazon.com"
      ],
      "client_id":
      [
```

```

        "1example23456789"
    ]
  },
  "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
  "userPoolId": "us-west-2_aaaaaaaaa"
},
"requestID": "7a63e7c2-b057-4f3d-a171-9d9113264fff",
"eventID": "5e7b27a0-6870-4226-adb4-f86cd51ac5d8",
"readOnly": true,
"eventType": "AwsServiceEvent",
"managementEvent": true,
"recipientAccountId": "123456789012",
"serviceEventDetails":
{
  "serviceAccountId": "111122223333"
},
"eventCategory": "Management"
}

```

Amazon Cognito consigne l'événement suivant quand un nouvel utilisateur choisit un nom d'utilisateur, saisit une adresse e-mail et choisit un mot de passe sur la page de connexion de votre application. Amazon Cognito n'enregistre pas les informations d'identification relatives à l'identité de l'utilisateur dans CloudTrail

```

{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-05-05T23:22:05Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "Signup_POST",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData":
  {
    "responseParameters":
    {
      "status": 302
    }
  }
}

```

```
    },
    "requestParameters":
    {
        "password":
        [
            "HIDDEN_DUE_TO_SECURITY_REASONS"
        ],
        "requiredAttributes[email]":
        [
            "HIDDEN_DUE_TO_SECURITY_REASONS"
        ],
        "response_type":
        [
            "code"
        ],
        "_csrf":
        [
            "HIDDEN_DUE_TO_SECURITY_REASONS"
        ],
        "redirect_uri":
        [
            "https://www.amazon.com"
        ],
        "client_id":
        [
            "1example23456789"
        ],
        "username":
        [
            "HIDDEN_DUE_TO_SECURITY_REASONS"
        ]
    },
    "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
    "userPoolId": "us-west-2_aaaaaaaaa"
},
"requestID": "9ad58dd8-3517-4aa8-96a5-d17a01df9eb4",
"eventID": "c75eb7a5-eb8c-43d1-8331-f4412e756e69",
"readOnly": false,
"eventType": "AwsServiceEvent",
"managementEvent": true,
"recipientAccountId": "123456789012",
"serviceEventDetails":
{
    "serviceAccountId": "111122223333"
```

```
  },
  "eventCategory": "Management"
}
```

Amazon Cognito enregistre l'événement suivant lorsqu'un nouvel utilisateur accède à la page de confirmation de l'utilisateur dans l'interface utilisateur hébergée après son inscription.

```
{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-05-05T23:22:06Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "Confirm_GET",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData":
  {
    "responseParameters":
    {
      "status": 200
    },
    "requestParameters":
    {
      "response_type":
      [
        "code"
      ],
      "redirect_uri":
      [
        "https://www.amazon.com"
      ],
      "client_id":
      [
        "1example23456789"
      ]
    },
    "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
  }
}
```

```
    "userPoolId": "us-west-2_aaaaaaaaaa"
  },
  "requestID": "58a5b170-3127-45bb-88cc-3e652d779e0b",
  "eventID": "7f87291a-6d50-409a-822f-e3a5ec7e60da",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "serviceEventDetails":
  {
    "serviceAccountId": "111122223333"
  },
  "eventCategory": "Management"
}
```

Amazon Cognito enregistre l'événement suivant lorsque, sur la page de confirmation utilisateur de l'interface utilisateur hébergée, un utilisateur saisit un code qu'Amazon Cognito lui a envoyé par e-mail.

```
{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-05-05T23:23:32Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "Confirm_POST",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData":
  {
    "responseParameters":
    {
      "status": 302
    },
    "requestParameters":
    {
      "confirm":
      [
```



```
    ""
  ],
  "deliveryMedium":
  [
    "EMAIL"
  ],
  "sub":
  [
    "704b1e47-34fe-40e9-8c41-504997494531"
  ],
  "code":
  [
    "HIDDEN_DUE_TO_SECURITY_REASONS"
  ],
  "destination":
  [
    "HIDDEN_DUE_TO_SECURITY_REASONS"
  ],
  "response_type":
  [
    "code"
  ],
  "_csrf":
  [
    "HIDDEN_DUE_TO_SECURITY_REASONS"
  ],
  "cognitoAsfData":
  [
    "HIDDEN_DUE_TO_SECURITY_REASONS"
  ],
  "redirect_uri":
  [
    "https://www.amazon.com"
  ],
  "client_id":
  [
    "1example23456789"
  ],
  "username":
  [
    "HIDDEN_DUE_TO_SECURITY_REASONS"
  ]
},
"userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
```

```

    "userPoolId": "us-west-2_aaaaaaaaaa"
  },
  "requestID": "9764300a-ed35-4f87-8a0f-b18b3fe2b11e",
  "eventID": "e24ac6e5-2f70-4c6e-ad4e-2f08a547bb36",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "serviceEventDetails":
  {
    "serviceAccountId": "111122223333"
  },
  "eventCategory": "Management"
}

```

Exemple CloudTrail d'événement pour une demande SAML

Amazon Cognito enregistre l'événement suivant lorsqu'un utilisateur qui s'est authentifié auprès de votre fournisseur d'identité SAML soumet l'assertion SAML à votre point de terminaison `/saml2/idpresponse`.

```

{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-05-06T00:50:57Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "SAML2Response_POST",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData":
  {
    "responseParameters":
    {
      "status": 302
    },
    "requestParameters":
    {

```

```

    "RelayState":
      [
        "HIDDEN_DUE_TO_SECURITY_REASONS"
      ],
    "SAMLResponse":
      [
        "HIDDEN_DUE_TO_SECURITY_REASONS"
      ]
  },
  "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
  "userPoolId": "us-west-2_aaaaaaaaa"
},
"requestID": "4f6f15d1-c370-4a57-87f0-aac4817803f7",
"eventID": "9824b50f-d9d1-4fb8-a2c1-6aa78ca5902a",
"readOnly": false,
"eventType": "AwsServiceEvent",
"managementEvent": true,
"recipientAccountId": "625647942648",
"serviceEventDetails":
{
  "serviceAccountId": "111122223333"
},
"eventCategory": "Management"
}

```

Exemples d' CloudTrail événements pour les demandes adressées au point de terminaison du jeton

Voici des exemples d'événements de demandes au [Point de terminaison de jeton](#).

Amazon Cognito enregistre l'événement suivant lorsqu'un utilisateur qui s'est authentifié et a reçu un code d'autorisation soumet le code à votre point de terminaison /oauth2/token.

```

{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-05-12T22:12:30Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "Token_POST",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",

```

```
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
"requestParameters": null,
"responseElements": null,
"additionalEventData":
{
  "responseParameters":
  {
    "status": 200
  },
  "requestParameters":
  {
    "code":
    [
      "HIDDEN_DUE_TO_SECURITY_REASONS"
    ],
    "grant_type":
    [
      "authorization_code"
    ],
    "redirect_uri":
    [
      "https://www.amazon.com"
    ],
    "client_id":
    [
      "1example23456789"
    ]
  },
  "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
  "userPoolId": "us-west-2_aaaaaaaaa"
},
"requestID": "f257f752-cc14-4c52-ad5b-152a46915238",
"eventID": "0bd1586d-cd3e-4d7a-abaf-fd8bfc3912fd",
"readOnly": false,
"eventType": "AwsServiceEvent",
"managementEvent": true,
"recipientAccountId": "123456789012",
"serviceEventDetails":
{
  "serviceAccountId": "111122223333"
},
"eventCategory": "Management"
}
```

Amazon Cognito consigne l'événement suivant quand votre système backend soumet une demande `client_credentials` de jeton d'accès à votre point de terminaison `/oauth2/token`.

```
{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-05-12T21:07:05Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "Token_POST",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData":
  {
    "responseParameters":
    {
      "status": 200
    },
    "requestParameters":
    {
      "grant_type":
      [
        "client_credentials"
      ],
      "client_id":
      [
        "1example23456789"
      ]
    },
    "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
    "userPoolId": "us-west-2_aaaaaaaaa"
  },
  "requestID": "4f871256-6825-488a-871b-c2d9f55caff2",
  "eventID": "473e5cbc-a5b3-4578-9ad6-3dfdc8a6d34",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
```

```
"serviceEventDetails":
{
  "serviceAccountId": "111122223333"
},
"eventCategory": "Management"
}
```

Amazon Cognito enregistre l'événement suivant lorsque votre application échange un jeton d'actualisation pour un nouvel identifiant et un nouveau jeton d'accès avec votre point de terminaison `/oauth2/token`.

```
{
  "eventVersion": "1.08",
  "userIdentity":
  {
    "accountId": "123456789012"
  },
  "eventTime": "2022-05-12T22:16:40Z",
  "eventSource": "cognito-idp.amazonaws.com",
  "eventName": "Token_POST",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData":
  {
    "responseParameters":
    {
      "status": 200
    },
    "requestParameters":
    {
      "refresh_token":
      [
        "HIDDEN_DUE_TO_SECURITY_REASONS"
      ],
      "grant_type":
      [
        "refresh_token"
      ],
      "client_id":
      [
```

```

        "1example23456789"
    ]
},
"userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
"userPoolId": "us-west-2_aaaaaaaaa"
},
"requestID": "2829f0c6-a3a9-4584-b046-11756dfe8a81",
"eventID": "12bd3464-59c7-44fa-b8ff-67e1cf092018",
"readOnly": false,
"eventType": "AwsServiceEvent",
"managementEvent": true,
"recipientAccountId": "123456789012",
"serviceEventDetails":
{
    "serviceAccountId": "111122223333"
},
"eventCategory": "Management"
}

```

Exemple CloudTrail d'événement pour CreateIdentityPool

L'exemple suivant illustre l'entrée de journal d'une demande exécutée pour l'action `CreateIdentityPool`. Cette demande a été effectuée par un utilisateur IAM prénommé Alice.

```

{
  "eventVersion": "1.03",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "['EXAMPLE_KEY_ID']",
    "userName": "Alice"
  },
  "eventTime": "2016-01-07T02:04:30Z",
  "eventSource": "cognito-identity.amazonaws.com",
  "eventName": "CreateIdentityPool",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "USER_AGENT",
  "requestParameters": {
    "identityPoolName": "TestPool",
    "allowUnauthenticatedIdentities": true,

```

```

    "supportedLoginProviders": {
      "graph.facebook.com": "0000000000000000"
    }
  },
  "responseElements": {
    "identityPoolName": "TestPool",
    "identityPoolId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
    "allowUnauthenticatedIdentities": true,
    "supportedLoginProviders": {
      "graph.facebook.com": "0000000000000000"
    }
  },
  "requestID": "15cc73a1-0780-460c-91e8-e12ef034e116",
  "eventID": "f1d47f93-c708-495b-bff1-cb935a6064b2",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}

```

Exemple CloudTrail d'événement pour GetCredentialsForIdentity

L'exemple suivant illustre l'entrée de journal d'une demande exécutée pour l'action `GetCredentialsForIdentity`.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown"
  },
  "eventTime": "2023-01-19T16:55:08Z",
  "eventSource": "cognito-identity.amazonaws.com",
  "eventName": "GetCredentialsForIdentity",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.4",
  "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
command/cognito-identity.get-credentials-for-identity",
  "requestParameters": {
    "logins": {
      "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
"HIDDEN_DUE_TO_SECURITY_REASONS"
    },
    "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
  },
  "responseElements": {

```



```

    },
    "responseElements": {
      "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "dc28def9-07c8-460a-a8f3-3816229e6664",
    "eventID": "c5c459d9-40ec-41fd-8f6b-57865d5a9975",
    "readOnly": false,
    "resources": [{
      "accountId": "111122223333",
      "type": "AWS::Cognito::IdentityPool",
      "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
  }
}

```

Exemple CloudTrail d'événement pour GetOpenIdToken

L'exemple suivant illustre l'entrée de journal d'une demande exécutée pour l'action GetOpenIdToken.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown"
  },
  "eventTime": "2023-01-19T16:55:08Z",
  "eventSource": "cognito-identity.amazonaws.com",
  "eventName": "GetOpenIdToken",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.4",
  "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
command/cognito-identity.get-open-id-token",
  "requestParameters": {
    "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
    "logins": {
      "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
"HIDDEN_DUE_TO_SECURITY_REASONS"
    }
  },
}

```

```

"responseElements": {
  "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
},
"requestID": "a506ba18-10d7-4fdb-9548-a8187b2e38bb",
"eventID": "19ffc1a6-6ed8-4580-a4e1-3062c5ce6457",
"readOnly": false,
"resources": [{
  "accountId": "111122223333",
  "type": "AWS::Cognito::IdentityPool",
  "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
}],
"eventType": "AwsApiCall",
"managementEvent": false,
"recipientAccountId": "111122223333",
"eventCategory": "Data"
}

```

Exemple CloudTrail d'événement pour GetOpenIdTokenForDeveloperIdentity

L'exemple suivant illustre l'entrée de journal d'une demande exécutée pour l'action `GetOpenIdTokenForDeveloperIdentity`.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROEXAMPLE:johns-AssumedRoleSession",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/johns-AssumedRoleSession",
    "accountId": "111122223333",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROEXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "attributes": {
        "creationDate": "2023-01-19T16:53:14Z",
        "mfaAuthenticated": "false"
      }
    }
  }
}

```

```

    }
  },
  "eventTime": "2023-01-19T16:55:08Z",
  "eventSource": "cognito-identity.amazonaws.com",
  "eventName": "GetOpenIdTokenForDeveloperIdentity",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "27.0.3.154",
  "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
command/cognito-identity.get-open-id-token-for-developer-identity",
  "requestParameters": {
    "tokenDuration": 900,
    "identityPoolId": "us-east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE",
    "logins": {
      "JohnsDeveloperProvider": "HIDDEN_DUE_TO_SECURITY_REASONS"
    }
  },
  "responseElements": {
    "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
  },
  "requestID": "b807df87-57e7-4dd6-b90c-b06f46a61c21",
  "eventID": "f26fed91-3340-4d70-91ae-cdf555547b76",
  "readOnly": false,
  "resources": [{
    "accountId": "111122223333",
    "type": "AWS::Cognito::IdentityPool",
    "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
  }],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "111122223333",
  "eventCategory": "Data"
}

```

Exemple CloudTrail d'événement pour UnlinkIdentity

L'exemple suivant illustre l'entrée de journal d'une demande exécutée pour l'action `UnlinkIdentity`.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown"
  }
}

```

```
  },
  "eventTime": "2023-01-19T16:55:08Z",
  "eventSource": "cognito-identity.amazonaws.com",
  "eventName": "UnlinkIdentity",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.4",
  "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
command/cognito-identity.unlink-identity",
  "requestParameters": {
    "logins": {
      "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaa":
"HIDDEN_DUE_TO_SECURITY_REASONS"
    },
    "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
    "loginsToRemove": ["cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaa"]
  },
  "responseElements": null,
  "requestID": "99c2c8e2-9c29-416f-bb17-b650a5cbada9",
  "eventID": "d8e26126-202a-43c2-b458-3f225efaedc7",
  "readOnly": false,
  "resources": [{
    "accountId": "111122223333",
    "type": "AWS::Cognito::IdentityPool",
    "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
  }],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "111122223333",
  "eventCategory": "Data"
}
```

Analyse des CloudTrail événements Amazon Cognito avec Amazon CloudWatch Logs Insights

Vous pouvez rechercher et analyser vos CloudTrail événements Amazon Cognito avec Amazon CloudWatch Logs Insights. Lorsque vous configurez votre parcours pour envoyer des événements à CloudWatch Logs, il CloudTrail envoie uniquement les événements correspondant à vos paramètres de suivi.

Pour interroger ou rechercher vos CloudTrail événements Amazon Cognito, dans la CloudTrail console, assurez-vous de sélectionner l'option Gestion des événements dans vos paramètres de

suivi afin de pouvoir surveiller les opérations de gestion effectuées sur vos AWS ressources. Lorsque vous souhaitez identifier des erreurs, une activité inhabituelle ou un comportement inhabituel de l'utilisateur dans votre compte, vous pouvez éventuellement sélectionner l'option Événements Insights dans les paramètres de votre journal d'activité.

Exemples de requêtes Amazon Cognito

Vous pouvez utiliser les requêtes suivantes dans la CloudWatch console Amazon.

Requêtes générales

Rechercher les 25 derniers événements ajoutés au journal.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com"
```

Consultez la liste des 25 derniers événements de journal ajoutés qui incluent des exceptions.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com" and @message like /Exception/
```

Exception et requêtes d'erreur

Recherchez les 25 derniers événements de journal ajoutés avec un code d'erreur `NotAuthorizedException` avec le groupe d'utilisateurs Amazon Cognito `sub`.

```
fields @timestamp, additionalEventData.sub as user | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com" and errorCode=
  "NotAuthorizedException"
```

Recherchez le nombre d'enregistrements avec la source `IPAddress` et l'événement `eventName` correspondant.

```
filter eventSource = "cognito-idp.amazonaws.com"
| stats count(*) by sourceIPAddress, eventName
```

Recherchez les 25 premières adresses IP qui ont déclenché une erreur `NotAuthorizedException`.

```
filter eventSource = "cognito-idp.amazonaws.com" and errorCode=
  "NotAuthorizedException"
```

```
| stats count(*) as count by sourceIPAddress, eventName  
| sort count desc | limit 25
```

Recherchez les 25 premières adresses IP qui ont appelé l'API `ForgotPassword`.

```
filter eventSource = "cognito-idp.amazonaws.com" and eventName = 'ForgotPassword'  
| stats count(*) as count by sourceIPAddress  
| sort count desc | limit 25
```

Validation de conformité pour Amazon Cognito

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon Cognito dans le cadre de plusieurs programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d'autres.

Pour obtenir la liste des AWS services concernés par des programmes de conformité spécifiques, voir [AWS Services concernés par programme de conformitéAWS](#) . Pour obtenir des renseignements généraux, consultez [Programmes de conformitéAWS](#) .

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 d'Amazon Cognito est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que la législation et la réglementation applicables. AWS fournit les ressources suivantes pour faciliter le respect de la conformité :

- [Guides de démarrage rapide de la sécurité et de la conformité](#) : ces guides de déploiement traitent de considérations architecturales et indiquent comment déployer des environnements de référence axés sur la sécurité et la conformité sur AWS.
- Livre blanc [sur l'architecture pour la sécurité et la conformité HIPAA — Ce livre blanc](#) décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- AWS ressources de <https://aws.amazon.com/compliance/resources/> de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [Évaluation des ressources à l'aide des règles](#) énoncées dans le guide du AWS Config développeur : AWS Config évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.

- [AWS Security Hub](#)— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Résilience dans Amazon Cognito

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

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

Rubriques

- [Considérations sur les données régionales](#)

Considérations sur les données régionales

Les groupes d'utilisateurs Amazon Cognito sont chacun créés dans une AWS région et ils stockent les données de profil utilisateur uniquement dans cette région. Les groupes d'utilisateurs peuvent envoyer des données utilisateur vers une autre AWS région, en fonction de la configuration des fonctionnalités facultatives.

- Si la valeur par défaut `no-reply@verificationemail.com` de l'adresse e-mail est utilisée pour le routage de la vérification des courriels pour les groupes d'utilisateurs Amazon Cognito, les courriels sont acheminés via la même région que le groupe d'utilisateurs associé.
- Si une adresse e-mail différente est utilisée pour configurer Amazon Simple Email Service (Amazon SES) avec des groupes d'utilisateurs Amazon Cognito, cette adresse e-mail est acheminée AWS via la région associée à l'adresse e-mail dans Amazon SES.
- Les messages SMS provenant des groupes d'utilisateurs Amazon Cognito sont acheminés via la même région Amazon SNS, sauf indication contraire dans [Configuration de la vérification par courriel ou par téléphone](#).

- Si des analyses Amazon Pinpoint sont utilisées avec des groupes d'utilisateurs Amazon Cognito, les données d'événement sont acheminées vers la région USA Est (Virginie du Nord).

Note

Amazon Pinpoint est disponible dans plusieurs AWS régions d'Amérique du Nord, d'Europe, d'Asie et d'Océanie. Les régions Amazon Pinpoint incluent l'API Amazon Pinpoint. Si une région Amazon Pinpoint est prise en charge par Amazon Cognito, Amazon Cognito envoie des événements à des projets Amazon Pinpoint dans la même région Amazon Pinpoint. Si une région n'est pas prise en charge par Amazon Pinpoint, Amazon Cognito prend en charge uniquement l'envoi d'événements dans la région us-east-1. Pour obtenir des informations détaillées sur une région Amazon Pinpoint, consultez [Points de terminaison et quotas Amazon Pinpoint](#) et [Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito](#).

Sécurité de l'infrastructure dans Amazon Cognito

En tant que service géré, Amazon Cognito est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Amazon Cognito via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Configuration et analyse des vulnérabilités dans les groupes d'utilisateurs Amazon Cognito

AWS gère les tâches de sécurité de base telles que l'application de correctifs au système d'exploitation client (OS) et aux bases de données, la configuration du pare-feu et la reprise après sinistre. Ces procédures ont été vérifiées et certifiées par les tiers appropriés. Pour plus de détails, consultez les ressources suivantes :

- [Validation de conformité pour Amazon Cognito](#)
- [Modèle de responsabilité partagée](#)

AWS politiques gérées pour Amazon Cognito

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 le client IAM](#) qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent les cas d'utilisation courants et sont disponibles dans votre AWS compte. Pour plus d'informations sur les politiques AWS gérées, voir les [politiques AWS gérées](#) dans le guide de l'utilisateur IAM.

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 la liste des politiques de fonctions professionnelles et leurs descriptions, consultez la page [politiques gérées par AWS pour les fonctions de tâche](#) dans le Guide de l'utilisateur IAM.

Plusieurs politiques sont disponibles via la console IAM que vous pouvez utiliser pour accorder l'accès à Amazon Cognito :

- `AmazonCognitoPowerUser` : autorisations d'accès et de gestion de tous les aspects des groupes d'identités et d'utilisateurs. Pour consulter les autorisations associées à cette politique, consultez [AmazonCognitoPowerUser](#).
- `AmazonCognitoReadOnly` : autorisations d'accès en lecture seule à vos groupes d'identités et à vos groupes d'utilisateurs. Pour consulter les autorisations associées à cette politique, consultez [AmazonCognitoReadOnly](#).
- `AmazonCognitoDeveloperAuthenticatedIdentities` : autorisations d'intégration de votre système d'authentification avec Amazon Cognito. Pour consulter les autorisations associées à cette politique, consultez [AmazonCognitoDeveloperAuthenticatedIdentities](#).

Ces politiques étant gérées par l'équipe Amazon Cognito, vos utilisateurs continuent de bénéficier du même niveau d'accès, même quand de nouvelles API sont ajoutées.

Note

Lorsque vous créez une nouvelle réserve d'identités, vous pouvez créer automatiquement de nouveaux rôles pour l'accès des utilisateurs authentifiés et invités. L'administrateur qui crée votre réserve d'identités avec de nouveaux rôles IAM doit également disposer des autorisations IAM nécessaires pour créer des rôles.

Les pools d'identités dotés d'un accès invité non authentifié appliquent une politique AWS gérée supplémentaire `AmazonCognitoUnAuthedIdentitiesSessionPolicy`, en tant que [stratégie de session](#) aux utilisateurs non authentifiés. Cette politique AWS gérée n'est pas destinée à un usage administratif. Elle limite plutôt la portée des autorisations que vous pouvez appliquer aux utilisateurs invités dans le [flux d'authentification amélioré](#) des réserves d'identités. Pour plus d'informations, consultez [Rôles IAM](#).

Amazon Cognito met à jour les politiques gérées AWS

Consultez les informations relatives aux mises à jour des politiques AWS gérées pour Amazon Cognito depuis que ce service a commencé à suivre ces modifications. Pour obtenir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS de la page [Historique du document](#) d'Amazon Cognito.

Modification	Description	Date
AmazonCognitoUnAuthenticatedIdentitiesSessionPolicy : nouvelle politique	Ajout d'une politique AWS gérée pour la limitation des privilèges des utilisateurs invités dans les pools d'identités.	14 juillet 2023
AmazonCognitoPowerUser et AmazonCognitoReadOnly : mises à jour des politiques existantes	<p>Ajout de nouvelles autorisations pour permettre aux utilisateurs expérimentés de consulter et de gérer les associations d'ACL AWS WAF Web aux groupes d'utilisateurs Amazon Cognito.</p> <p>De nouvelles autorisations ont été ajoutées pour permettre aux utilisateurs en lecture seule de consulter les associations d'ACL AWS WAF Web avec les groupes d'utilisateurs Amazon Cognito.</p>	19 juillet 2022
AmazonCognitoPowerUser - mise à jour d'une politique existante	Ajout d'une nouvelle autorisation permettant à Amazon Cognito d'appeler les opérations PutIdentityPolicy et ListConf	17 novembre 2021

Modification	Description	Date
	<p>gurationSets d'Amazon Simple Notification Service.</p> <p>Cette modification permet aux groupes d'utilisateurs Amazon Cognito de mettre à jour les politiques d'autorisation d'envoi Amazon SES et d'appliquer des jeux de configuration Amazon SES lorsque vous configurez l'envoi de courriels dans votre groupe d'utilisateurs.</p>	
<p>AmazonCognitoPowerUser - mise à jour d'une politique existante</p>	<p>Ajout d'une nouvelle autorisation permettant à Amazon Cognito d'appeler l'opération GetSMSSandboxAccountStatus d'Amazon Simple Notification Service.</p> <p>Cette modification permet aux groupes d'utilisateurs Amazon Cognito de décider si vous devez quitter l'environnement de test (sandbox) Amazon Simple Notification Service pour envoyer des messages à tous les utilisateurs finaux via des groupes d'utilisateurs.</p>	<p>1er juin 2021</p>
<p>Amzon Cognito a commencé à suivre les modifications</p>	<p>Amazon Cognito a commencé à suivre les modifications apportées à ses politiques AWS gérées.</p>	<p>1er mars 2021</p>

Identification des ressources Amazon Cognito

Une identification est une étiquette de métadonnées que vous AWS attribuez à une ressource AWS. Chaque balise se compose d'une clé et d'une valeur. Pour les balises que vous affectez, vous définissez la clé et la valeur. Par exemple, vous pouvez définir la clé sur `stage` et la valeur pour une ressource sur `test`.

Les balises vous permettent d'effectuer les actions suivantes :

- Identifier et organiser vos ressources AWS. De nombreux services AWS prennent en charge l'identification. Vous pouvez donc attribuer la même identification à des ressources de différents services. Cela vous aide à indiquer quelles ressources sont liées. Par exemple, vous pouvez affecter à un groupe d'utilisateurs Amazon Cognito la même étiquette que celle que vous affectez à une table Amazon DynamoDB.
- Suivre vos coûts AWS. Vous pouvez activer ces identifications sur le tableau de bord AWS Billing and Cost Management. AWS utilise des identifications d'allocation des coûts pour catégoriser vos coûts et vous fournir un rapport mensuel d'allocation des coûts. Pour de plus amples informations, veuillez consulter [Utilisation des identifications d'allocation des coûts](#) dans le Guide de l'utilisateur AWS Billing.
- Contrôler l'accès à vos ressources en fonction des balises qui leur sont affectées. Vous pouvez contrôler l'accès en spécifiant des clés et des valeurs d'identification dans les conditions d'une politique AWS Identity and Access Management (IAM). Par exemple, vous ne pouvez autoriser un utilisateur à mettre à jour un groupe d'utilisateurs que si le groupe d'utilisateurs possède une balise `owner` dont la valeur est le nom de cet utilisateur. Pour de plus amples informations, veuillez consulter [Contrôle de l'accès à l'aide d'identifications](#) dans le Guide de l'utilisateur IAM.

Vous pouvez utiliser la AWS Command Line Interface ou l'API Amazon Cognito pour ajouter, modifier ou supprimer des identifications pour des groupes d'utilisateurs et des groupes d'identités. Vous pouvez également gérer les identifications pour les groupes d'utilisateurs en utilisant la console Amazon Cognito.

Pour accéder à des conseils sur l'utilisation des identifications, consultez l'article [politiques d'identification AWS](#) sur le blog AWS Answers.

Les sections suivantes fournissent des informations supplémentaires sur les étiquettes pour Amazon Cognito.

Ressources prises en charge dans Amazon Cognito

Les ressources suivantes dans Amazon Cognito prennent en charge l'étiquetage :

- Groupes d'utilisateurs
- Réserves d'identités

Restrictions liées aux étiquettes

Les restrictions de base suivantes s'appliquent aux identifications sur les ressources Amazon Cognito :

- Nombre maximum d'étiquettes que vous pouvez attribuer à une ressource – 50
- Longueur de clé maximale – 128 caractères Unicode
- Longueur de valeur maximale – 256 caractères Unicode
- Caractères valides pour les clés et valeurs : a-z, A-Z, 0-9, espace et les caractères suivants : _ . : / = + - @
- Les clés et les valeurs sont sensibles à la casse.
- N'utilisez pas `aws:` comme préfixe pour les clés ; seul AWS peut utiliser cette valeur.

Gestion des identifications à l'aide de la console Amazon Cognito

Vous pouvez utiliser la console Amazon Cognito pour gérer les étiquettes affectées à vos groupes d'utilisateurs.

Pour ajouter des balises à un groupe d'utilisateurs

1. Accédez à la [console Amazon Cognito](#). Si vous y êtes invité, saisissez vos AWS Informations d'identification.
2. Choisissez Groupes d'utilisateurs.
3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez un groupe d'utilisateurs](#).
4. Choisissez l'onglet User pool properties (Propriétés du groupe d'utilisateurs) et localisez Tags (Identifications).

5. Choisissez Ajouter des identifications pour ajouter votre première identification. Si vous avez déjà attribué des identifications à ce groupe d'utilisateurs, dans Gérer les identifications, choisissez Ajouter une autre.
6. Spécifiez les valeurs de Clé d'identification et Valeur d'identification.
7. Pour chaque identification supplémentaire à ajouter, choisissez Ajouter une autre identification.
8. Une fois que vous avez ajouté des identifications, choisissez Enregistrer les modifications.

Sur la page Gérer les identifications, vous pouvez également modifier les clés et les valeurs des identifications existantes. Pour supprimer une identification, choisissez Supprimer.

Exemples AWS CLI

La AWS CLI propose des commandes qui vous permettent de gérer les identifications que vous affectez à vos groupes d'utilisateurs et d'identités Amazon Cognito.

Affectation de balises

Utilisez les commandes suivantes pour affecter des balises à vos groupes d'utilisateurs et groupes d'identités existants.

Exemple Commande **tag-resource** pour les groupes d'utilisateurs

Pour affecter des identifications à un groupe d'utilisateurs, utilisez [tag-resource](#) dans l'ensemble de commandes `cognito-idp` :

```
$ aws cognito-idp tag-resource \  
> --resource-arn user-pool-arn \  
> --tags Stage=Test
```

Cette commande comprend les paramètres suivants :

- `resource-arn` – Amazon Resource Name (ARN) du groupe d'utilisateurs auquel vous appliquez les étiquettes. Pour rechercher l'ARN, choisissez le groupe d'utilisateurs dans la console Amazon Cognito, puis examinez la valeur ARN du groupe sous l'onglet Paramètres généraux.
- `tags` – Paires clé-valeur des identifications au format *key=value*.

Pour affecter plusieurs balises à la fois, spécifiez-les dans une liste séparée par des virgules :


```
$ aws cognito-idp tag-resource \  
> --resource-arn user-pool-arn \  
> --tags Stage=Test, CostCenter=80432, Owner=SysEng
```

Exemple Commande **tag-resource** pour les groupes d'identités

Pour affecter des balises à un groupe d'identités, utilisez [tag-resource](#) dans l'ensemble de commandes cognito-identity :

```
$ aws cognito-identity tag-resource \  
> --resource-arn identity-pool-arn \  
> --tags Stage=Test
```

Cette commande comprend les paramètres suivants :

- **resource-arn** – Amazon Resource Name (ARN) du groupe d'identités auquel vous appliquez les étiquettes. Pour rechercher l'ARN, choisissez le groupe d'identités dans la console Amazon Cognito, puis Modifier le groupe d'identités. Ensuite, dans ID du groupe d'identités, choisissez Afficher l'ARN.
- **tags** – Paires clé-valeur des identifications au format *key=value*.

Pour affecter plusieurs balises à la fois, spécifiez-les dans une liste séparée par des virgules :

```
$ aws cognito-identity tag-resource \  
> --resource-arn identity-pool-arn \  
> --tags Stage=Test, CostCenter=80432, Owner=SysEng
```

Affichage des balises

Utilisez les commandes suivantes pour afficher les balises que vous avez attribuées à vos groupes d'utilisateurs et groupes d'identités.

Exemple Commande **list-tags-for-resource** pour les groupes d'utilisateurs

Pour afficher les balises qui sont affectées à un groupe d'utilisateurs, utilisez [list-tags-for-resource](#) dans l'ensemble de commandes cognito-idp :

```
$ aws cognito-idp list-tags-for-resource --resource-arn user-pool-arn
```

Exemple Commande **list-tags-for-resource** pour les groupes d'identités

Pour afficher les balises qui sont affectées à un groupe d'identités, utilisez [list-tags-for-resource](#) dans l'ensemble de commandes `cognito-identity` :

```
$ aws cognito-identity list-tags-for-resource --resource-arn identity-pool-arn
```

Suppression de balises

Utilisez les commandes suivantes pour supprimer des balises de vos groupes d'utilisateurs et groupes d'identités.

Exemple Commande **untag-resource** pour les groupes d'utilisateurs

Pour supprimer des identifications d'un groupe d'utilisateurs, utilisez [untag-resource](#) dans le jeu de commandes `cognito-idp` :

```
$ aws cognito-idp untag-resource \  
> --resource-arn user-pool-arn \  
> --tag-keys Stage CostCenter Owner
```

Pour `--tag-keys`, indiquez une ou plusieurs clés d'identification. N'incluez pas les valeurs d'identification. Clés séparées par des espaces.

Exemple Commande **untag-resource** pour les groupes d'identités

Pour supprimer des balises d'un groupe d'identités, utilisez [untag-resource](#) dans l'ensemble de commandes `cognito-identity` :

```
$ aws cognito-identity untag-resource \  
> --resource-arn identity-pool-arn \  
> --tag-keys Stage CostCenter Owner
```

Pour `--tag-keys`, indiquez une ou plusieurs clés d'identification. N'incluez pas les valeurs d'identification.

⚠ Important

Après avoir supprimé un groupe d'utilisateurs ou d'identités, les identifications associées au groupe supprimé peuvent toujours apparaître dans la console ou les appels d'API jusqu'à 30 jours après la suppression.

Application de balises au moment de créer des ressources

Utilisez les commandes suivantes pour affecter des balises au moment de créer un groupe d'utilisateurs ou un groupe d'identités.

Exemple Commande **create-user-pool** avec des identifications

Lorsque vous créez un groupe d'utilisateurs à l'aide de la commande [create-user-pool](#), vous pouvez spécifier des balises avec le paramètre `--user-pool-tags` :

```
$ aws cognito-idp create-user-pool \  
> --pool-name user-pool-name \  
> --user-pool-tags Stage=Test, CostCenter=80432, Owner=SysEng
```

Les paires clé-valeur pour les identifications doivent être au format *key=value*. Si vous ajoutez plusieurs identifications, spécifiez-les dans une liste d'éléments séparés par une virgule

Exemple Commande **create-identity-pool** avec des identifications

Lorsque vous créez un groupe d'identités à l'aide de la commande [create-identity-pool](#), vous pouvez indiquer des balises avec le paramètre `--identity-pool-tags` :

```
$ aws cognito-identity create-identity-pool \  
> --identity-pool-name identity-pool-name \  
> --allow-unauthenticated-identities \  
> --identity-pool-tags Stage=Test, CostCenter=80432, Owner=SysEng
```

Les paires clé-valeur des identifications doivent être au format *key=value*. Si vous ajoutez plusieurs identifications, spécifiez-les dans une liste d'éléments séparés par une virgule.

Gestion des identifications à l'aide de l'API Amazon Cognito

Vous pouvez utiliser les actions suivantes dans l'API Amazon Cognito pour gérer les étiquettes de vos groupes d'utilisateurs et d'identités.

Actions d'API pour les identifications de groupe d'utilisateurs

Utilisez les actions d'API suivantes pour affecter, afficher et supprimer des balises pour les groupes d'utilisateurs.

- [TagResource](#)
- [ListTagsForResource](#)
- [UntagResource](#)
- [CreateUserPool](#)

Actions d'API pour les identifications de groupe d'identités

Utilisez les actions d'API suivantes pour affecter, afficher et supprimer des balises pour les groupes d'identités.

- [TagResource](#)
- [ListTagsForResource](#)
- [UntagResource](#)
- [CreateIdentityPool](#)

Quotas dans Amazon Cognito

Amazon Cognito dispose de quotas par défaut, anciennement appelés limites, pour le nombre maximal d'opérations que vous pouvez effectuer dans votre compte. Amazon Cognito présente également des quotas pour le nombre maximal et la taille maximale des ressources Amazon Cognito.

Chaque quota Amazon Cognito représente un volume maximum de demandes une par une Région AWS. Compte AWS Par exemple, vos applications peuvent effectuer des demandes d'API jusqu'à atteindre le taux de quota par défaut (RPS) pour les opérations `UserAuthentication` s'appliquant à tous vos groupes d'utilisateurs de la région USA Est (Virginie du Nord). Vos applications en Asie-Pacifique (Tokyo) peuvent générer le même volume de demandes pour tous vos groupes d'utilisateurs dans leur propre région. AWS ne peut accéder à une demande d'augmentation de quota que dans une seule région à la fois. Le fait d'obtenir une augmentation de quota dans la région USA Est (Virginie du Nord) n'a aucune incidence sur le taux de demandes maximal qui est le vôtre dans la région Asie-Pacifique (Tokyo).

Rubriques

- [Comprendre les quotas de taux de demandes d'API](#)
- [Gestion des quotas de taux de demandes d'API](#)
- [Catégories d'opérations d'API des groupes d'utilisateurs Amazon Cognito et quotas de taux de demandes](#)
- [Référence d'API des groupes d'identités Amazon Cognito \(identités fédérées\)](#)
- [Quotas relatifs au nombre et à la taille des ressources](#)

Comprendre les quotas de taux de demandes d'API

Catégorisation des quotas

Amazon Cognito impose un taux de requêtes maximal pour les opérations d'API. Pour en savoir plus sur les opérations d'API mises à disposition par Amazon Cognito, consultez [Références de points de terminaison et d'API Amazon Cognito](#). Pour les groupes d'utilisateurs, ces opérations sont groupées en catégories de cas d'utilisation courants comme `UserAuthentication` ou `UserCreation`. Pour obtenir la liste des opérations d'API du pool d'utilisateurs par catégorie, consultez [Catégories d'opérations d'API des groupes d'utilisateurs Amazon Cognito et quotas de taux de demandes](#).

Dans la [console Service Quotas](#), vous pouvez suivre l'utilisation de vos quotas par catégories, groupes d'utilisateurs et groupes d'identités. Si le taux de demandes de vos groupes d'utilisateurs Amazon Cognito dépasse ou dépasse un quota, vous pouvez acheter de la capacité supplémentaire. Vous pouvez suivre l'utilisation des quotas de votre groupe d'utilisateurs par catégorie et les augmentations des quotas d'achat dans la [console Service Quotas](#).

Les quotas d'opérations correspondent par définition au nombre maximal de demandes par seconde (RPS) pour toutes les opérations au sein d'une catégorie. Le service des groupes d'utilisateurs Amazon Cognito applique des quotas à toutes les opérations dans chaque catégorie. Par exemple, la catégorie `UserCreation` inclut quatre opérations : `SignUp`, `ConfirmSignUp`, `AdminCreateUser` et `AdminConfirmSignUp`. Un quota combiné de 50 RPS lui est alloué. Si plusieurs opérations ont lieu en même temps, chaque opération dans cette catégorie peut appeler jusqu'à 50 RPS séparément ou combinées.

Note

Les quotas de catégories ne s'appliquent qu'aux groupes d'utilisateurs. Amazon Cognito applique chaque quota de groupe d'identités à une seule opération. Pour les quotas de taux de demandes par catégorie et par opération, AWS mesure le taux agrégé de toutes les demandes provenant de tous les groupes d'utilisateurs ou groupes d'identités de votre région Compte AWS dans une même région.

Opérations API des groupes d'utilisateurs Amazon Cognito avec traitement des taux de demandes spéciaux

Les quotas d'opérations sont mesurés et appliqués pour l'ensemble des demandes combinées au niveau de la catégorie, excepté pour les opérations `AdminRespondToAuthChallenge` et `RespondToAuthChallenge`, où des règles de traitement spéciales sont appliquées.

La `UserAuthentication` catégorie inclut quatre opérations dans l'API des groupes d'utilisateurs Amazon Cognito : `AdminInitiateAuth`, `InitiateAuthAdminRespondToAuthChallenge`, et `RespondToAuthChallenge`. En outre, l'authentification des utilisateurs dans l'interface utilisateur hébergée contribue à ce quota. Les opérations `InitiateAuth` et `AdminInitiateAuth` sont mesurées et appliquées par quota de catégorie. Les opérations `RespondToAuthChallenge` et `AdminRespondToAuthChallenge` correspondantes sont soumises à un quota distinct qui est trois fois la limite de catégorie `UserAuthentication`. Ce quota élevé répond aux multiples

défis d'authentification définis dans vos applications. Le quota est suffisant pour couvrir la grande majorité des cas d'utilisation. Une fois que votre application a répondu jusqu'à trois à des problèmes d'authentification, les demandes supplémentaires sont prises en compte dans le quota de `UserAuthentication` catégories. [L'authentification multifactorielle \(MFA\)](#), [l'authentification des appareils et l'authentification personnalisée](#) sont autant d'exemples de demandes de défi que vous pourriez intégrer à votre groupe d'utilisateurs.

Par exemple, si votre quota pour `UserAuthentication` cette catégorie est de 80 RPS, vous pouvez appeler `RespondToAuthChallenge` ou `AdminRespondToAuthChallenge` à un débit allant jusqu'à 240 RPS (3 x 80 RPS). Si votre groupe d'utilisateurs vous invite à lancer quatre séries de défis par authentification et que 70 utilisateurs se connectent par seconde, le total `RespondToAuthChallenge` est de 280 RPS (70 x 4), soit 40 RPS de plus que le quota. Les 40 RPS supplémentaires sont ajoutés à 70 appels `InitiateAuth`, ce qui porte l'utilisation totale de la catégorie `UserAuthentication` à 110 RPS (40 + 70). Comme cette valeur dépasse le quota de catégorie défini à 80 RPS par 30 RPS, Amazon Cognito limite les demandes provenant de votre application.

Monthly active users (Utilisateurs actifs mensuels)

Lorsqu'Amazon Cognito calcule la facturation du groupe d'utilisateurs, il vous facture un tarif pour chaque utilisateur actif mensuel (MAU). Tenez compte de votre nombre de MAU actuel et prévu dans le cadre de la planification des demandes d'augmentation de quota. Un utilisateur est comptabilisé comme un MAU si, au cours d'un mois civil, une opération d'identité est associée à cet utilisateur. Les activités qui rendent un utilisateur actif incluent les suivantes.

- Inscription ou création administrative d'un utilisateur
- Connexion
- Déconnexion
- Confirmation du compte d'utilisateur ou vérification des attributs
- Réinitialisation du mot de passe
- Modification des attributs utilisateur, de l'appartenance au groupe ou des préférences MFA
- Requêtes d'attributs détaillés d'un utilisateur
- Activation, désactivation ou suppression d'un utilisateur

Note

La catégorie Query detailed attributes of a user inclut le fonctionnement de l'API [AdminGetUser](#), mais pas [ListUsers](#). Une user-by-user requête détaillée auprès d'un large groupe d'utilisateurs peut avoir un impact significatif sur votre AWS facture. Pour éviter des frais supplémentaires, collectez les données utilisateur ListUsers ou stockez-les dans une base de données externe.

Gestion des quotas de taux de demandes d'API

Identifier les besoins en matière de quota

Important

Si vous augmentez les quotas Amazon Cognito pour des catégories telles que `UserAuthentication`, ou `UserCreationAccountRecovery`, vous devrez peut-être augmenter les quotas pour d'autres catégories. Services AWS Par exemple, les messages envoyés par Amazon Cognito avec Amazon Simple Notification Service (Amazon SNS) et Amazon Simple Email Service (Amazon SES) peuvent échouer si les quotas de taux de demande sont insuffisants dans ces services.

Pour calculer les besoins en matière de quota, déterminez le nombre d'utilisateurs actifs qui interagiront avec votre application au cours d'une période donnée. Par exemple, si votre application s'attend à ce qu'en moyenne 1 million d'utilisateurs actifs se connectent par période de 8 heures, vous devriez pouvoir authentifier en moyenne 35 utilisateurs par seconde.

En outre, si vous supposez que la session moyenne d'un utilisateur dure deux heures et que vous configurez les jetons pour qu'ils expirent après une heure, chaque utilisateur doit rafraîchir ses jetons une fois au cours de sa session. Ensuite, le quota moyen requis pour la catégorie `UserAuthentication` afin de supporter cette charge est de 70 RPS.

Si vous supposez un peak-to-average ratio de 3:1 en tenant compte de la variation de la fréquence de connexion des utilisateurs au cours de la période de huit heures, vous avez besoin du `UserAuthentication` quota souhaité de 200 RPS.

Note

Si vous appelez plusieurs opérations pour chaque action utilisateur, vous devez additionner les taux d'appels d'opération individuels au niveau de la catégorie.

Optimisation des taux de demandes pour les limites de quotas

Étant donné que l'augmentation des limites de débit des API augmente les coûts de votre AWS facture, pensez à ajuster votre modèle d'utilisation avant de demander une augmentation de quota. Voici quelques exemples d'architecture d'applications qui optimisent les taux de demandes.

Faire une nouvelle tentative après une période d'interruption

Vous pouvez détecter l'erreur à chaque appel de l'API, puis faire une nouvelle tentative après une période d'interruption. Vous pouvez ajuster l'algorithme d'interruption en fonction des besoins de l'activité et de la charge. Les kits SDK Amazon intègrent une logique de nouvelle tentative. Pour plus d'informations, consultez la section [Outils sur lesquels vous pouvez vous appuyer AWS](#).

Utiliser une base de données externe pour les attributs fréquemment mis à jour

Si votre application nécessite plusieurs appels à un groupe d'utilisateurs pour lire ou écrire des attributs personnalisés, utilisez un stockage externe. Vous pouvez utiliser votre base de données préférée pour stocker des attributs personnalisés, ou utiliser une couche de cache pour charger un profil utilisateur lors de la connexion. Vous pouvez référencer ce profil à partir du cache si nécessaire au lieu de recharger le profil utilisateur à partir d'un groupe d'utilisateurs.

Validez les jetons Web JSON (JWT) côté client

Les applications doivent valider les jetons JWT avant de les approuver. Vous pouvez vérifier la signature et la validité des jetons côté client sans envoyer de demandes d'API à un groupe d'utilisateurs. Une fois le jeton validé, vous pouvez approuver les revendications qu'il contient et utiliser celles-ci au lieu de faire plus d'appels d'API `getUser`. Pour plus d'informations, consultez la page [Jeton JWT \(JSON Web Token\)](#).

Limiter le trafic vers votre application web avec une salle d'attente

Si vous vous attendez à ce que le trafic provienne d'un grand nombre d'utilisateurs qui se connectent pendant un événement limité dans le temps, comme passer un examen ou assister à un événement en direct, vous pouvez optimiser le trafic des demandes à l'aide de mécanismes d'auto-limitation. Vous pouvez, par exemple, mettre en place une salle d'attente où les utilisateurs

peuvent patienter jusqu'à ce qu'une session soit disponible, ce qui vous permet de traiter les demandes lorsque vous disposez d'une capacité disponible. Consultez [Solution AWS Virtual Waiting Room](#) pour une architecture de référence d'une salle d'attente.

JWT en cache

Réutilisez les jetons d'accès jusqu'à leur expiration. Pour un exemple de framework avec mise en cache de jetons dans une API Gateway, consultez [Jetons de mise en cache](#). Au lieu de générer des demandes d'API pour demander des informations sur les utilisateurs, mettez en cache les jetons d'identification jusqu'à leur expiration et lisez les attributs utilisateur depuis le cache.

Pour plus d'informations sur l'utilisation des taux de demandes d'API dans AWS, consultez [la section Gestion et surveillance de la limitation des API dans vos](#) charges de travail. Pour plus d'informations sur l'optimisation des opérations Amazon Cognito qui ajoutent des coûts à votre AWS facture, consultez. [Gestion des coûts](#)

Suivre l'usage des quotas

Amazon Cognito génère `CallCount` des `ThrottleCount` statistiques dans Amazon CloudWatch pour chaque catégorie d'opérations d'API au niveau du compte. Vous pouvez utiliser la métrique `CallCount` pour suivre le nombre total d'appels effectués par des clients en rapport avec une catégorie. Vous pouvez utiliser la métrique `ThrottleCount` pour suivre le nombre total d'appels limités en rapport avec une catégorie. Vous pouvez utiliser les métriques `CallCount` et `ThrottleCount` avec la statistique `Sum` pour compter le nombre total d'appels dans une catégorie. Pour plus d'informations, consultez la section [Mesures CloudWatch d'utilisation](#).

Lors de la surveillance des quotas de service, l'utilisation correspond au pourcentage d'un quota de service utilisé. Par exemple, si la valeur du quota est de 200 ressources et que 150 ressources sont utilisées, l'utilisation est de 75 %. L'usage est le nombre de ressources ou d'opérations utilisées pour un quota de service.

Suivi de l'utilisation par le biais de CloudWatch métriques

Vous pouvez suivre et collecter les statistiques d'utilisation des groupes d'utilisateurs Amazon Cognito avec CloudWatch. Le CloudWatch tableau de bord affiche des statistiques sur tout Service AWS ce que vous utilisez. Vous pouvez ainsi créer des alarmes métriques pour vous avertir ou modifier une ressource spécifique que vous surveillez. CloudWatch Pour plus d'informations sur CloudWatch les statistiques, consultez la section [Suivi de vos statistiques CloudWatch d'utilisation](#).

Suivi de l'utilisation via les métriques de Service Quotas

Les groupes d'utilisateurs Amazon Cognito sont intégrés à Service Quotas, une interface de console permettant d'afficher et de gérer l'utilisation de vos quotas de service. Dans la console Service Quotas, vous pouvez rechercher la valeur d'un quota spécifique, consulter les informations de surveillance, demander une augmentation de quota ou configurer des CloudWatch alarmes. Une fois que votre compte a été actif pendant un certain temps, vous pouvez consulter un graphique de l'utilisation de vos ressources.

La colonne Valeur de quota appliquée au niveau du compte dans la console Service Quotas pour les groupes d'[utilisateurs Amazon Cognito](#) et les groupes d'[identités Amazon Cognito affiche votre quota actuel](#). La colonne Utilisation indique votre taux actuel d'utilisation des quotas. Les quotas ajustables des groupes d'utilisateurs requests-per-second (RPS) Amazon Cognito indiquent leur utilisation actuelle. La console Service Quotas peut également vous faire accéder aux CloudWatch métriques pour examiner de plus près une métrique de quota sélectionnée. Pour plus d'informations sur l'affichage des quotas dans la console Service Quotas, consultez [Affichage de Service Quotas](#).

Suivez les utilisateurs actifs mensuels (MAU)

Le nombre d'utilisateurs actifs (MAU) mensuels de votre groupe d'utilisateurs fournit des données importantes pour planifier l'augmentation des quotas de taux de demandes. Vous pouvez comparer le taux de vos demandes d'API au nombre d'utilisateurs que vous avez actifs au cours d'une période donnée. Grâce à ces connaissances, vous pouvez calculer l'impact d'une augmentation du nombre d'utilisateurs actifs de vos applications sur les quotas de votre modèle d'utilisation. Par exemple, imaginez que vos applications combinées dans l'ouest des États-Unis (Oregon) aient généré 2 millions d'utilisateurs actifs en un mois et que votre `UserAuthentication` catégorie subisse des erreurs de régulation occasionnelles au quota par défaut de 120 demandes par seconde (RPS). Le mois précédent, avant le succès de votre campagne publicitaire, vous disposiez d'un million de MAU et vos applications n'ont jamais dépassé 80 RPS. Si vous prévoyez un pic similaire à la suite d'un nouveau spot télévisé, vous pourriez acheter 40 RPS supplémentaires pour répondre aux besoins du prochain million d'utilisateurs avec un quota ajusté de 160 RPS.

Pour revoir votre MAU

Accédez à la [AWS Billing console](#) et consultez une facture récente. Dans la section Frais par service, vous pouvez filtrer sur Cognito pour afficher le détail de vos MAU pour cette période de facturation.

Demande d'augmentation de quota

Amazon Cognito impose un quota pour le nombre maximum d'opérations par seconde que vous pouvez effectuer dans vos groupes d'utilisateurs et vos groupes d'identités dans chacun d'eux.

Région AWS Vous pouvez acheter une augmentation des quotas ajustables de taux de demandes d'API pour les groupes d'utilisateurs Amazon Cognito. Vérifiez votre quota actuel et achetez une augmentation depuis la console Service Quotas ou via les opérations de l'API Service Quotas `ListAWSDefaultServiceQuotas` et `RequestServiceQuotaIncrease`.

- Pour acheter une augmentation de quota à l'aide de la console Service Quotas, consultez la section [Demander une augmentation de quota d'API](#) dans le Guide de l'utilisateur de Service Quotas.
- AWS vise à traiter les demandes d'augmentation de quota dans un délai de 10 jours. Cependant, plusieurs facteurs peuvent faire en sorte que le délai de traitement des demandes dépasse 10 jours. Certaines demandes, par exemple, peuvent nécessiter qu'Amazon Cognito fournisse une capacité matérielle supplémentaire, et les augmentations saisonnières du volume de demandes peuvent entraîner des retards.
- Si le quota n'est pas encore disponible dans Service Quotas, utilisez le [Formulaire d'augmentation de limite de service](#).

Important

Seuls des quotas ajustables peuvent être augmentés. Vous devez acheter une capacité de quota accrue. Pour connaître les tarifs en fonction de l'augmentation des quotas, consultez la tarification d'[Amazon Cognito](#).

Catégories d'opérations d'API des groupes d'utilisateurs Amazon Cognito et quotas de taux de demandes

Sachant qu'il existe dans Amazon Cognito des classes d'opérations d'API qui se chevauchent avec des [modèles d'autorisation différents](#), chaque opération appartient à une catégorie. Chaque catégorie dispose de son propre quota mis en commun pour toutes les opérations d'API membres, à l'échelle de tous les groupes d'utilisateurs d'une Région AWS de votre compte. Vous pouvez uniquement demander une augmentation pour des quotas de catégorie ajustables. Pour plus d'informations, consultez [Demande d'augmentation de quota](#). Les ajustements de quota s'appliquent aux groupes d'utilisateurs de votre compte d'une même région. Amazon Cognito limite les opérations de certaines catégories³ à 5 demandes par seconde (RPS), par groupe d'utilisateurs. Le quota par défaut (RPS) s'applique également à tous les groupes d'utilisateurs d'un Compte AWS.

Note

Pour chaque catégorie, le quota est mesuré en utilisateurs actifs mensuels. Les Comptes AWS présentant moins de deux millions d'utilisateurs actifs mensuels peuvent fonctionner dans les limites du quota par défaut. Si vous avez moins d'un million d'unités MAU et qu'Amazon Cognito limite les demandes, pensez à optimiser votre application. Pour plus d'informations, consultez [Optimisation des taux de demandes pour les limites de quotas](#).

Les quotas d'opérations de catégorie s'appliquent à l'ensemble des utilisateurs de tous les groupes d'utilisateurs d'une même Région AWS. Amazon Cognito gère également un quota pour le nombre de demandes que votre application peut générer pour un même utilisateur. Vous devez limiter le nombre de demandes d'API par utilisateur comme indiqué dans le tableau suivant.

Quotas de taux de demandes par utilisateur pour les groupes d'utilisateurs Amazon Cognito

Opération	Nombre d'opérations par utilisateur à la seconde
Lecture d'un profil utilisateur Exemples : <code>GetUser</code> , <code>GetDevice</code>	10
Écriture d'un profil utilisateur Exemples : <code>UpdateUserAttributes</code> , <code>SetUserSettings</code>	10

Vous devez limiter le nombre de demandes d'API par catégorie comme indiqué dans le tableau suivant.

Quotas de taux de demandes par catégorie pour les groupes d'utilisateurs Amazon Cognito

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserAuthentication	Opérations qui authentifient un	120	Oui

Catégorie	Description	Quota par défaut (RPS)	Ajustable
<ul style="list-style-type: none"> • InitiateAuth • Actualisation du jeton avec InitiateAuth ou Point de terminaison de jeton • RespondToAuthChallenge¹ • AdminInitiateAuth • AdminRespondToAuthChallenge¹ • Connexion à l'interface utilisateur hébergée et MFA dans les autorisations autorisation-code ou implicites² 	<p>utilisateur (lors de la connexion).</p> <p>Ces opérations sont sujettes à Opérations API des groupes d'utilisateurs Amazon Cognito avec traitement des taux de demandes spéciaux .</p>		
<p>UserCreation</p> <ul style="list-style-type: none"> • SignUp • ConfirmSignUp • AdminCreateUser • AdminConfirmSignUp 	<p>Opérations qui créent ou confirment un utilisateur local Amazon Cognito. Il s'agit d'un utilisateur créé et vérifié directement par vos groupes d'utilisateurs Amazon Cognito.</p>	50	Oui

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserFederation	Opérations qui soumettent une réponse du fournisseur d'identité à un point de terminaison de fédération de groupes d'utilisateurs. Les opérations OIDC ou de fournisseur social qui aboutissent à un jeton de fournisseur d'identité, ainsi que toutes les demandes SAML, contribuent à ce quota.	25	Oui

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserAccountRecovery <ul style="list-style-type: none"> • ChangePassword • ConfirmForgotPassword • ForgotPassword • AdminResetUserPassword • AdminSetUserPassword • RespondToAuthChallenge¹ • AdminRespondToAuthChallenge¹ • Réinitialisation du mot de passe de l'interface utilisateur hébergée 	Opérations qui récupèrent le compte d'un utilisateur, ou qui modifient ou mettent à jour le mot de passe d'un utilisateur.	30	Non
UserRead <ul style="list-style-type: none"> • AdminGetUser • GetUser 	Opérations qui extraient un utilisateur de vos groupes d'utilisateurs.	120	Oui

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserUpdate <ul style="list-style-type: none"> • AdminAddUserToGroup • AdminDeleteUserAttributes • AdminUpdateUserAttributes • AdminDeleteUser • AdminDisableUser • AdminEnableUser • AdminLinkProviderForUser • AdminDisableProviderForUser • VerifyUserAttribute • DeleteUser • DeleteUserAttributes • UpdateUserAttributes • AdminUserGlobalSignOut • GlobalSignOut • AdminRemoveUserFromGroup 	Opérations que vous utilisez pour gérer les utilisateurs et les attributs utilisateur	25	Non
UserToken <ul style="list-style-type: none"> • RevokeToken 	Opérations de la gestion des jetons	120	Oui

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserResourceRead <ul style="list-style-type: none">• AdminGetDevice• AdminListGroupsWithUser• AdminListDevices• GetDevice• ListDevices• GetUserAttributeVerificationCode• ResendConfirmationCode• AdminListUserAuthEvents	Opérations qui extraient d'Amazon Cognito des informations sur des ressources utilisateur telles qu'un appareil ou un groupe.	50	Oui

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserResourceUpdate	Opérations qui mettent à jour les informations sur les ressources d'un utilisateur, telles qu'un appareil mémorisé ou une appartenance à un groupe.	25	Non
	<ul style="list-style-type: none"> • AdminForgetDevice • AdminUpdateAuthEventFeedback • AdminSetUserReferenceMFAP • AdminSetUserSettings • AdminUpdateDeviceStatus • UpdateDeviceStatus • UpdateAuthEventFeedback • ConfirmDevice • SetUserReferenceMFAP • SetUserSettings • VerifySoftwareToken • AssociateSoftwareToken • ForgetDevice 		

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserList <ul style="list-style-type: none">• ListUsers• ListUsersInGroup	Opérations qui renvoient une liste d'utilisateurs.	30	Non
UserPoolRead <ul style="list-style-type: none">• DescribeUserPool• ListUserPools	Opérations qui lisent vos groupes d'utilisateurs.	15	Non
UserPoolUpdate <ul style="list-style-type: none">• CreateUserPool• UpdateUserPool• DeleteUserPool	Opérations qui créent, mettent à jour et suppriment vos groupes d'utilisateurs.	15	Non

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserPoolResourceRead	Opérations qui récupèrent des informations sur les ressources, telles que des groupes ou des serveurs de ressources, à partir d'un groupe d'utilisateurs. ³	20	Non
	<ul style="list-style-type: none"> • DescribeIdentityProvider • DescribeResourceServer • DescribeUserImportJob • DescribeUserPoolDomain • GetCSVHeader • GetGroup • GetSigningCertificate • GetIdentityProviderByIdentifier • GetUserPoolMfaConfig • ListGroups • ListIdentityProviders • ListResourceServers • ListTagsForResource • ListUserImportJobs • DescribeRiskConfiguration • GetUICustomization 		

Catégorie	Description	Quota par défaut (RPS)	Ajustable
UserPoolResourceUpdate <ul style="list-style-type: none"> • AddCustomAttribute • CreateGroup • CreateIdentityProvider • CreateResourceServer • CreateUserImportJob • CreateUserPoolDomain • DeleteGroup • DeleteIdentityProvider • DeleteResourceServer • DeleteUserPoolDomain • SetUserPoolMfaConfig • StartUserImportJob • StopUserImportJob • UpdateGroup • UpdateIdentityProvider • UpdateResourceServer 	Opérations qui modifient des ressources, telles que des groupes ou des serveurs de ressources, dans un groupe d'utilisateurs. ³	15	Non

Catégorie	Description	Quota par défaut (RPS)	Ajustable
<ul style="list-style-type: none"> • UpdateUse rPoolDomain • SetRiskConfigurati on • SetUICustomization • TagResource • UntagResource 			
UserPoolC lientRead <ul style="list-style-type: none"> • DescribeU serPoolClient • ListUserPoolClients 	Opérations qui récupèrent des informations sur vos clients de groupe d'utilisateurs. ³	15	Non
UserPoolC lientUpdate <ul style="list-style-type: none"> • CreateUserPoolClie nt • DeleteUserPoolClie nt • UpdateUse rPoolClient 	Opérations qui créent, mettent à jour et suppriment des clients de votre groupe d'utilisateurs. ³	15	Non
ClientAut hentication Demandes de type d'autoris ation client_cr edentials au point de terminaison du jeton.	Opérations qui génèrent des informati ons d'identification à utiliser pour autoriser les demandes machine-to-machine	150	Non

¹ Une `RespondToAuthChallenge` ou une `AdminRespondToAuthChallenge` réponse avec un `ChallengeName` de `NEW_PASSWORD_REQUIRED` compte pour la `UserAccountRecovery` catégorie. Toutes les autres réponses au défi sont prises en compte dans `UserAuthentication` cette catégorie.

² Chaque opération d'interface utilisateur hébergée pendant la connexion contribue à une demande au quota. Par exemple, un utilisateur qui se connecte et fournit un code MFA envoie deux demandes. L'échange de jetons sous forme de subventions avec code d'autorisation est soumis à une allocation de quota supplémentaire au même taux que votre quota dans la catégorie. `UserAuthentication`

³ Toute opération individuelle de cette catégorie comporte une contrainte qui empêche l'opération d'être appelée à un débit supérieur à 5 RPS pour un seul groupe d'utilisateurs.

Référence d'API des groupes d'identités Amazon Cognito (identités fédérées)

Opération	Description	Quota par défaut (RPS) ¹	Ajustable	Éligibilité à une augmentation de quotas
<code>GetId</code>	Récupérez un identifiant d'identité à partir d'un groupe d'identités.	25	Oui	Contactez l'équipe de votre compte.
<code>GetOpenIdToken</code>	Récupérez un jeton OpenID à partir d'un groupe d'identités dans le flux de travail classique.	200	Oui	Contactez l'équipe de votre compte.
<code>GetCredentialsForIdentity</code>	Récupérez les AWS informations d'identité	200	Oui	Contactez l'équipe de votre compte.

Opération	Description	Quota par défaut (RPS) ¹	Ajustable	Éligibilité à une augmentation de quotas
	lication d'un pool d'identités dans le flux de travail amélioré.			
GetOpenIdTokenForDeveloperIdentity	Récupérez un jeton OpenID à partir d'un groupe d'identités dans le flux de travail des développeurs.	50	Oui	Contactez l'équipe de votre compte.
ListIdentities	Récupérez la liste des ID d'identité figurant dans une réserve d'identités.	5	Oui	Contactez l'équipe de votre compte.
DeleteIdentities	Supprimez une ou plusieurs identités enregistrées d'une réserve d'identités.	10	Oui	Contactez l'équipe de votre compte.
TagResource	Appliquez une balise à une réserve d'identités.	5	Oui	Contactez l'équipe de votre compte.

Opération	Description	Quota par défaut (RPS) ¹	Ajustable	Éligibilité à une augmentation de quotas
UntagResource	Supprimez une balise d'une réserve d'identités.	5	Oui	Contactez l'équipe de votre compte.
ListTagsForResource	Affichez la liste des balises appliquées à une réserve d'identités.	10	Oui	Contactez l'équipe de votre compte.

¹ Le quota par défaut est le quota de taux de demandes minimum pour les pools d'identités de chacun Région AWS de vos groupes Compte AWS. Votre quota RPS peut être supérieur dans certaines régions.

Quotas relatifs au nombre et à la taille des ressources

Les quotas de ressources correspondent au nombre ou à la taille maximum des ressources, des champs de saisie, de la durée et d'autres fonctionnalités diverses d'Amazon Cognito.

Vous pouvez demander un ajustement pour certains quotas de ressources dans la console Service Quotas ou à partir d'un [formulaire d'augmentation des limites de service](#). Pour demander un quota à l'aide la console Service Quotas, consultez [Demande d'augmentation de quota](#) dans le Guide de l'utilisateur Service Quotas. Si le quota n'est pas encore disponible dans Service Quotas, utilisez le [Formulaire d'augmentation de limite de service](#).

Note

Les quotas de ressources au Compte AWS niveau, tels que les groupes d'utilisateurs par région, s'appliquent aux ressources Amazon Cognito de chaque région. Région AWS Par exemple, vous pouvez avoir 1 000 groupes d'utilisateurs dans la région USA Est (Virginie du Nord) et 1 000 autres dans la région Europe (Stockholm).

Les tableaux suivants indiquent les quotas de ressources par défaut et s'ils sont ajustables ou non.

Quotas de ressources de groupes d'utilisateurs Amazon Cognito

Ressource	Quota	Ajustable	Quota maximal
Clients d'applications par groupe d'utilisateurs	1 000	Oui	10 000
Groupes d'utilisateurs par région	1 000	Oui	10 000
Fournisseurs d'identité par groupe d'utilisateurs	300	Oui	1 000
Serveurs de ressources par groupe d'utilisateurs	25	Oui	300
Utilisateurs par groupe d'utilisateurs	40 000 000	Oui	Contactez l'équipe de votre compte.
Total des changements combinés dans le déclencheur Lambda avant la génération du jeton ¹	5 000	Oui	Contactez l'équipe de votre compte.
Attributs personnalisés par groupe d'utilisateurs	50	Non	N/A
Nombre maximal de caractères par attribut	2 048 octets	Non	N/A

Ressource	Quota	Ajustable	Quota maximal
Nom maximum de caractères d'un nom d'attribut personnalisé	20	Non	N/A
Caractères de mot de passe minimaux requis dans la politique	6–99	Non	N/A
Messages électroniques envoyés quotidiennement par Compte AWS ²	50	Non	N/A
Nombre maximal de caractères par objet d'e-mail	140	Non	N/A
Nombre maximum de caractères d'un e-mail	20 000	Non	N/A
Nombre maximal de caractères dans le message de vérification par SMS	140	Non	N/A
Nombre maximal de caractères dans le mot de passe	256	Non	N/A
Nombre maximum de caractères d'un nom de fournisseur d'identité	32	Non	N/A

Ressource	Quota	Ajustable	Quota maximal
Nombre maximal d'identifiants par fournisseur d'identité	50	Non	N/A
Identités liées à un utilisateur	5	Non	N/A
Nombre maximal d'URL de rappel par client d'application	100	Non	N/A
Nombre maximal d'URL de déconnexion par client d'application	100	Non	N/A
Portée du serveur de ressources	100	Non	N/A
Portées par client d'application	50	Non	N/A
Domaines personnalisés par compte	4	Non	N/A
Groupes auxquels chaque utilisateur peut appartenir	100	Non	N/A
Groupes par groupe d'utilisateurs	10 000	Non	N/A

¹ Ce quota peut être atteint dans les jetons provenant d'un [Déclencheur Lambda avant génération de jeton](#). Le nombre de demandes existantes et ajoutées plus celui des champs d'application des jetons d'accès et d'identité doivent être inférieurs ou égaux à ce quota. Les demandes et les champs d'application supprimés ne contribuent pas à ce quota.

² Ce quota s'applique uniquement si vous utilisez la fonction de messagerie par défaut pour un groupe d'utilisateurs Amazon Cognito. Pour permettre un volume de remise d'e-mails plus élevé, configurez votre groupe d'utilisateurs afin qu'il utilise votre configuration d'e-mail Amazon SES. Pour plus d'informations, consultez [Paramètres d'e-mail pour les groupes d'utilisateurs Amazon Cognito](#).

Paramètres de validité de session de groupes d'utilisateurs Amazon Cognito

Jeton	Quota
Jeton d'identification	5 minutes – 1 jour
Jeton d'actualisation	1 heure – 3 650 jours
Jeton d'accès	5 minutes – 1 jour
Cookie de session d'interface utilisateur hébergée	1 heure
Jeton d'authentification de session	3 minutes à 15 minutes

Quotas de ressources de sécurité de code des groupes d'utilisateurs Amazon Cognito (non ajustables)

Ressource	Quota
Période de validité du code de confirmation d'inscription	24 heures
Période de validité du code de vérification des attributs utilisateur	24 heures
Période de validité du code d'authentification MFA	3 à 15 minutes
Période de validité du code d'oubli de mot de passe	1 heure

Ressource	Quota
Nombre maximum de demandes ConfirmForgotPassword et ForgotPassword par utilisateur et par heure ¹	5–20
Nombre maximum de demandes ResendConfirmationCode par utilisateur et par heure	5
Nombre maximum de demandes ConfirmSignUp par utilisateur et par heure	15
Nombre maximum de demandes ChangePassword par utilisateur et par heure	5
Nombre maximum de demandes GetUserAttributeVerificationCode par utilisateur et par heure	5
Nombre maximum de demandes VerifyUserAttribute par utilisateur et par heure	15

¹ Amazon Cognito évalue les facteurs de risque dans la demande de mise à jour des mots de passe et attribue un quota lié au niveau de risque évalué. Pour plus d'informations, consultez [Comportement en cas d'oubli de mot de passe](#).

Quotas de ressources de tâches d'importation d'utilisateurs des groupes d'utilisateurs Amazon Cognito

Ressource	Quota	Ajustable	Quota maximal
Tâches d'importation d'utilisateurs par groupe d'utilisateurs	1 000	Oui	Contactez l'équipe de votre compte.
Nombre maximal de caractères par ligne	16,000	Non	N/A

Ressource	Quota	Ajustable	Quota maximal
CSV d'importation d'utilisateurs			
Taille maximale du fichier CSV	100 Mo	Non	N/A
Nombre maximal d'utilisateurs par fichier CSV	500 000	Non	N/A

Référence d'API des groupes d'identités Amazon Cognito (identités fédérées)

Ressource	Quota	Ajustable	Quota maximal
Groupes d'identités par compte	1 000	Oui	N/A
Nombre maximum de fournisseurs de groupes d'utilisateurs Amazon Cognito par groupe d'identités	50	Oui	1 000
Nombre maximal de caractères pour le nom du groupe d'identités	128 bytes	Non	N/A
Nombre maximal de caractères pour le nom du fournisseur de connexion	2 048 octets	Non	N/A
Identités par groupe d'identités	Illimité	Non	N/A

Ressource	Quota	Ajustable	Quota maximal
Nombre maximum de fournisseurs d'identité pour lesquels des mappages de rôles peuvent être spécifiés	10	Non	N/A
Nombre maximum de résultats renvoyés par appel d'API de liste ou de recherche	60	Non	N/A
Règles de contrôle d'accès en fonction du rôle (RBAC)	25	Non	N/A

Quotas de ressources Amazon Cognito Sync

Ressource	Quota	Ajustable	Quota maximal
Jeux de données par identité	20	Oui	Contactez l'équipe de votre compte.
Enregistrements par jeu de données	1,024	Oui	Contactez l'équipe de votre compte.
Taille maximum d'un seul jeu de données	1 Mo	Oui	Contactez l'équipe de votre compte.
Nombre maximum de caractères d'un nom de jeu de données	128 bytes	Non	N/A
Temps d'attente minimal pour une publication groupée	24 heures	Non	N/A

Ressource	Quota	Ajustable	Quota maximal
après une requête réussie			

Références de points de terminaison et d'API Amazon Cognito

Les informations de référence suivantes décrivent les points de terminaison de service pour chaque fonctionnalité d'Amazon Cognito. Les groupes d'utilisateurs Amazon Cognito proposent les options suivantes : les [points de terminaison des groupes d'utilisateurs](#) avec un domaine de groupe d'utilisateurs et l'[API des groupes d'utilisateurs](#). Pour obtenir le détail des classes d'opérations d'API avec l'API des groupes d'utilisateurs Amazon Cognito, consultez [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#).

Pour obtenir la liste des points de terminaison de service pour l'API des groupes d'utilisateurs par Région AWS, consultez [Points de terminaison de service](#) dans Référence générale d'AWS.

Rubriques

- [Référence de l'interface utilisateur hébergée et des points de terminaison de la fédération des groupes d'utilisateurs](#)
- [Référence d'API des groupes d'utilisateurs Amazon Cognito](#)
- [Référence d'API des groupes d'identités Amazon Cognito \(identités fédérées\)](#)
- [Référence d'API Amazon Cognito Sync](#)

Référence de l'interface utilisateur hébergée et des points de terminaison de la fédération des groupes d'utilisateurs

Amazon Cognito active les pages web publiques indiquées ici lorsque vous attribuez un domaine à votre groupe d'utilisateurs. Votre domaine sert de point d'accès central à tous vos clients d'application. Ils incluent l'interface utilisateur hébergée, où vos utilisateurs peuvent s'inscrire, se connecter ([Point de terminaison de connexion](#)) et se déconnecter ([Point de terminaison de déconnexion](#)). Pour plus d'information sur ces ressources, consultez [Configuration et utilisation de l'interface utilisateur hébergée par Amazon Cognito et des points de terminaison de fédération](#).

Ces pages incluent également les ressources Web publiques qui permettent à votre groupe d'utilisateurs de communiquer avec des fournisseurs d'identité SAML, OpenID Connect (OIDC) et OAuth 2.0 tiers (). IdPs Pour connecter un utilisateur avec un fournisseur d'identité fédérée, vos utilisateurs doivent lancer une demande auprès du [Point de terminaison de connexion](#) de l'interface

utilisateur hébergée ou du [Point de terminaison d'autorisation](#) OIDC. Le point de terminaison d'autorisation redirige vos utilisateurs vers votre interface utilisateur hébergée ou vers la page de connexion de votre fournisseur d'identité.

Votre application peut également connecter des utilisateurs locaux avec l'[API des groupes d'utilisateurs Amazon Cognito](#). Un utilisateur local existe exclusivement dans l'annuaire de votre groupe d'utilisateurs sans fédération via un fournisseur d'identité externe.

Outre l'interface utilisateur hébergée et les points de terminaison de fédération, Amazon Cognito s'intègre aux SDK pour Android, iOS JavaScript, etc. Les kits SDK fournissent des outils permettant d'effectuer des opérations d'API de groupes d'utilisateurs avec les points de terminaison des services d'API Amazon Cognito. Pour plus d'informations sur les points de terminaison de service, consultez [Amazon Cognito Identity endpoints and quotas](#) (Points de terminaison d'identité et quotas Amazon Cognito).

Warning

N'épinglez pas les certificats TLS (entité finale ou intermédiaire) pour les domaines Amazon Cognito. AWS gère tous les certificats pour tous les points de terminaison et domaines de préfixes de votre groupe d'utilisateurs. Les autorités de certification (CA) figurant dans la chaîne d'approbation qui prend en charge les certificats Amazon Cognito effectuent une rotation et se renouvellent de manière dynamique. Lorsque vous épinglez votre application à un certificat intermédiaire ou secondaire, votre application peut échouer sans préavis lors de la AWS rotation des certificats.

Épinglez plutôt votre application à tous les [certificats racine Amazon](#) disponibles. Pour plus d'informations, consultez les bonnes pratiques et recommandations fournies dans [Épinglage de certificat](#), dans le Guide de l'utilisateur AWS Certificate Manager .

Rubriques

- [Référence des points de terminaison d'interface utilisateur hébergée](#)
- [Référence des points de terminaison de fédération OAuth 2.0, OpenID Connect et SAML 2.0](#)
- [Octrois OAuth 2.0](#)
- [Utilisation de PKCE pour octroyer des codes d'autorisation aux groupes d'utilisateurs Amazon Cognito](#)
- [Réponses aux erreurs de l'interface utilisateur hébergée et de fédération](#)

Référence des points de terminaison d'interface utilisateur hébergée

Amazon Cognito active les points de terminaison d'interface utilisateur hébergée dans cette section lorsque vous ajoutez un domaine à votre groupe d'utilisateurs. Il s'agit de pages Web sur lesquelles vos utilisateurs peuvent effectuer les principales opérations d'authentification d'un groupe d'utilisateurs. Elles incluent des pages pour la gestion des mots de passe, l'authentification multifactorielle (MFA) et la vérification des attributs. Pour plus d'informations sur l'expérience utilisateur dans l'interface utilisateur hébergée, consultez [Inscription et connexion à l'aide de l'interface utilisateur hébergée](#).

Les pages Web qui constituent l'interface utilisateur hébergée sont une application Web frontale pour les sessions utilisateur interactives avec vos clients. Votre application doit invoquer l'interface utilisateur hébergée dans les navigateurs de vos utilisateurs. Amazon Cognito ne prend pas en charge l'accès par programmation aux pages Web de ce chapitre. Les points de terminaison de fédération dans la [Référence des points de terminaison de fédération OAuth 2.0, OpenID Connect et SAML 2.0](#) qui renvoient une réponse JSON peuvent être interrogés directement dans le code de votre application. Le [Point de terminaison d'autorisation](#) redirige soit vers l'interface utilisateur hébergée, soit vers une page de connexion de l'IdP et doit également être ouvert dans les navigateurs des utilisateurs.

Les rubriques de ce guide décrivent en détail les points de terminaison d'interface utilisateur hébergée fréquemment utilisés. Amazon Cognito met les pages Web suivantes à votre disposition lorsque vous attribuez un domaine à votre groupe d'utilisateurs.

Points de terminaison d'interface utilisateur hébergée

URL de point de terminaison	Description	Procédure d'accès
<code>https://<i>Domaine de votre groupe d'utilisateurs</i> / login</code>	Connecte les utilisateurs locaux et fédérés du groupe d'utilisateurs.	Redirigez depuis des points de terminaison tels que Point de terminaison d'autorisation , <code>/logout</code> et <code>/confirmforgotPassword</code> . veuillez consulter Point de terminaison de connexion .
<code>https://<i>Domaine de votre groupe d'utilisateurs</i> / logout</code>	Déconnecte les utilisateurs du groupe d'utilisateurs.	Lien direct. veuillez consulter Point de terminaison de déconnexion .

URL de point de terminaison	Description	Procédure d'accès
<a href="https://<i>domaine de votre groupe d'utilisateurs</i> /confirmUser">https://<i>domaine de votre groupe d'utilisateurs</i> /confirmUser	Confirme les utilisateurs qui ont sélectionné un lien de messagerie pour vérifier leur compte utilisateur.	Lien sélectionné par l'utilisateur dans un e-mail.
<a href="https://<i>domaine de votre groupe d'utilisateurs</i> /signup">https://<i>domaine de votre groupe d'utilisateurs</i> /signup	Inscrit un nouvel utilisateur. La page /login redirige votre utilisateur vers /signup lorsqu'il sélectionne Sign up (S'inscrire).	Lien direct avec les mêmes paramètres que /oauth2/authorize .
<a href="https://<i>domaine de votre groupe d'utilisateurs</i> /confirm">https://<i>domaine de votre groupe d'utilisateurs</i> /confirm	Une fois que votre groupe d'utilisateurs a envoyé un code de confirmation à un utilisateur qui s'est inscrit, votre utilisateur est invité à entrer ce code.	Redirection uniquement depuis /signup.
<a href="https://<i>Domaine de votre groupe d'utilisateurs</i> /forgotPassword">https://<i>Domaine de votre groupe d'utilisateurs</i> /forgotPassword	Demande à votre utilisateur son nom d'utilisateur et envoie un code de réinitialisation de mot de passe. La page /login redirige votre utilisateur vers /forgotPassword lorsqu'il sélectionne Forgot your password? (Mot de passe oublié ?).	<ol style="list-style-type: none"> 1. À partir du lien Mot de passe oublié dans /login. 2. Lien direct avec les mêmes paramètres que /oauth2/authorize .

URL de point de terminaison	Description	Procédure d'accès
https://Domaine de votre groupe d'utilisateurs /confirmforgotPassword	Demande à votre utilisateur le code de réinitialisation de mot de passe et un nouveau mot de passe. La page /forgotPassword redirige votre utilisateur vers /confirmforgotPassword lorsqu'il sélectionne Reset your password (Réinitialiser votre mot de passe).	Redirection uniquement depuis /forgotPassword .
https://domaine de votre groupe d'utilisateurs /resendcode	Envoie un nouveau code de confirmation à un utilisateur qui s'est inscrit dans votre groupe d'utilisateurs.	Redirection uniquement à partir du lien Envoyer un nouveau code dans /confirm.

Rubriques

- [Point de terminaison de connexion](#)
- [Point de terminaison de déconnexion](#)

Point de terminaison de connexion

Le point de terminaison de connexion est un serveur d'authentification et une destination de redirection depuis le [Point de terminaison d'autorisation](#). Il s'agit du point d'entrée dans l'interface utilisateur hébergée lorsque vous ne spécifiez pas de fournisseur d'identité. Lorsque vous générez une redirection vers le point de terminaison de connexion, il charge la page de connexion et présente les options d'authentification configurées pour le client à l'utilisateur.

Note

Le point de terminaison de connexion est un composant de l'interface utilisateur hébergée. Dans votre application, invoquez les pages de fédération et d'interface utilisateur hébergée

qui redirigent vers le point de terminaison de connexion. L'accès direct des utilisateurs au point de terminaison de connexion n'est pas une bonne pratique.

The image shows a login interface with two main sections. On the left, under 'Sign in with your corporate ID', there is a blue button labeled 'MYSSO'. Below that, under 'Sign In with your social account', there are four buttons: 'Continue with Apple' (black), 'Continue with Login with Amazon' (yellow), 'Continue with Google' (blue), and 'Continue with Facebook' (dark blue). A small text at the bottom left reads 'We won't post to any of your accounts without asking first'. On the right, under 'Sign in with your username and password', there are two input fields: 'Username' and 'Password'. Below the password field is a link 'Forgot your password?' and a blue 'Sign in' button. At the bottom right, there is a link 'Need an account? Sign up'. The word 'OR' is placed between the social account buttons and the username/password form.

GET /login

Le point de terminaison `/login` prend en charge uniquement HTTPS GET pour la demande initiale de votre utilisateur. Votre application invoque la page dans un navigateur tel que Chrome ou Firefox. Lorsque vous redirigez vers `/login` depuis le [Point de terminaison d'autorisation](#), il transmet tous les paramètres que vous avez fournis dans votre demande initiale. Le point de terminaison de connexion prend en charge tous les paramètres de demande du point de terminaison d'autorisation. Vous pouvez également accéder directement au point de terminaison de connexion. Une bonne pratique consiste à initier toutes les sessions de vos utilisateurs dans `/oauth2/authorize`.

Exemple : demander à l'utilisateur de se connecter

Cet exemple affiche l'écran de connexion.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/login?  
    response_type=code&  
    client_id=ad398u21ijw3s9w3939&  
    redirect_uri=https://YOUR_APP/redirect_uri&  
    state=STATE&  
    scope=openid+profile+aws.cognito.signin.user.admin
```

Exemple — réponse

Le serveur d'authentification procède à une redirection vers votre application avec l'état et le code d'autorisation. Le serveur doit renvoyer le code et l'état dans les paramètres de chaîne de requête et non pas dans le fragment.

```
HTTP/1.1 302 Found  
    Location: https://YOUR_APP/redirect_uri?  
code=AUTHORIZATION_CODE&state=STATE
```

Demande de connexion initiée par l'utilisateur

Une fois que votre utilisateur a chargé le point de terminaison `/login`, il peut saisir un nom d'utilisateur et un mot de passe, puis choisir `Se connecter`. Ce faisant, il génère une demande HTTPS POST avec les mêmes paramètres de demande d'en-tête que la demande GET, ainsi qu'un corps de demande contenant son nom d'utilisateur, son mot de passe et une empreinte digitale d'appareil.

Point de terminaison de déconnexion

Le point de terminaison `/logout` est un point de terminaison de redirection. Il déconnecte l'utilisateur et le redirige soit vers une URL de déconnexion autorisée pour le client de votre application, soit vers le `/login` point de terminaison. Les paramètres disponibles dans une demande GET adressée au point de terminaison `/logout` sont adaptés aux cas d'utilisation de l'interface utilisateur hébergée Amazon Cognito.

Pour rediriger votre utilisateur vers l'interface utilisateur hébergée afin de se reconnecter, ajoutez un paramètre `redirect_uri` à votre demande. Une demande `logout` avec un

paramètre `redirect_uri` doit également inclure les paramètres de votre demande suivante au [Point de terminaison de connexion](#), comme `client_id`, `response_type` et `scope`.

Le point de terminaison de déconnexion est une application Web frontale pour les sessions utilisateur interactives avec vos clients. Votre application doit invoquer cet élément et d'autres points de terminaison d'interface utilisateur hébergés dans les navigateurs de vos utilisateurs.

Pour rediriger votre utilisateur vers la page de votre choix, ajoutez des URL de déconnexion autorisées à votre client d'application. Dans les demandes que vos utilisateurs adressent au point de terminaison `logout`, ajoutez les paramètres `logout_uri` et `client_id`. Si la valeur de `logout_uri` est l'une des URL de déconnexion autorisées pour votre client d'application, Amazon Cognito redirige les utilisateurs vers cette URL.

Avec la déconnexion unique (SLO) pour SAML 2.0 IdPs, Amazon Cognito redirige d'abord votre utilisateur vers le point de terminaison SLO que vous avez défini dans votre configuration IdP. Une fois que votre IdP a redirigé votre utilisateur vers, Amazon `saml2/logout` Cognito répond par une autre redirection vers ou depuis votre demande. `redirect_uri` `logout_uri` Pour plus d'informations, consultez [Flux de déconnexion SAML](#).

Le point de terminaison de déconnexion ne déconnecte pas les utilisateurs de l'OIDC ou des fournisseurs d'identité sociale (S)IdPs. Pour déconnecter les utilisateurs de leur session avec un IdP externe, dirigez-les vers la page de déconnexion de ce fournisseur.

GET /logout

Le point de terminaison `/logout` prend uniquement en charge HTTPS `GET`. Le client du groupe d'utilisateurs adresse généralement cette demande via le navigateur du système. Ce navigateur est généralement l'onglet Chrome personnalisé sous Android ou Safari View Control sous iOS.

Paramètres de demande

`client_id`

ID client d'application pour votre application. Pour obtenir un ID client d'application, vous devez inscrire l'application dans le groupe d'utilisateurs. Pour plus d'informations, consultez [Clients de l'application de groupe d'utilisateurs](#).

Obligatoire.

logout_uri

Redirigez votre utilisateur vers une page de déconnexion personnalisée avec un paramètre `logout_uri`. Définissez sa valeur sur l'URL de déconnexion du client d'application, où vous souhaitez rediriger votre utilisateur une fois qu'il se déconnecte. Utilisez `logout_uri` uniquement avec un paramètre `client_id`. Pour plus d'informations, consultez [Clients de l'application de groupe d'utilisateurs](#).

Vous pouvez également utiliser le paramètre `logout_uri` pour rediriger votre utilisateur vers la page de connexion d'un autre client d'application. Définissez la page de connexion de l'autre client d'application comme URL de rappel autorisée dans votre client d'application. Dans votre demande adressée au point de terminaison `/logout`, définissez la valeur du paramètre `logout_uri` sur la page de connexion codée en URL.

Amazon Cognito nécessite un paramètre `logout_uri` ou `redirect_uri` dans votre demande au point de terminaison `/logout`. Un paramètre `logout_uri` redirige votre utilisateur vers un autre site web. Si les paramètres `logout_uri` et `redirect_uri` sont inclus dans votre demande au point de terminaison `/logout`, Amazon Cognito utilisera exclusivement le paramètre `logout_uri`, en remplaçant le paramètre `redirect_uri`.

redirect_uri

Redirigez votre utilisateur vers votre page de connexion pour qu'il s'authentifie avec un paramètre `redirect_uri`. Définissez sa valeur sur l'URL de rappel autorisée du client d'application, où vous souhaitez rediriger votre utilisateur une fois qu'il se reconnecte. Ajoutez les paramètres `client_id`, `scope`, `state` et `response_type` que vous souhaitez transmettre à votre point de terminaison `/login`.

Amazon Cognito nécessite un paramètre `logout_uri` ou `redirect_uri` dans votre demande au point de terminaison `/logout`. Pour rediriger votre utilisateur vers votre `/login` point de terminaison afin de s'authentifier à nouveau et de transmettre des jetons à votre application, ajoutez un paramètre `redirect_uri`. Si les paramètres `logout_uri` et `redirect_uri` sont inclus dans votre demande au point de terminaison, `/logout` Amazon Cognito remplace le paramètre `redirect_uri` et traite le paramètre `logout_uri` exclusivement.

response_type

Réponse OAuth 2.0 que vous souhaitez recevoir d'Amazon Cognito une fois que votre utilisateur s'est connecté. `code` et `token` sont les valeurs valides du paramètre `response_type`.

Nécessaire si vous utilisez un paramètre `redirect_uri`.

state

Lorsque votre application ajoute un paramètre d'état à une demande, Amazon Cognito renvoie sa valeur à votre application lorsque le `/oauth2/logout` point de terminaison redirige votre utilisateur.

Ajoutez cette valeur à vos demandes afin de protéger votre système contre les attaques [CSRF](#) (cross-site request forgery, falsification de requête intersites).

Vous ne pouvez pas définir la valeur d'un paramètre `state` sur une chaîne JSON encodée par URL. Pour transmettre une chaîne correspondant à ce format dans un `state` paramètre, codez la chaîne en base64, puis décodez-la dans votre application.

Fortement recommandée si vous utilisez un paramètre `redirect_uri`.

scope

Étendues OAuth 2.0 que vous souhaitez demander à Amazon Cognito une fois que vous avez déconnecté l'utilisateur avec un paramètre `redirect_uri`. Amazon Cognito redirige votre utilisateur vers le point de terminaison `/login` avec le paramètre `scope` dans votre demande au point de terminaison `/logout`.

Facultatif si vous utilisez un paramètre `redirect_uri`. Si vous n'incluez pas de paramètre `scope`, Amazon Cognito redirige votre utilisateur vers le point de terminaison `/login` avec un paramètre `scope`. Quand Amazon Cognito redirige votre utilisateur et renseigne automatiquement `scope`, le paramètre inclut toutes les étendues autorisées pour votre client d'application.

Exemples de demandes

Exemple : déconnexion et redirection de l'utilisateur vers le client

À l'exception de `logout_uri` et `client_id`, tous les paramètres de requête possibles pour ce point de terminaison sont transmis au [Point de terminaison d'autorisation](#). Amazon Cognito redirige les sessions utilisateur vers l'URL avec la valeur `logout_uri`, en ignorant tous les autres paramètres de demande, lorsque les demandes incluent `logout_uri` et `client_id`. Cette URL doit être une URL de déconnexion autorisée pour le client de l'application.

Voici un exemple de demande de déconnexion et de redirection vers `https://www.example.com/welcome`.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/logout?
```

```
client_id=1example23456789&
logout_uri=https%3A%2F%2Fwww.example.com%2Fwelcome
```

Exemple : déconnectez-vous et demandez à l'utilisateur de se connecter en tant qu'autre utilisateur

Lorsque les demandes omettent `logout_uri`, mais fournissent les paramètres qui constituent une demande bien formée adressée au point de terminaison autorisé, Amazon Cognito redirige les utilisateurs vers une interface utilisateur hébergée. Le point de terminaison de déconnexion ajoute les paramètres de votre demande initiale à la destination de redirection. Le paramètre `redirect_uri` d'une demande adressée au point de terminaison de déconnexion n'est pas une URL de déconnexion, mais une URL de connexion que vous souhaitez transmettre au point de terminaison autorisé.

Voici un exemple de demande qui déconnecte un utilisateur, le redirige vers la page de connexion et fournit un code d'autorisation `https://www.example.com` après la connexion.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/logout?
  response_type=code&
  client_id=1example23456789&
  redirect_uri=https%3A%2F%2Fwww.example.com&
  state=example-state-value&
  nonce=example-nonce-value&
  scope=openid+profile+aws.cognito.signin.user.admin
```

Référence des points de terminaison de fédération OAuth 2.0, OpenID Connect et SAML 2.0

Amazon Cognito active les points de terminaison dans cette section lorsque vous ajoutez un domaine à votre groupe d'utilisateurs. Les points de terminaison de la fédération ne sont pas interactifs pour l'utilisateur. Ils jouent un rôle de service permettant à votre application de communiquer avec des fournisseurs d'identité OAuth 2.0, OIDC et SAML 2.0 tiers (). IdPs

Les rubriques de ce guide décrivent plusieurs points de terminaison OAuth 2.0 et OIDC fréquemment utilisés. Amazon Cognito crée les points de terminaison suivants lorsque vous affectez un domaine à votre groupe d'utilisateurs.

Points de terminaison de fédération de groupes d'utilisateurs

URL de point de terminaison	Description	Procédure d'accès
<code>https://<i>Domaine de votre groupe d'utilisateurs</i> / oauth2/authorize</code>	Redirige un utilisateur vers l'interface utilisateur hébergée ou pour qu'il se connecte auprès de son fournisseur d'identité.	Invoqué dans le navigateur du client pour démarrer l'authentification de l'utilisateur. veuillez consulter Point de terminaison d'autorisation .
<code>https://<i>Domaine de votre groupe d'utilisateurs</i> / oauth2/token</code>	Renvoie des jetons en fonction d'un code d'autorisation ou d'une demande d'informations d'identification client.	Demandé par l'application pour récupérer des jetons. veuillez consulter Point de terminaison de jeton .
<code>https://<i>Domaine de votre groupe d'utilisateurs</i> / oauth2/userinfo</code>	Renvoie les attributs utilisateur en fonction des étendues OAuth 2.0 et de l'identité de l'utilisateur dans un jeton d'accès.	Demandé par l'application pour récupérer le profil utilisateur. veuillez consulter Point de terminaison UserInfo .
<code>https://<i>Domaine de votre groupe d'utilisateurs</i> / oauth2/revoke</code>	Révoque un jeton d'actualisation et les jetons d'accès associés.	Demandé par l'application pour révoquer un jeton. veuillez consulter Point de terminaison de révocation .
<code>https://cognito-idp.<i>Région</i>.amazonaws.com/<i>ID de votre groupe d'utilisateurs</i> /.well-known/openid-configuration</code>	Répertoire de l'architecture OIDC de votre groupe d'utilisateurs.	Demandé par l'application pour localiser les métadonnées de l'émetteur du pool d'utilisateurs.
<code>https://cognito-idp.<i>Région</i>.amazonaws.com/<i>ID de votre groupe d'utilisateurs</i> /.well-known/jwks.json</code>	Clés publiques que vous pouvez utiliser pour valider les jetons Amazon Cognito.	Demandé par l'application pour vérifier les JWT.

URL de point de terminaison	Description	Procédure d'accès
https://Domaine de votre groupe d'utilisateurs /oauth2/idpresponse	Les fournisseurs d'identité sociale doivent rediriger vos utilisateurs vers ce point de terminaison à l'aide d'un code d'autorisation. Amazon Cognito échange le code contre un jeton lorsqu'il authentifie votre utilisateur fédéré.	Redirigé depuis la connexion au fournisseur d'identité OIDC en tant qu'URL de rappel du client du fournisseur d'identité.
https://Domaine de votre groupe d'utilisateurs /saml2/idpresponse	URL d'Assertion Consumer Response (ACS) pour l'intégration avec les fournisseurs d'identité SAML 2.0.	Redirigé depuis SAML 2.0 IdP en tant qu'URL ACS ou point d'origine pour la connexion initiée par l'IdP. ¹
https://Le domaine de votre groupe d'utilisateurs /saml2/logout	URL de déconnexion unique (SLO) pour l'intégration avec les fournisseurs d'identité SAML 2.0.	Redirigé depuis SAML 2.0 IdP en tant qu'URL de déconnexion unique (SLO). Accepte uniquement la reliure POST.

¹ Pour plus d'informations sur la connexion SAML initiée par l'IdP, consultez [Utilisation de la connexion SAML initiée par l'IdP](#)

Pour plus d'informations sur les standards OpenID Connect et OAuth, consultez [OpenID Connect 1.0](#) et [OAuth 2.0](#).

Rubriques

- [Point de terminaison d'autorisation](#)
- [Point de terminaison de jeton](#)
- [Point de terminaison UserInfo](#)
- [Point de terminaison de révocation](#)
- [point de terminaison saml2/idpresponse](#)

Point de terminaison d'autorisation

Le point de terminaison `/oauth2/authorize` est un point de terminaison de redirection qui prend en charge deux destinations de redirection. Si vous incluez un paramètre `identity_provider` ou `idp_identifieur` dans l'URL, il redirige en mode silencieux votre utilisateur vers la page de connexion de ce fournisseur d'identité (IdP). Sinon, il redirige vers le [Point de terminaison de connexion](#) avec les mêmes paramètres d'URL que ceux que vous avez inclus dans votre demande.

Le point de terminaison d'autorisation redirige vers l'interface utilisateur hébergée ou vers la page de connexion de l'IdP. La destination d'une session utilisateur sur ce point de terminaison est une page Web avec laquelle votre utilisateur doit interagir directement dans son navigateur.

Pour utiliser le point de terminaison d'autorisation, appelez le navigateur de votre utilisateur à l'adresse `/oauth2/authorize` avec des paramètres qui fournissent à votre groupe d'utilisateurs des informations sur les détails suivants du groupe d'utilisateurs.

- Client d'application auquel vous souhaitez vous connecter.
- URL de rappel à laquelle vous souhaitez accéder.
- Les portées OAuth 2.0 que vous souhaitez demander dans le jeton d'accès de votre utilisateur.
- IdP tiers (facultatif) que vous souhaitez utiliser pour vous connecter.

Vous pouvez également fournir les paramètres `state` et `nonce` utilisés par Amazon Cognito pour valider les demandes entrantes.

GET `/oauth2/authorize`

Le point de terminaison `/oauth2/authorize` prend uniquement en charge HTTPS GET. Votre application lance généralement cette demande dans le navigateur de votre utilisateur. Vous pouvez adresser vos demandes au point de terminaison `/oauth2/authorize` uniquement via HTTPS.

Pour en savoir plus sur la définition du point de terminaison d'autorisation dans la norme OpenID Connect (OIDC), veuillez consulter [Authorization Endpoint](#) (français non disponible).

Paramètres de demande

response_type

(Obligatoire) Type de réponse. Doit être `code` ou `token`.

Une demande réussie avec un `response_type` égal à `code` renvoie un octroi de code d'autorisation. L'octroi de code d'autorisation est un paramètre `code` ajouté par Amazon Cognito à votre URL de redirection. Votre application peut échanger ce code avec le [Point de terminaison de jeton](#) pour obtenir les jetons d'accès, d'identification et d'actualisation. À titre de bonne pratique en matière de sécurité et pour recevoir des jetons d'actualisation pour vos utilisateurs, utilisez un octroi de code d'autorisation dans votre application.

Une demande réussie avec un `response_type` égal à `token` renvoie un octroi implicite. Un octroi implicite est un jeton d'identifiant et d'accès ajouté par Amazon Cognito à votre URL de redirection. Un octroi implicite est moins sûr car il expose les jetons et les informations d'identification potentielles aux utilisateurs. Vous pouvez désactiver la prise en charge des octrois implicites dans la configuration de votre client d'application.

client_id

(Obligatoire) L'ID du client de l'application.

La valeur de `client_id` doit être l'ID d'un client d'application du groupe d'utilisateurs dans lequel vous effectuez la demande. Votre client d'application doit prendre en charge la connexion par des utilisateurs locaux Amazon Cognito ou par au moins un fournisseur d'identité tiers.

redirect_uri

(Obligatoire) URL vers laquelle le serveur d'authentification redirige le navigateur une fois qu'Amazon Cognito a autorisé l'utilisateur.

Un identificateur de ressource uniforme (URI) de redirection doit avoir les attributs suivants :

- Il doit s'agir d'un URI absolu.
- Vous devez avoir préalablement enregistré l'URI avec un client.
- Il ne peut pas inclure un composant de fragment.

Veillez consulter [OAuth 2.0 - Redirection Endpoint](#).

Amazon Cognito exige que votre URI de redirection utilise HTTPS, à l'exception de `http://localhost`, que vous pouvez définir comme URL de rappel à des fins de test.

Amazon Cognito prend également en charge les URL de rappel d'application comme `myapp://example`.

state

(Facultatif, recommandé) Lorsque votre application ajoute un paramètre d'état à une demande, Amazon Cognito renvoie sa valeur à votre application lorsque le `/oauth2/authorize` point de terminaison redirige votre utilisateur.

Ajoutez cette valeur à vos demandes afin de protéger votre système contre les attaques [CSRF](#) (cross-site request forgery, falsification de requête intersites).

Vous ne pouvez pas définir la valeur d'un paramètre `state` sur une chaîne JSON encodée par URL. Pour transmettre une chaîne correspondant à ce format dans un `state` paramètre, encodez-la en base64, puis décodez-la dans votre application.

identity_provider

(Facultatif) Ajoutez ce paramètre pour contourner l'interface utilisateur hébergée et rediriger votre utilisateur vers la page de connexion d'un fournisseur. La valeur du paramètre `identity_provider` est le nom du fournisseur d'identité, tel qu'il apparaît dans votre groupe d'utilisateurs.

- Pour les fournisseurs sociaux, vous pouvez utiliser les valeurs `identity_providerFacebook`, `Google`, `LoginWithAmazon` et `SignInWithApple`
- Pour les groupes d'utilisateurs Amazon Cognito, utilisez la valeur `COGNITO`
- Pour les fournisseurs d'identité SAML 2.0 et OpenID Connect (OIDC) (IdPs), utilisez le nom que vous avez attribué à l'IdP dans votre groupe d'utilisateurs.

idp_identifiant

(Facultatif) Ajoutez ce paramètre pour rediriger vers un fournisseur avec un autre nom pour le nom `identity_provider`. Vous pouvez saisir des identifiants pour vos fichiers SAML 2.0 et OIDC IdPs depuis l'onglet Expérience de connexion de la console Amazon Cognito.

scope

(Facultatif) Il peut s'agir d'une combinaison d'étendues réservées au système ou de portées personnalisées associées à un client. Les périmètres doivent être séparés par des espaces. Les périmètres dédiés à un système sont `openid`, `email`, `phone`, `profile` et `aws.cognito.signin.user.admin`. Tout périmètre utilisé doit être associé au client. Dans le cas contraire, il sera ignoré lors de l'exécution.

Si le client ne demande pas de périmètre, le serveur d'authentification utilise tous les périmètres associés au client.

Un jeton d'identification est renvoyé uniquement si le paramètre `openid` est demandé. Le jeton d'accès peut être utilisé pour des groupes d'utilisateurs Amazon Cognito que si le périmètre `aws.cognito.signin.user.admin` est demandé. Les paramètres `scope phone`, `email` et `profile` peuvent uniquement être demandés si le paramètre `scope openid` est également demandé. Ces paramètres `scope` régissent les revendications qui font partie du jeton d'identification.

code_challenge_method

(Facultatif) Le protocole de hachage que vous avez utilisé pour générer le défi. Le [PKCE RFC](#) définit deux méthodes, `S256` et « `plain` ». Cependant, le serveur d'authentification Amazon Cognito ne prend en charge que la méthode `S256`.

code_challenge

(Facultatif) Le défi que vous avez généré à partir du `code_verifier`.

Obligatoire uniquement lorsque vous spécifiez un paramètre `code_challenge_method`.

nonce

(Facultatif) Une valeur aléatoire que vous pouvez ajouter à la demande. La valeur `nonce` que vous fournissez est incluse dans le jeton d'identification émis par Amazon Cognito. Pour se prémunir contre les attaques par rejeu, votre application peut inspecter la revendication `nonce` dans le jeton d'identification et la comparer à celle que vous avez générée. Pour de plus amples informations sur la revendication `nonce`, veuillez consulter [ID token validation](#) (français non disponible) dans la norme OpenID Connect.

Exemples de demandes avec réponses positives

Les exemples suivants illustrent le format des requêtes HTTP adressées au `/oauth2/authorize` point de terminaison.

Octroi de code d'autorisation

Il s'agit d'un exemple de demande d'octroi de code d'autorisation.

Exemple — requête GET

La demande suivante lance une session pour récupérer un code d'autorisation que votre utilisateur transmet à votre application à `redirect_uri` destination. Cette session demande les champs d'application des attributs utilisateur et l'accès aux opérations de l'API en libre-service Amazon Cognito.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=openid+profile+aws.cognito.signin.user.admin
```

Exemple — réponse

Le serveur d'authentification Amazon Cognito effectue une redirection vers votre application avec le code d'autorisation et l'état. Le code d'autorisation est valide pendant cinq minutes.

```
HTTP/1.1 302 Found
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111&state=abcdefg
```

Octroi de code d'autorisation avec PKCE

Il s'agit d'un exemple de demande d'octroi de code d'autorisation avec [PKCE](#).

Exemple — requête GET

La demande suivante ajoute un `code_challenge` paramètre à la demande précédente. Pour terminer l'échange d'un code contre un jeton, vous devez inclure le `code_verifier` paramètre dans votre demande au `/oauth2/token` point de terminaison.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin&
code_challenge_method=S256&
code_challenge=a1b2c3d4...
```

Exemple — réponse

Le serveur d'authentification redirige vers votre application avec le code d'autorisation et l'état. Le code et l'état doivent être renvoyés dans les paramètres de la chaîne de requête et non dans le fragment :

```
HTTP/1.1 302 Found
```

```
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-EXAMPLE11111&state=abcdefg
```

Octroi de jeton sans paramètre de périmètre **openid**

Il s'agit d'un exemple de demande qui génère une autorisation implicite et renvoie les JWT directement à la session de l'utilisateur.

Exemple — requête GET

La demande suivante concerne une autorisation implicite de la part de votre serveur d'autorisation. Le jeton d'accès d'Amazon Cognito autorise les opérations d'API en libre-service.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=token&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin
```

Exemple — réponse

Le serveur d'autorisation Amazon Cognito effectue une redirection vers votre application avec un jeton d'accès. Comme la portée **openid** n'a pas été demandée, Amazon Cognito ne renvoie pas de jeton d'identification. De plus, Amazon Cognito ne renvoie pas de jeton d'actualisation dans ce flux. Amazon Cognito renvoie le jeton d'accès et l'état dans le fragment et non dans la chaîne de requête :

```
HTTP/1.1 302 Found
Location: https://YOUR\_APP/redirect\_uri#access\_token=ACCESS\_TOKEN&token\_type=bearer&expires\_in=3600&state=STATE
```

Octroi de jeton avec paramètre de portée **openid**

Il s'agit d'un exemple de demande qui génère une autorisation implicite et renvoie les JWT directement à la session de l'utilisateur.

Exemple — requête GET

La demande suivante concerne une autorisation implicite de la part de votre serveur d'autorisation. Le jeton d'accès d'Amazon Cognito autorise l'accès aux attributs utilisateur et aux opérations d'API en libre-service.

```
GET
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=token&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin+openid+profile
```

Exemple — réponse

Le serveur d'autorisation redirige vers votre application avec un jeton d'accès et un jeton d'identification (car le `openid` champ d'application a été inclus) :

```
HTTP/1.1 302 Found
Location: https://
www.example.com#id_token=eyJra67890EXAMPLE&access_token=eyJra12345EXAMPLE&token_type=bearer&exp
```

Exemples de réponses négatives

Amazon Cognito peut refuser votre demande. Les demandes négatives sont accompagnées d'un code d'erreur HTTP et d'une description que vous pouvez utiliser pour corriger les paramètres de votre demande. Voici des exemples de réponses négatives.

- Si `client_id` et `redirect_uri` sont valides, mais que les paramètres de la demande ne sont pas correctement formatés, le serveur d'authentification redirige l'erreur vers celle du client `redirect_uri` et ajoute un message d'erreur dans un paramètre d'URL. Voici des exemples de formatage incorrect.
 - La demande n'inclut aucun `response_type` paramètre.
 - La demande d'autorisation a fourni un `code_challenge` paramètre, mais pas un `code_challenge_method` paramètre.
 - La valeur du `code_challenge_method` paramètre ne l'est pas `S256`.

Voici la réponse à un exemple de demande dont le formatage est incorrect.

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_request
```

- Si le client demande code ou reçoit token ces demandes `response_type`, mais qu'il n'est pas autorisé à les traiter, le serveur d'autorisation Amazon Cognito retourne `unauthorized_client` vers celui du client `redirect_uri`, comme suit :

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=unauthorized_client
```

- Si le client demande une portée inconnue, incorrecte ou non valide, le serveur d'autorisation Amazon Cognito renvoie `invalid_scope` à l'URI `redirect_uri` du client, comme suit :

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_scope
```

- En cas d'erreur inattendue sur le serveur, le serveur d'authentification `server_error` revient sur celui du client `redirect_uri`. Comme l'erreur HTTP 500 n'est pas envoyée au client, elle ne s'affiche pas dans le navigateur de l'utilisateur. Le serveur d'autorisation renvoie l'erreur suivante.

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=server_error
```

- Lorsqu'Amazon Cognito s'authentifie par le biais d'une fédération auprès d'un tiers, Amazon IdPs Cognito peut rencontrer des problèmes de connexion, tels que les suivants :
 - Si un délai de connexion se produit lors de la demande d'un jeton auprès du fournisseur d'identité, le serveur d'authentification redirige l'erreur vers l'URI `redirect_uri` du client de la façon suivante :

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=Timeout+occurred+in+calling+IdP+token
+endpoint
```

- En cas d'expiration du délai de connexion lors de l'appel du `jwtks_uri` point de terminaison pour la validation du jeton d'identification, le serveur d'authentification redirige avec une erreur vers celui du client `redirect_uri` comme suit :

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=error_description=Timeout+in+calling+jwks
+uri
```

- Lors de l'authentification par fédération auprès d'un tiers IdPs, les fournisseurs peuvent renvoyer des réponses d'erreur. Cela peut être dû à des erreurs de configuration ou à d'autres raisons, telles que les suivantes :

- Si une réponse d'erreur est reçue de la part d'autres fournisseurs, le serveur d'authentification redirige l'erreur vers le `redirect_uri` du client de la façon suivante :

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=[IdP name]+Error+--[status code]+error
getting token
```

- Si une réponse d'erreur est reçue de Google, le serveur d'authentification redirige l'erreur vers le `redirect_uri` du client de la façon suivante :

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=Google+Error+--[status code]+[Google-
provided error code]
```

- Lorsqu'Amazon Cognito rencontre une exception de communication lorsqu'il se connecte à un IdP externe, le serveur d'authentification redirige avec une erreur vers le client avec l'un `redirect_uri` des messages suivants :

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=Connection+reset
```

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=Read+timed+out
```

Point de terminaison de jeton

Le [point de terminaison du jeton](#) OAuth 2.0 dans `/oauth2/token` émet des jetons Web JSON (JWT).

Le serveur d'autorisation OAuth 2.0 de votre groupe d'utilisateurs émet des jetons Web JSON (JWT) depuis le point de terminaison du jeton vers les types de sessions suivants :

1. Utilisateurs ayant effectué une demande d'octroi de code d'autorisation. L'utilisation réussie d'un code retourne les jetons d'ID, d'accès et d'actualisation.
2. Sessions Machine-to-machine (M2M) ayant obtenu une autorisation d'identification client. Une autorisation réussie avec le secret du client renvoie un jeton d'accès.
3. Utilisateurs qui se sont déjà connectés et ont reçu des jetons d'actualisation. L'authentification par jeton d'actualisation renvoie un nouvel identifiant et des jetons d'accès.

Note

Les utilisateurs qui se connectent avec un code d'autorisation octroyé dans l'interface utilisateur hébergée ou via la fédération peuvent toujours actualiser leurs jetons depuis le point de terminaison du jeton. Les utilisateurs qui se connectent à l'aide des opérations `InitiateAuth` de l'API `AdminInitiateAuth` peuvent actualiser leurs jetons avec le point de terminaison du jeton lorsque [les appareils mémorisés](#) ne sont pas actifs dans votre groupe d'utilisateurs. Si les appareils mémorisés sont actifs, actualisez les jetons avec les requêtes `AuthFlow` of `REFRESH_TOKEN_AUTH` in `InitiateAuth` ou `AdminInitiateAuth` API.

Le point de terminaison du jeton devient accessible au public lorsque vous ajoutez un domaine à votre groupe d'utilisateurs. Il accepte les demandes HTTP POST. Pour la sécurité des applications, utilisez PKCE avec les événements de connexion à votre code d'autorisation. PKCE vérifie que l'utilisateur qui transmet un code d'autorisation est le même que celui qui s'est authentifié. Pour plus d'informations sur le PKCE, consultez la norme [IETF RFC 7636](#).

Pour en savoir plus sur les clients de l'application du groupe d'utilisateurs et leurs types de subventions, leurs secrets clients, leurs portées autorisées et leurs ID clients, consultez [Clients de l'application de groupe d'utilisateurs](#). Vous pouvez en savoir plus sur l'autorisation M2M, les autorisations d'identification des clients et les étendues d'autorisation avec jetons d'accès à l'adresse. [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#)

Pour récupérer des informations sur un utilisateur à partir de son jeton d'accès, transmettez-les à votre demande [Point de terminaison UserInfo](#) ou à une demande d'[GetUserAPI](#).

POST /oauth2/jeton

Le point de terminaison /oauth2/token prend uniquement en charge HTTPS POST. Votre application envoie des requêtes à ce point de terminaison directement, sans passer par le navigateur de l'utilisateur.

Le point de terminaison du jeton prend en charge `client_secret_basic` et l'authentification `client_secret_post`. Pour plus d'informations sur la spécification OpenID Connect, consultez la section Authentification [du client](#). Pour plus d'informations sur le point de terminaison de jeton à partir de la norme OpenID Connect, consultez [Point de terminaison de jeton](#).

Paramètres de demande dans l'en-tête

Authorization

Si un secret a été attribué au client, ce dernier doit transmettre les paramètres `client_id` et `client_secret` dans l'en-tête d'autorisation en tant qu'autorisation HTTP `client_secret_basic`. Vous pouvez également inclure les `client_id` et `client_secret` dans le corps de la demande en tant qu'autorisation `client_secret_post`.

La chaîne d'en-tête d'autorisation est `Basic Base64Encode(client_id:client_secret)`. L'exemple suivant est un en-tête d'autorisation pour le client de l'application `djc98u3jiedmi283eu928` avec le secret `clientabcdef01234567890`, utilisant la version codée en Base64 de la chaîne : `djc98u3jiedmi283eu928:abcdef01234567890`

```
Authorization: Basic ZGpj0Th1M2ppZWRTaTI4M2V10TI40mFiY2RlZjAxMjM0NTY3ODkw
```

Content-Type

Définissez la valeur de ce paramètre sur 'application/x-www-form-urlencoded'.

Paramètres de la demande dans le corps

grant_type

(Obligatoire) Type de subvention OIDC que vous souhaitez demander.

Doit être `authorization_code`, `refresh_token` ou `client_credentials`. Vous pouvez demander un jeton d'accès pour une étendue personnalisée auprès du point de terminaison du jeton dans les conditions suivantes :

- Vous avez activé l'étendue demandée dans la configuration du client de votre application.
- Vous avez configuré votre client d'application avec un secret client.
- Vous activez l'octroi d'informations d'identification client dans le client de votre application.

client_id

(Facultatif) L'ID d'un client d'application dans votre groupe d'utilisateurs. Spécifiez le même client d'application qui a authentifié votre utilisateur.

Vous devez fournir ce paramètre si le client est public et n'a pas de secret, ou s'il n'est pas `client_secret_post` autorisé. `client_secret`

client_secret

(Facultatif) Le secret du client de l'application qui a authentifié votre utilisateur. Obligatoire si votre client d'application dispose d'un secret client et vous n'avez pas envoyé un en-tête `Authorization`.

scope

(Facultatif) Il peut s'agir d'une combinaison de toutes les étendues personnalisées associées à un client d'application. Toute étendue que vous demandez doit être activée pour le client de l'application. Dans le cas contraire, Amazon Cognito l'ignorera. Si le client ne demande aucune étendue, le serveur d'authentification attribue toutes les étendues personnalisées que vous avez autorisées dans la configuration de votre client d'application.

Utilisé uniquement si le `grant_type` est `client_credentials`.

redirect_uri

(Facultatif) Doit être le même `redirect_uri` que celui utilisé pour `authorization_code` entrer `/oauth2/authorize`.

Vous devez fournir ce paramètre si tel `grant_type` est le cas `authorization_code`.

refresh_token

(Facultatif) Pour générer de nouveaux jetons d'accès et d'identification pour la session d'un utilisateur, définissez la valeur d'un `refresh_token` paramètre de votre `/oauth2/token` demande sur un jeton d'actualisation précédemment émis par le même client d'application.

code

(Facultatif) Le code d'autorisation issu de l'octroi d'un code d'autorisation. Vous devez fournir ce paramètre si votre demande d'autorisation inclut un `grant_type` de `authorization_code`.

code_verifier

(Facultatif) La valeur arbitraire que vous avez utilisée pour calculer `code_challenge` dans une demande d'octroi de code d'autorisation avec PKCE.

Exemples de demandes avec réponses positives

Échange d'un code d'autorisation contre des jetons

Exemple — requête POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token&
      Content-Type='application/x-www-form-urlencoded'&

Authorization=Basic ZGpj0Th1M2ppZWRtaTI4M2V1OTI4OmFiY2RLZjAxMjM0NTY3ODkw

      grant_type=authorization_code&
      client_id=1example23456789&
      code=AUTHORIZATION_CODE&
      redirect_uri=com.myclientapp://myclient/redirect
```

Exemple — réponse

```
HTTP/1.1 200 OK

      Content-Type: application/json

      {
        "access_token":"eyJra1example",
        "id_token":"eyJra2example",
        "refresh_token":"eyJj3example",
        "token_type":"Bearer",
        "expires_in":3600
      }
```

Note

Le point de terminaison de jeton renvoie `refresh_token` uniquement lorsque la valeur de `grant_type` est `authorization_code`.

Échange d'informations d'identification de client contre un jeton d'accès : secret de client dans l'en-tête d'autorisation

Exemple — requête POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
      Content-Type='application/x-www-form-urlencoded'&

Authorization=Basic ZGpj0Th1M2ppZWRtaTI4M2V1OTI4OmFiY2RLZjAxMjM0NTY3ODkw

      grant_type=client_credentials&
```

```
client_id=1example23456789&
```

```
scope=resourceServerIdentifier1/scope1 resourceServerIdentifier2/scope2
```

Exemple — réponse

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{  
  "access_token": "eyJra1example",  
  "token_type": "Bearer",  
  "expires_in": 3600  
}
```

Échange d'informations d'identification de client contre un jeton d'accès : secret de client dans le corps de la demande

Exemple — requête POST

```
POST /oauth2/token HTTP/1.1
```

```
Content-Type: application/x-www-form-urlencoded
```

```
X-Amz-Target: AWSCognitoIdentityProviderService.Client credentials request
```

```
User-Agent: USER_AGENT
```

```
Accept: /
```

```
Accept-Encoding: gzip, deflate, br
```

```
Content-Length: 177
```

```
Referer: http://auth.example.com/oauth2/token
```

```
Host: auth.example.com
```

```
Connection: keep-alive
```

```
grant_type=client_credentials&client_id=1example23456789&scope=my_resource_server_identifier%2F
```

Exemple — réponse

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=UTF-8
```

```
Date: Tue, 05 Dec 2023 16:11:11 GMT
```

```
x-amz-cognito-request-id: 829f4fe2-a1ee-476e-b834-5cd85c03373b
```

```
{
  "access_token": "eyJra12345EXAMPLE",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```

Échange d'un octroi de code d'autorisation avec PKCE contre des jetons

Exemple — requête POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token
      Content-Type='application/x-www-form-urlencoded'&

Authorization=Basic ZGpj0Th1M2ppZWRtaTI4M2V1OTI4OmFiY2RLZjAxMjM0NTY3ODkw

      grant_type=authorization_code&
      client_id=1example23456789&
      code=AUTHORIZATION_CODE&
      code_verifier=CODE_VERIFIER&
      redirect_uri=com.myclientapp://myclient/redirect
```

Exemple — réponse

```
HTTP/1.1 200 OK

      Content-Type: application/json

      {
        "access_token": "eyJra1example",
        "id_token": "eyJra2example",
        "refresh_token": "eyJj3example",
        "token_type": "Bearer",
        "expires_in": 3600
      }
```

Note

Le point de terminaison de jeton renvoie `refresh_token` uniquement lorsque la valeur de `grant_type` est `authorization_code`.

Échange d'un jeton d'actualisation contre des jetons

Exemple — requête POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
      Content-Type='application/x-www-form-urlencoded'&

Authorization=Basic ZGpj0Th1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

      grant_type=refresh_token&
      client_id=1example23456789&
      refresh_token=eyJj3example
```

Exemple — réponse

```
HTTP/1.1 200 OK

      Content-Type: application/json

      {
        "access_token": "eyJra1example",
        "id_token": "eyJra2example",
        "token_type": "Bearer",
        "expires_in": 3600
      }
```

Note

Le point de terminaison de jeton renvoie `refresh_token` uniquement lorsque la valeur de `grant_type` est `authorization_code`.

Exemples de réponses négatives

Exemple — réponse à une erreur

```
HTTP/1.1 400 Bad Request

      Content-Type: application/json;charset=UTF-8

      {
```

```
"error": "invalid_request|invalid_client|invalid_grant|
unauthorized_client|unsupported_grant_type"
}
```

invalid_request

Un paramètre obligatoire n'est pas inclus dans la demande, la demande comprend une valeur de paramètre non pris en charge (autre que `unsupported_grant_type`) ou la demande présente un autre défaut. Par exemple, `grant_type` est `refresh_token` mais `refresh_token` n'est pas inclus.

invalid_client

Échec de l'authentification du client. Par exemple, lorsque le client comprend `client_id` et `client_secret` dans l'en-tête d'autorisation, mais il n'existe pas de client avec ce `client_id` et ce `client_secret`.

invalid_grant

Le jeton d'actualisation a été révoqué.

Le code d'autorisation a déjà été utilisé ou n'existe pas.

Le client d'application n'a pas accès en lecture à tous les [attributs](#) dans l'étendue demandée. Par exemple, votre application demande l'étendue `email` et votre client d'application peut lire l'attribut `email`, mais pas `email_verified`.

unauthorized_client

Le client n'a pas d'autorisation pour le flux d'octroi de code ou pour les jetons d'actualisation.

unsupported_grant_type

Renvoyé si `grant_type` est différent de `authorization_code`, `refresh_token` ou `client_credentials`.

Point de terminaison UserInfo

Le point de terminaison `userInfo` est un [point de terminaison userInfo](#) OpenID Connect (OIDC). Il répond par des attributs utilisateur lorsque les fournisseurs de services présentent les jetons d'accès issus par votre [Point de terminaison de jeton](#). Les portées du jeton d'accès de votre utilisateur définissent les attributs utilisateur que le point de terminaison `userInfo` renvoie dans sa réponse. La portée `openid` doit correspondre à l'une des demandes de jeton d'accès.

Amazon Cognito émet des jetons d'accès en réponse aux demandes d'API des groupes d'utilisateurs comme [InitiateAuth](#). Comme ces jetons d'accès ne contiennent pas de portées, le point de terminaison `userInfo` ne les accepte pas. À la place, vous devez présenter les jetons d'accès de votre point de terminaison de jeton.

Votre fournisseur d'identité (IdP) tiers OAuth 2.0 héberge également un point de terminaison `userInfo`. Lorsque votre utilisateur s'authentifie auprès de cet IdP, Amazon Cognito échange silencieusement un code d'autorisation avec le point de terminaison de l'IdP. Votre groupe d'utilisateurs transmet le jeton d'accès IdP pour autoriser la récupération des informations utilisateur depuis le point de terminaison `userInfo`.

GET /oauth2/userInfo

Votre application envoie des demandes à ce point de terminaison directement, sans passer par un navigateur.

Pour en savoir plus, consultez [Point de terminaison UserInfo](#) dans la spécification OpenID Connect (OIDC).

Rubriques

- [Paramètres de demande dans l'en-tête](#)
- [Exemple — demande](#)
- [Exemple — réponse positive](#)
- [Exemple de réponses négatives](#)

Paramètres de demande dans l'en-tête

Authorization: Bearer <access_token>

Passez le jeton d'accès dans le champ d'en-tête d'autorisation.

Obligatoire.

Exemple — demande

```
GET /oauth2/userInfo HTTP/1.1
Content-Type: application/x-amz-json-1.1
Authorization: Bearer eyJra12345EXAMPLE
```

```
User-Agent: [User agent]
Accept: */*
Host: auth.example.com
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Exemple — réponse positive

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: [Integer]
Date: [Timestamp]
x-amz-cognito-request-id: [UUID]
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Server: Server
Connection: keep-alive
{
  "sub": "[UUID]",
  "email_verified": "true",
  "custom:mycustom1": "CustomValue",
  "phone_number_verified": "true",
  "phone_number": "+12065551212",
  "email": "bob@example.com",
  "username": "bob"
}
```

Pour obtenir la liste des revendications OIDC, consultez la référence aux [revendications standard](#). Actuellement, Amazon Cognito renvoie les valeurs pour `email_verified` et `phone_number_verified` sous forme de chaînes.

Exemple de réponses négatives

Exemple — mauvaise demande

```
HTTP/1.1 400 Bad Request
WWW-Authenticate: error="invalid_request",
```

```
error_description="Bad OAuth2 request at UserInfo Endpoint"
```

invalid_request

Il manque un paramètre obligatoire à la demande, elle inclut une valeur de paramètre non prise en charge ou elle est mal formée.

Exemple : mauvais jeton

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: error="invalid_token",
error_description="Access token is expired, disabled, or deleted, or the user has
globally signed out."
```

invalid_token

Le jeton d'accès est expiré, révoqué, mal formé ou il n'est pas valide.

Point de terminaison de révocation

Le point de `oauth2/revoke` terminaison/révoque le jeton d'accès d'un utilisateur initialement émis par Amazon Cognito avec le jeton d'actualisation que vous fournissez. Ce point de terminaison révoque également tous les jetons d'accès et d'identité ultérieurs du même jeton d'actualisation. Une fois que le point de terminaison a révoqué les jetons, vous ne pouvez pas utiliser les jetons d'accès révoqués pour accéder aux API authentifiées par les jetons Amazon Cognito.

POST `/oauth2/revoke`

Le point de terminaison `/oauth2/revoke` prend uniquement en charge HTTPS POST. Le client du groupe d'utilisateurs adresse les demandes directement à ce point de terminaison, et non via le navigateur du système.

Paramètres de demande dans l'en-tête

Authorization

Si le client de votre application possède un secret client, l'application doit transmettre son code `client_id` et `client_secret` dans l'en-tête d'autorisation via l'autorisation HTTP de base. Le secret est [Basic](#) `Base64Encode(client_id:client_secret)`.

Content-Type

Doit toujours être 'application/x-www-form-urlencoded'.

Paramètres de la demande dans le corps

token

(Obligatoire) Le jeton d'actualisation que le client souhaite révoquer. La demande révoque également tous les jetons d'accès émis par Amazon Cognito avec ce jeton d'actualisation.

Obligatoire.

client_id

(Facultatif) L'ID client de l'application pour le jeton que vous souhaitez révoquer.

Obligatoire si le client est public et n'a pas de secret.

Exemples de demandes de révocation

Exemple 1 : Révocation d'un jeton pour un client d'application sans secret client

```
POST /oauth2/revoke HTTP/1.1
Host: https://mydomain.auth.us-east-1.amazoncognito.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
token=2YotnFZFEjr1zCsicMWpAA&
client_id=djc98u3jiedmi283eu928
```

Exemple 2 : Révocation d'un jeton pour un client d'application avec un secret client

```
POST /oauth2/revoke HTTP/1.1
Host: https://mydomain.auth.us-east-1.amazoncognito.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
token=2YotnFZFEjr1zCsicMWpAA
```

Réponse d'erreur Révocation

Une réponse réussie contient un corps vide. La réponse d'erreur est un objet JSON avec un champ `error` et, dans certains cas, un champ `error_description`.

Erreurs de point de terminaison

- Si le jeton n'est pas présent dans la demande ou si la fonction est désactivée pour le client d'application, vous recevez HTTP 400 et l'erreur `invalid_request`.
- Si le jeton envoyé par Amazon Cognito dans la demande de révocation n'est pas un jeton d'actualisation, vous recevez HTTP 400 et l'erreur `unsupported_token_type`.
- Si les informations d'identification du client ne sont pas valides, vous recevez HTTP 401 et l'erreur `invalid_client`.
- Si le jeton a été révoqué ou si le client a soumis un jeton non valide, vous recevez HTTP 200 OK.

point de terminaison saml2/idpresponse

Il `/saml2/idpresponse` reçoit des assertions SAML. Lors de la connexion service-provider-initiated (initiée par le SP), votre fournisseur d'identité (IdP) SAML 2.0 redirige votre utilisateur vers ce point de terminaison avec sa réponse SAML. Lors de la connexion initiée par le SP, votre application n'interagit pas avec ce point de terminaison. Configurez votre IdP avec le chemin d'accès à votre URL en `saml2/idpresponse` tant qu'URL du service client d'assertions (ACS). Pour plus d'informations sur le lancement de session, consultez [Lancement de séance SAML dans les groupes d'utilisateurs Amazon Cognito](#).

Lors de la connexion initiée par l'IdP, vos utilisateurs peuvent se connecter avec votre IdP par le biais de votre propre processus et soumettre une assertion SAML dans le corps d'une demande via HTTPS. HTTP POST Le corps de votre POST demande doit être un SAMLResponse paramètre et un Relaystate paramètre. Pour plus d'informations, consultez [Utilisation de la connexion SAML initiée par l'IdP](#).

POSTER `/saml2/idpresponse`

Pour utiliser le `/saml2/idpresponse` point de terminaison lors d'une connexion initiée par un IdP, générez une requête POST avec des paramètres qui fournissent à votre groupe d'utilisateurs des informations sur la session de votre utilisateur.

- Le client d'application auquel ils souhaitent se connecter.
- L'URL de rappel à laquelle ils veulent se retrouver.

- Le champ d'application OAuth 2.0 qu'ils souhaitent demander dans le jeton d'accès de votre utilisateur.
- L'IdP à l'origine de la demande de connexion.

Paramètres du corps de la demande initiée par l'IDP

Réponse SAML

Une assertion SAML codée en Base64 provenant d'un IdP associé à un client d'application valide et à une configuration IdP dans votre groupe d'utilisateurs.

RelayState

Un RelayState paramètre contient les paramètres de demande que vous transmettriez autrement au oauth2/authorize point de terminaison. Pour obtenir des informations détaillées sur ces paramètres, consultez [Point de terminaison d'autorisation](#).

response_type

Type de subvention OAuth 2.0.

client_id

ID du client d'application.

redirect_uri

URL vers laquelle le serveur d'authentification redirige le navigateur une fois qu'Amazon Cognito a autorisé l'utilisateur.

identity_provider

Le nom du fournisseur d'identité vers lequel vous souhaitez rediriger votre utilisateur.

idp_identifier

L'identifiant du fournisseur d'identité vers lequel vous souhaitez rediriger votre utilisateur.

scope

Le champ d'application OAuth 2.0 que vous souhaitez que votre utilisateur demande au serveur d'autorisation.

Exemples de demandes avec réponses positives

Exemple — requête POST

La demande suivante concerne l'octroi d'un code d'autorisation à un utilisateur depuis l'IdP MySAMLIdP dans le client de l'application. `1example23456789` L'utilisateur redirige vers `https://www.example.com` avec son code d'autorisation, qui peut être échangé contre des jetons incluant un jeton d'accès avec les champs d'application OAuth 2.0, et. `openid email phone`

```
POST /saml2/idpresponse HTTP/1.1
User-Agent: USER_AGENT
Accept: */*
Host: example.auth.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded

SAMLResponse=[Base64-encoded SAML assertion]&RelayState=identity_provider
%3DMySAMLIdP%26client_id%3D1example23456789%26redirect_uri%3Dhttps%3A%2F
%2Fwww.example.com%26response_type%3Dcode%26scope%3Demail%2Bopenid%2Bphone
```

Exemple — réponse

Voici la réponse à la demande précédente.

```
HTTP/1.1 302 Found
Date: Wed, 06 Dec 2023 00:15:29 GMT
Content-Length: 0
x-amz-cognito-request-id: 8aba6eb5-fb54-4bc6-9368-c3878434f0fb
Location: https://www.example.com?code=[Authorization code]
```

Octrois OAuth 2.0

Le serveur d'autorisation OAuth 2.0 du groupe d'utilisateurs Amazon Cognito émet des jetons en réponse à trois types d'[octrois d'autorisations](#) d'OAuth 2.0. Vous pouvez définir les types d'octroi pris en charge pour chaque client d'application de votre groupe d'utilisateurs. Vous ne pouvez pas activer les octrois d'informations d'identification client dans le même client d'application que les octrois de code implicite ou de code d'autorisation. Les demandes d'octroi de code implicite et de code d'autorisation commencent à votre [Point de terminaison d'autorisation](#), et les demandes d'octroi d'informations d'identification client commencent à votre [Point de terminaison de jeton](#).

Octroi de code d'autorisation

En réponse à votre demande d'authentification réussie, le serveur d'autorisation ajoute un code d'autorisation dans un paramètre `code` de votre URL de rappel. Vous devez alors échanger ce code avec le [Point de terminaison de jeton](#) pour obtenir des jetons d'identification, d'accès et

d'actualisation. Pour demander l'octroi d'un code d'autorisation, définissez `response_type` sur `code` dans votre demande. Pour obtenir un exemple de demande, consultez [Octroi de code d'autorisation](#).

L'octroi d'un code d'autorisation est la forme la plus sécurisée d'octroi d'autorisation. Le contenu des jetons n'est pas montré directement à vos utilisateurs. À la place, votre application est chargée de récupérer et de stocker en toute sécurité les jetons de vos utilisateurs. Dans Amazon Cognito, l'octroi d'un code d'autorisation est le seul moyen d'obtenir les trois types de jetons (ID, accès et actualisation) auprès du serveur d'autorisation. Vous pouvez également obtenir les trois types de jetons lors de l'authentification via l'API des groupes d'utilisateurs Amazon Cognito, mais l'API ne délivre pas de jetons d'accès avec des étendues autres que `aws.cognito.signin.user.admin`.

Octroi implicite

En réponse à votre demande d'authentification réussie, le serveur d'autorisation ajoute un jeton d'accès dans un paramètre `access_token` et un jeton d'identification dans un paramètre `id_token`, dans votre URL de rappel. Un octroi implicite ne nécessite aucune interaction supplémentaire avec le [Point de terminaison de jeton](#). Pour demander un octroi implicite, définissez `response_type` sur `token` dans votre demande. L'octroi implicite génère uniquement un identifiant et un jeton d'accès. Pour obtenir un exemple de demande, consultez [Octroi de jeton sans paramètre de périmètre openid](#).

L'octroi implicite est un octroi d'autorisation hérité. Contrairement à l'octroi de code d'autorisation, les utilisateurs peuvent intercepter et inspecter vos jetons. Pour empêcher la livraison de jetons par le biais d'un octroi implicite, configurez votre client d'application pour qu'il prenne en charge uniquement l'octroi de code d'autorisation.

Informations d'identification client

Les informations d'identification du client sont une autorisation d'accès uniquement. machine-to-machine Pour recevoir un octroi d'informations d'identification client, contournez le [Point de terminaison d'autorisation](#) et générez une demande directement auprès du [Point de terminaison de jeton](#). Votre client d'application doit disposer d'un secret client et prendre en charge les octrois d'informations d'identification client. En réponse à votre demande réussie, le serveur d'autorisation renvoie un jeton d'accès.

Le jeton d'accès issu d'un octroi d'informations d'identification client est un mécanisme d'autorisation qui contient des étendues OAuth 2.0. Généralement, le jeton contient des champs standard personnalisés d'étendue qui autorisent les opérations HTTP à accéder à des API

dont l'accès est protégé. Pour plus d'informations, consultez [Champs d'application, M2M et autorisation d'API avec les serveurs de ressources](#).

Les informations d'identification des clients vous permettent d'ajouter des frais à votre AWS facture. Pour plus d'informations, consultez [Tarification d'Amazon Cognito](#).

Utilisation de PKCE pour octroyer des codes d'autorisation aux groupes d'utilisateurs Amazon Cognito

Amazon Cognito prend en charge l'authentification par clé de preuve pour l'échange de code (PKCE) lors de l'octroi de codes d'autorisation. PKCE est une extension de l'octroi de code d'autorisation OAuth 2.0 pour les clients publics. Le PKCE protège contre le rachat de codes d'autorisation interceptés.

Comment Amazon Cognito utilise PKCE

Pour démarrer l'authentification avec PKCE, votre application doit générer une valeur de chaîne unique. Cette chaîne est le vérificateur de code, une valeur secrète qu'Amazon Cognito utilise pour comparer le client demandant l'autorisation initiale au client échangeant le code d'autorisation contre des jetons.

Votre application doit appliquer un hachage SHA256 à la chaîne du vérificateur de code et encoder le résultat en base64. Passez la chaîne hachée en [Point de terminaison d'autorisation](#) tant que `code_challenge` paramètre dans le corps de la demande. Lorsque votre application échange le code d'autorisation contre des jetons, elle doit inclure la chaîne du vérificateur de code en texte brut en tant que `code_verifier` paramètre dans le corps de la demande au [Point de terminaison de jeton](#) Amazon Cognito effectue la même hash-and-encode opération sur le vérificateur de code. Amazon Cognito ne renvoie les jetons d'identification, d'accès et d'actualisation que s'il détermine que le vérificateur de code génère le même défi de code que celui qu'il a reçu dans la demande d'autorisation.

Pour implémenter le flux d'octroi d'autorisations avec PKCE

1. Ouvrez la [console Amazon Cognito](#). Si vous y êtes invité, entrez vos AWS informations d'identification.
2. Choisissez Groupes d'utilisateurs.

3. Choisissez un groupe d'utilisateurs existant dans la liste ou [créez-en un](#). Si vous créez un groupe d'utilisateurs, vous serez invité à configurer un client d'application et à configurer l'interface utilisateur hébergée pendant l'assistant.
 - a. Si vous créez un nouveau groupe d'utilisateurs, configurez un client d'application et configurez l'interface utilisateur hébergée lors de la configuration guidée.
 - b. Si vous configurez un groupe d'utilisateurs existant, ajoutez un [domaine](#) et un [client d'application public](#), si ce n'est pas déjà fait.
4. Générez une chaîne alphanumérique aléatoire, généralement un identifiant unique universel (UUID), afin de créer un défi de code pour le PKCE. Cette chaîne est la valeur du `code_verifier` paramètre que vous soumettrez dans votre demande au [Point de terminaison de jeton](#).
5. Hachez la `code_verifier` chaîne avec l'algorithme SHA256. Codez le résultat de l'opération de hachage en base64. Cette chaîne est la valeur du `code_challenge` paramètre que vous soumettrez dans votre demande au [Point de terminaison d'autorisation](#).

L'Pythonexemple suivant génère un `code_verifier` et calcule le `code_challenge` :

```
#!/usr/bin/env python3

import random
from base64 import urlsafe_b64encode
from hashlib import sha256
from string import ascii_letters
from string import digits

# use a cryptographically strong random number generator source
rand = random.SystemRandom()

code_verifier = ''.join(rand.choices(ascii_letters + digits, k=128))
code_verifier_hash = sha256(code_verifier.encode()).digest()
code_challenge = urlsafe_b64encode(code_verifier_hash).decode().rstrip('=')

print(f"code challenge: {code_challenge}")
print(f"code verifier: {code_verifier}")
```

Voici un exemple de sortie du Python script :

```
code challenge: Eh0mg-0Zv7BAyo-tdv_vYamx1bo0YDulDklyXoMDtLg
```

```
code verifier: 9D-aW_iygXrgQcWJd0y0tNVMPsXsChIc2xceDhvYVdGLCBk-  
JWFTmBNjvKSd0rjTTYaz0FbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFgyE8yA-05-_v7Dxf3EIYJH
```

6. Connexion complète à l'interface utilisateur hébergée avec une demande d'octroi de code d'autorisation auprès de PKCE. Voici un exemple d'URL :

```
https://mydomain.us-east-1.amazonaws.com/oauth2/authorize?  
response_type=code&client_id=example23456789&redirect_uri=https://  
www.example.com&code_challenge=Eh0mg-0Zv7BAyo-  
tdv_vYamx1bo0YDuLDklyXoMDtLg&code_challenge_method=S256
```

7. Collectez l'autorisation code et échangez-la contre des jetons avec le point de terminaison du jeton. Voici un exemple de demande :

```
POST /oauth2/token HTTP/1.1  
Host: mydomain.us-east-1.amazonaws.com  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 296  
  
redirect_uri=https%3A%2F%2Fwww.example.com&  
client_id=example23456789&  
code=7378f445-c87f-400c-855e-0297d072ff03&  
grant_type=authorization_code&  
code_verifier=9D-aW_iygXrgQcWJd0y0tNVMPsXsChIc2xceDhvYVdGLCBk-  
JWFTmBNjvKSd0rjTTYaz0FbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFgyE8yA-05-_v7Dxf3EIYJH
```

8. Passez en revue la réponse. Il contiendra des jetons d'identification, d'accès et d'actualisation. Pour plus d'informations sur l'utilisation des jetons du pool d'utilisateurs Amazon Cognito, consultez [Utilisation des jetons avec des groupes d'utilisateurs](#)

Réponses aux erreurs de l'interface utilisateur hébergée et de fédération

Un processus de connexion dans l'interface utilisateur hébergée ou dans la connexion fédérée peut renvoyer une erreur. Les conditions suivantes peuvent avoir pour conséquence que l'authentification se termine avec une erreur.

- Un utilisateur effectue une opération que votre groupe d'utilisateurs ne peut pas effectuer.
- Un déclencheur Lambda ne répond pas avec la syntaxe attendue.
- Votre fournisseur d'identité (IdP) renvoie une erreur.
- Amazon Cognito n'a pas pu valider les informations d'attribut fournies par votre utilisateur.

- Votre fournisseur d'identité n'a pas envoyé les champs standard correspondant aux attributs requis.

Quand Amazon Cognito rencontre une erreur, il la communique de l'une des manières suivantes.

1. Amazon Cognito envoie une URL de redirection avec l'erreur dans les paramètres de demande.
2. Amazon Cognito affiche une erreur dans l'interface utilisateur hébergée.

Les erreurs qu'Amazon Cognito ajoute aux paramètres de demande ont le format suivant.

```
https://<Callback URL>/?error_description=error+description&error=error+name
```

Lorsque vous aidez vos utilisateurs à envoyer des informations d'erreur lorsqu'ils ne peuvent pas effectuer une opération, demandez-leur de capturer l'URL et le texte ou une capture d'écran de la page.

Note

Les descriptions d'erreurs Amazon Cognito ne sont pas des chaînes fixes et vous ne devez pas utiliser de logique basée sur un modèle ou un format fixe.

Messages d'erreur des fournisseurs d'identité sociale et OIDC

Votre fournisseur d'identité peut renvoyer une erreur. Quand un fournisseur d'identité OIDC ou OAuth 2.0 renvoie une erreur conforme aux standards, Amazon Cognito redirige votre utilisateur vers l'URL de rappel et ajoute la réponse d'erreur du fournisseur aux paramètres de demande d'erreur. Amazon Cognito ajoute le nom du fournisseur et le code d'erreur HTTP aux chaînes d'erreur existantes.

L'URL suivante est un exemple de redirection depuis un fournisseur d'identité qui a renvoyé une erreur à Amazon Cognito.

```
https://www.amazon.com/?error_description=LoginWithAmazon+Error+-+400+invalid_request+The+request+is+missing+a+required+parameter+%3A+client_secret&error=invalid_request
```

Comme Amazon Cognito renvoie uniquement ce qu'il reçoit d'un fournisseur, votre utilisateur peut voir un sous-ensemble de ces informations.

Quand votre utilisateur rencontre un problème avec la connexion initiale via votre fournisseur d'identité, le fournisseur d'identité remet les messages d'erreur directement à votre utilisateur. Amazon Cognito transmet un message d'erreur à votre utilisateur lorsqu'il demande à votre fournisseur d'identité de valider la session de votre utilisateur. Amazon Cognito transmet les messages d'erreur des fournisseurs d'identité OAuth et OIDC provenant des points de terminaison suivants.

`/token`

Amazon Cognito échange un code d'autorisation du fournisseur d'identité pour obtenir un jeton d'accès.

`/.well-known/openid-configuration`

Amazon Cognito découvre le chemin d'accès aux points de terminaison de votre émetteur.

`/.well-known/jwks.json`

Pour vérifier les jetons Web JSON (JWT) de votre utilisateur, Amazon Cognito découvre les clés Web JSON (JWK) que votre fournisseur d'identité utilise pour signer les jetons.

Comme Amazon Cognito ne lance pas de sessions sortantes vers des fournisseurs SAML 2.0 susceptibles de renvoyer des erreurs HTTP, les erreurs de vos utilisateurs dans le cadre d'une session avec un fournisseur d'identité SAML 2.0 n'incluent pas ce type de message d'erreur de fournisseur.

Référence d'API des groupes d'utilisateurs Amazon Cognito

Avec les groupes d'utilisateurs Amazon Cognito, vous pouvez inscrire et connecter des utilisateurs à l'aide de votre application web et mobile. Vous pouvez modifier les mots de passe des utilisateurs authentifiés et lancer des flux d'oubli de mot de passe pour les utilisateurs non authentifiés. Pour plus d'informations, consultez [Flux d'authentification de groupe d'utilisateurs](#) et [Utilisation des jetons avec des groupes d'utilisateurs](#).

L'API des groupes d'utilisateurs Amazon Cognito propose des opérations permettant d'afficher et de modifier les groupes d'utilisateurs et les utilisateurs et d'assurer l'authentification et l'autorisation des utilisateurs. Pour obtenir une description des classes d'opérations d'API qui se combinent dans l'API des groupes d'utilisateurs Amazon Cognito, consultez [Utilisation de l'API des groupes d'utilisateurs et des points de terminaison d'un groupe d'utilisateurs Amazon Cognito](#).

Pour obtenir une liste détaillée des opérations de l'API des groupes d'utilisateurs Amazon Cognito et de leur syntaxe, consultez [Référence d'API des groupes d'utilisateurs Amazon Cognito](#). Dans chaque page d'informations de référence sur l'API des groupes d'utilisateurs Amazon Cognito, vous trouverez des liens vers des documents de référence contenant la syntaxe et des exemples pour différents kits SDK AWS.

Référence d'API des groupes d'identités Amazon Cognito (identités fédérées)

Un groupe d'identités Amazon Cognito permet aux utilisateurs de votre application web ou mobile d'obtenir des informations d'identification AWS temporaires à privilèges limités pour accéder à d'autres services AWS.

Pour accéder à une référence d'API complète sur les groupes d'identités (identités fédérées), consultez [Référence d'API Amazon Cognito](#).

Référence d'API Amazon Cognito Sync

Amazon Cognito Sync est un service AWS et une bibliothèque client qui permet la synchronisation entre appareils des données utilisateur liées à une application.

Pour plus d'informations sur la référence d'API Amazon Cognito Sync, consultez [Référence d'API Amazon Cognito Sync](#).

Historique du document pour Amazon Cognito

Le tableau ci-dessous décrit les ajouts majeurs apportés à la documentation pour Amazon Cognito. Nous mettons aussi la documentation à jour régulièrement pour prendre en compte les commentaires qui nous sont envoyés. Pour envoyer des commentaires, utilisez le lien [Commentaire](#) situé en bas de toutes les pages de la documentation Amazon Cognito.

Modification	Description	Date
Ajout de la prise en charge des objets complexes dans le déclencheur Lambda pré-jeton	Vous pouvez désormais ajouter des tableaux et des objets JSON aux demandes d'identification et de jeton d'accès.	30 mai 2024
Informations mises à jour sur les autorisations vérifiées et Amazon Cognito.	Amazon Verified Permissions s'intègre désormais de manière plus directe à Amazon Cognito.	15 mai 2024
Identités vérifiées par Amazon SES dans plusieurs régions.	Dans certaines régions où Régions AWS Amazon SES n'est pas disponible, les groupes d'utilisateurs d'Amazon Cognito répartissent la charge des e-mails entre deux régions distantes.	10 mai 2024
Ajout d'informations sur l'autorisation M2M et la gestion des coûts.	Découvrez comment utiliser les autorisations d'identification client pour des cas d'utilisation machine-to-machine (M2M) avec des groupes d'utilisateurs Amazon Cognito.	9 mai 2024

Amazon Cognito est désormais disponible en Europe (Espagne) et en Asie-Pacifique (Hyderabad). Régions AWS	Vous pouvez désormais créer des ressources Amazon Cognito dans les régions Europe (Espagne) et Asie-Pacifique (Hyderabad).	15 avril 2024
Amazon Cognito est désormais disponible dans la région Asie-Pacifique (Melbourne). Région AWS	Vous pouvez désormais créer des ressources Amazon Cognito dans la région Asie-Pacifique (Melbourne).	4 avril 2024
Ajout d'un exemple d'application Android dans Flutter pour les groupes d'utilisateurs Amazon Cognito.	Vous pouvez créer une application mobile de démarrage pour Amazon Cognito à partir d'un exemple d'application Flutter sur GitHub	4 avril 2024
Nouveau contenu pour démarrer	Du contenu étendu pour démarrer, des scénarios courants, les meilleures pratiques en matière de mutualisation et l'accès aux ressources après la connexion	1er avril 2024
Amazon Cognito est désormais disponible en Europe (Zurich). Région AWS	Vous pouvez désormais créer des ressources Amazon Cognito dans la région Europe (Zurich).	14 mars 2024
Amazon Cognito est désormais disponible au Moyen-Orient (Émirats arabes unis). Région AWS	Vous pouvez désormais créer des ressources Amazon Cognito dans la région du Moyen-Orient (EAU).	8 mars 2024

Nouvelles fonctionnalités SAML et contenu amélioré.	Vous pouvez désormais signer les demandes SAML, chiffrer les réponses SAML et configurer le SSO SAML initié par l'IdP.	1 février 2024
Augmentations de quotas disponibles.	Vous pouvez désormais acheter de la capacité supplémentaire pour les quotas de taux de demande Amazon Cognito.	25 janvier 2024
Les pools d'identités Amazon Cognito prennent en charge les taux de demandes dans les Quotas de Service.	Vous pouvez désormais surveiller les quotas requests-per-second (RPS) pour les pools d'identités Amazon Cognito et demander une augmentation dans la console Service Quotas.	19 décembre 2023
Ajout d'une nouvelle fonctionnalité de personnalisation du contenu des jetons d'accès.	Vous pouvez désormais ajouter, modifier et supprimer des demandes et des étendues dans les jetons d'accès aux groupes d'utilisateurs.	12 décembre 2023
Contenu amélioré concernant les clients d'applications et les champs d'application OAuth.	Modifications et corrections de clarification apportées à Clients de l'application de groupe d'utilisateurs et Champs d'application, M2M et autorisation d'API avec les serveurs de ressources . Suppression des anciennes instructions relatives à la console.	14 novembre 2023

Contenu amélioré sur les appareils et l'authentification des appareils.	Nouveau contenu sur l'utilisation des clés des appareils et sur l'authentification SRP des appareils.	18 octobre 2023
AWS Management Console Directives mises à jour.	Suppression de la référence à la console des groupes d'utilisateurs, redistribution des rubriques dans les sujets connexes, et ajout de conseils sur l'organisation par onglets dans la console Amazon Cognito.	30 août 2023
L'accent a été mis sur l'accès direct au point de terminaison LOGIN.	Ajout d'un aperçu visuel du groupe d'utilisateurs Point de terminaison de connexion et mise en avant du fait de commencer l'authentification par Point de terminaison d'autorisation .	30 août 2023
Amazon Cognito est désormais disponible en Asie-Pacifique (Osaka) et en Israël (Tel Aviv). Régions AWS	Vous pouvez désormais créer des ressources Amazon Cognito dans les régions Asie-Pacifique (Osaka) et Israël (Tel Aviv).	30 août 2023
Informations introduites sur l'autorisation d'Amazon Cognito avec les autorisations Amazon Verified.	Dans votre application, vous pouvez invoquer l'API Verified Permissions pour générer des décisions d'accès émanant d'une autorité centrale.	1er août 2023

Ajout d'une nouvelle fonctionnalité permettant de consigner l'activité détaillée des utilisateurs du groupe d'utilisateurs dans Amazon CloudWatch Logs.	Vous pouvez désormais enregistrer les erreurs de livraison d'e-mails et de SMS dans les groupes de CloudWatch journaux.	1er août 2023
Informations mises à jour sur la politique AWS gérée pour les utilisateurs invités du pool d'identités.	L'étendue des autorisations pour les utilisateurs invités du pool d'identités inclut désormais à la fois une politique de session intégrée et une stratégie de session AWS gérée.	16 mai 2023
Amélioration du contenu et nouvelles instructions de console pour les pools d'identités Amazon Cognito.	Ajout de nouvelles procédures pas à pas pour refléter la nouvelle expérience de console, amélioration des détails d'intégration du code pour les réserves d'identités.	16 mai 2023
Ajouts et améliorations à la page d'accueil du service et à la page d'accueil des groupes d'utilisateurs.	Pages de présentation mises à jour pour Amazon Cognito et les groupes d'utilisateurs .	16 mai 2023
Améliorations générales apportées à la documentation relative aux jetons du pool d'utilisateurs.	Mise à jour des exemples de jetons, ajout de nouvelles informations sur la vérification des jetons.	16 février 2023

[Vous pouvez désormais enregistrer les événements liés aux données des pools d'identités Amazon Cognito.](#)
[AWS CloudTrail](#)

CloudTrail prend en charge la sélection de pools d'identités Amazon Cognito, des opérations d'API à volume élevé dans des sentiers qui enregistrent les événements liés aux données.

15 février 2023

[Exemples et descriptions de déclencheurs Lambda mis à jour.](#)

Les exemples de déclencheurs Lambda sont mis à jour vers la JavaScript version 3. Vous pouvez désormais corréler directement les déclencheurs Lambda aux actions d'API.

31 janvier 2023

[Les pools d'identités Amazon Cognito appliquent une politique AWS gérée aux sessions non authentifiées.](#)

Les utilisateurs du pool d'identités qui s'authentifient à l'aide du flux amélioré disposent désormais d'une politique AWS gérée supplémentaire appliquée à leur session.

31 janvier 2023

[Exemples de code ajoutés.](#)

Ce guide inclut désormais des exemples de code pour votre application Amazon Cognito dans divers langages de programmation.

23 janvier 2023

[Ajout d'informations sur les modèles d'API et l'authentification auprès des groupes d'utilisateurs Amazon Cognito.](#)

Les groupes d'utilisateurs Amazon Cognito disposent de plusieurs interfaces et formats d'API pour l'autorisation des demandes.

15 décembre 2022

Amazon Cognito est désormais disponible en Europe (Milan). Région AWS	Vous pouvez désormais créer des groupes d'utilisateurs Amazon Cognito dans la région Europe (Milan).	6 décembre 2022
Ajout d'informations sur la protection contre la suppression des groupes d'utilisateurs.	Lorsque vous créez un nouveau groupe d'utilisateurs avec le AWS Management Console, il est désormais protégé contre la suppression par défaut.	20 octobre 2022
Ajout d'un guide de l'utilisateur pour l'interface utilisateur hébergée et d'informations sur le TOTP MFA dans l'interface utilisateur hébergée.	Vos utilisateurs peuvent désormais enregistrer un appareil MFA par TOTP dans l'interface utilisateur hébergée par Amazon Cognito. Vous pouvez désormais prévisualiser l'interface utilisateur hébergée par défaut.	8 septembre 2022
Ajout d'informations sur AWS WAF Amazon Cognito.	Vous pouvez désormais associer une ACL AWS WAF Web à un groupe d'utilisateurs Amazon Cognito.	3 août 2022
Ajout d'autres exemples AWS CloudTrail d'événements.	Amazon Cognito enregistre désormais les demandes de fédération et d'interface utilisateur hébergée dans votre journal de suivi.	15 juin 2022

[Ajout d'informations sur la vérification des attributs en deux étapes.](#)

Vous pouvez désormais choisir si votre utilisateur doit vérifier une nouvelle adresse e-mail ou un nouveau numéro de téléphone avant de pouvoir se connecter avec ces informations.

9 juin 2022

[Documentation de fédération mise à jour. Nouvelle fonctionnalité de propagation des adresses IP.](#)

Procédures pas à pas mises à jour pour configurer un pool d'utilisateurs sur les réseaux sociaux. IdPs Ajout d'informations sur les profils utilisateur fédérés et le mappage d'attributs. Ajout de nouvelles informations sur les empreintes digitales des appareils pour une sécurité avancée.

31 mai 2022

[Connectez-vous aux utilisateurs fédérés sans interaction avec l'interface utilisateur hébergée](#)

Ajout d'une nouvelle page expliquant comment ajouter des applications à vos favoris afin qu'Amazon Cognito dirige silencieusement les utilisateurs vers la connexion fédérée.

29 mai 2022

[Messagerie SMS et e-mail régionale pour les groupes d'utilisateurs d'Amazon Cognito](#)

Vous pouvez désormais utiliser Amazon Simple Notification Service pour les SMS et Amazon Simple Email Service pour les e-mails appartenant au Région AWS même groupe d'utilisateurs.

14 mars 2022

[Mises à jour de la page des quotas](#)

Ajout et clarification des quotas de ressources et de taux de demande.

10 janvier 2022

Nouvelle expérience de console de groupes d'utilisateurs Amazon Cognito	Mise à jour des instructions pour créer et gérer les groupes d'utilisateurs dans la console Amazon Cognito mise à jour.	18 novembre 2021
RevokeToken API et point de terminaison de révocation	Vous pouvez utiliser cette RevokeToken opération pour révoquer un jeton d'actualisation pour un utilisateur.	10 juin 2021
Bonnes pratiques pour les locataires multiples	Ajout de meilleures pratiques pour les applications multi-locataires.	4 mars 2021
Attributs pour le contrôle d'accès	Les groupes d'identités Amazon Cognito fournissent des attributs pour le contrôle d'accès (AFAC) afin que les clients puissent accorder aux utilisateurs l'accès aux ressources. AWS Une autorisation peut être accordée en fonction des attributs des utilisateurs, obtenus du fournisseur d'identité qu'ils ont utilisé pour se fédérer avec Amazon Cognito.	15 janvier 2021

[Déclencheur Lambda d'expéditeur de SMS personnalisé et déclencheur Lambda d'expéditeur d'e-mail personnalisé](#)

Les déclencheurs Lambda Expéditeur de SMS personnalisé et Expéditeur d'e-mail personnalisé vous permettent d'autoriser un fournisseur tiers à envoyer des notifications par e-mail et SMS à vos utilisateurs à partir de votre code de fonction Lambda.

30 novembre 2020

[Mises à jour des jetons Amazon Cognito](#)

Des informations d'expiration mises à jour ont été ajoutées aux jetons d'accès, d'identification et d'actualisation.

29 octobre 2020

[Quotas du service Amazon Cognito](#)

Des Service Quotas sont disponibles pour les quotas de catégorie Amazon Cognito. Vous pouvez utiliser la console Service Quotas pour consulter l'utilisation des quotas, demander une augmentation des quotas et créer des CloudWatch alarmes pour surveiller l'utilisation de vos quotas. Dans le cadre de cette modification, la section CloudWatch Mesures disponibles pour les groupes d'utilisateurs Amazon Cognito a été mise à jour pour refléter les nouvelles informations. Le nouveau nom de section est : Suivi des quotas et de l'utilisation dans CloudWatch et Service Quotas

29 octobre 2020

Catégorisation des quotas Amazon Cognito	Des catégories de quotas sont disponibles pour vous aider à surveiller l'utilisation des quotas et à demander une augmentation de ceux-ci. Les quotas sont regroupés en catégories en fonction de cas d'utilisation courants.	17 août 2020
Amazon Cognito est pris en charge dans le système GovCloud américain AWS	Amazon Cognito est désormais pris en charge dans la région AWS GovCloud (États-Unis).	13 mai 2020
Mises à jour des documents Amazon Cognito Pinpoint	Un nouveau rôle lié à un service a été ajouté. Les instructions ont été mises à jour dans « Utilisation de l'analytique Amazon Pinpoint avec des groupes d'utilisateurs Amazon Cognito ».	13 mai 2020
Nouveau chapitre consacré à la sécurité d'Amazon Cognito	Le chapitre sur la sécurité peut aider votre organisation à obtenir des informations détaillées sur la sécurité intégrée et configurable des AWS services. Nos nouveaux chapitres fournissent des informations sur la sécurité du cloud et sur la sécurité dans le cloud.	30 avril 2020

Amazon Cognito Identity Pools prend désormais en charge la connexion avec Apple	Se connecter avec Apple est disponible dans toutes les régions où Amazon Cognito fonctionne, sauf dans la région cn-north-1.	7 avril 2020
Nouvelle gestion des versions de l'API Facebook	Ajout de la sélection de version à l'API Facebook.	3 avril 2020
Mise à jour de l'insensibilité aux majuscules	Ajout d'une recommandation sur l'activation de l'insensibilité à la casse du nom d'utilisateur avant de créer un groupe d'utilisateurs.	11 février 2020
Nouvelles informations sur AWS Amplify	Ajout d'informations sur l'intégration d'Amazon Cognito à votre application Web ou mobile à l'aide de AWS Amplify SDK et de bibliothèques. Suppression des informations sur l'utilisation des kits SDK Amazon Cognito antérieurs à AWS Amplify.	22 novembre 2019

[Nouvel attribut pour les déclencheurs du groupe d'utilisateurs](#)

Amazon Cognito inclut désormais un `clientMetadata` paramètre dans les informations d'événement qu'il transmet aux AWS Lambda fonctions pour la plupart des déclencheurs de groupes d'utilisateurs. Vous pouvez utiliser ce paramètre pour améliorer votre flux de travail d'authentification personnalisé avec des données supplémentaires.

4 octobre 2019

[Limite mise à jour](#)

La limite de limitation pour l'action de l' `ListUsers` API est mise à jour.

25 juin 2019

[Nouvelle limite](#)

Les limites flexibles pour les groupes d'utilisateurs incluent désormais une limite pour le nombre d'utilisateurs.

17 juin 2019

[Paramètres de messagerie Amazon SES pour les groupes d'utilisateurs Amazon Cognito](#)

Vous pouvez configurer un groupe d'utilisateurs de telle sorte qu'Amazon Cognito envoie des e-mails à vos utilisateurs en utilisant votre configuration Amazon SES. Ce paramètre permet à Amazon Cognito d'envoyer des e-mails avec un volume de remise supérieur à la normale.

8 avril 2019

Support de marquage	Ajout d'informations sur le balisage des ressources Amazon Cognito.	26 mars 2019
Modifier le certificat d'un domaine personnalisé	Si vous utilisez un domaine personnalisé pour héberger l'interface utilisateur hébergée par Amazon Cognito, vous pouvez modifier le certificat SSL de ce domaine en fonction des besoins.	19 décembre 2018
Nouvelle limite	Une nouvelle limite est ajoutée au nombre maximal de groupes auquel chaque utilisateur peut appartenir.	14 décembre 2018
Limites mises à jour	Les limites souples pour les groupes d'utilisateurs sont mises à jour.	11 décembre 2018
Mise à jour de la documentation pour vérifier les adresses e-mail et les numéros de téléphone	Ajout d'informations sur la configuration de votre groupe d'utilisateurs pour exiger la vérification par téléphone ou par e-mail quand l'utilisateur se connecte à votre application.	20 novembre 2018
Mise à jour de la documentation pour tester les e-mails	Ajout de conseils pour le déclenchement d'e-mails à partir d'Amazon Cognito lorsque vous testez votre application.	13 novembre 2018

Sécurité avancée d'Amazon Cognito	Ajout de nouvelles fonctionnalités de sécurité pour permettre aux développeurs de protéger les applications et les utilisateurs contre les robots malveillants, de sécuriser les comptes utilisateur contre les informations d'identification compromises et d'ajuster automatiquement les exigences à respecter pour pouvoir se connecter en fonction du risque calculé du nombre de tentatives de connexion.	14 juin 2018
Domaines personnalisés pour l'interface utilisateur hébergée par Amazon Cognito	Permettez aux développeurs d'utiliser leur propre domaine entièrement personnalisé pour l'interface utilisateur hébergée dans les groupes d'utilisateurs Amazon Cognito.	4 juin 2018
Regroupements d'utilisateurs Amazon Cognito Fournisseur d'identité OIDC	Connexion de groupe d'utilisateurs ajoutée via un fournisseur d'identité OpenID Connect (OIDC) tel que Salesforce ou Ping Identity.	17 mai 2018
Déclencheur de migration Amazon Cognito Lambda	Ajout de pages couvrant la fonction de Déclencheur de migration Lambda	8 avril 2018

[Mise à jour du guide du développeur Amazon Cognito](#)

Ajout des rubriques de niveau supérieur « Qu'est-ce qu'Amazon Cognito ? » et « Mise en route avec Amazon Cognito ». Ajout des scénarios courants et réorganisation de la table des matières des groupes d'utilisateurs. Ajout d'une nouvelle section « Mise en route avec les groupes d'utilisateurs Amazon Cognito ».

6 avril 2018

[Bêta de sécurité avancée d'Amazon Cognito](#)

Ajout de nouvelles fonctionnalités de sécurité pour permettre aux développeurs de protéger les applications et utilisateurs contre les robots malveillants, de sécuriser les comptes utilisateur contre les informations d'identification compromises ailleurs sur Internet et d'ajuster automatiquement les exigences à remplir pour pouvoir se connecter en fonction du risque calculé du nombre de tentatives de connexion.

28 novembre 2017

[Intégration avec Amazon Pinpoint](#)

Ajout de la possibilité d'utiliser Amazon Cognito pour fournir une analytique pour vos applications de groupes d'utilisateurs Amazon Cognito, et enrichir les données utilisateur pour les campagnes Amazon Pinpoint.

26 septembre 2017

[Fonctionnalités de fédération et d'interface utilisateur d'application intégrées aux groupes d'utilisateurs Amazon Cognito](#)

Ajout de la possibilité d'autoriser vos utilisateurs à se connecter à votre groupe d'utilisateurs via Facebook, Google, Login with Amazon ou via un fournisseur d'identité SAML. Ajout d'une IU d'application intégrée personnalisable et du support OAuth 2.0 avec des requêtes personnalisées.

le 10 août 2017

[Modifications des fonctionnalités liées à la conformité aux normes HIPAA et PCI](#)

Ajout de la possibilité d'autoriser vos utilisateurs à avoir recours à un numéro de téléphone ou une adresse e-mail comme nom d'utilisateur.

6 juillet 2017

[Groupes d'utilisateurs et fonctionnalités de contrôle d'accès basées sur les rôles](#)

Ajout d'une fonctionnalité administrative pour créer et gérer les groupes d'utilisateurs. Les administrateurs peuvent attribuer des rôles IAM à des utilisateurs en fonction de l'appartenance à un groupe et de règles créées par l'administrateur.

le 15 décembre 2016

Mise à jour de documentation	Exemples mis à jour qui montrent comment utiliser des AWS Lambda déclencheurs avec des groupes d'utilisateurs.	27 novembre 2016
Mise à jour de la documentation	Exemples de code iOS mis à jour.	le 18 novembre 2016
Mise à jour de la documentation	Ajout d'informations sur le flux de confirmation pour les comptes d'utilisateur.	9 novembre 2016
Fonctionnalité de création de comptes utilisateurs	Ajout d'une fonctionnalité administrative pour créer des comptes d'utilisateur via la console Amazon Cognito et l'API .	6 octobre 2016
Fonctionnalité d'importation utilisateur	Ajout d'une option d'importation en bloc pour les groupes d'utilisateurs Cognito. Utilisez cette fonction pour migrer des utilisateurs de votre fournisseur d'identité existant vers un groupe d'utilisateurs Amazon Cognito.	1 septembre 2016
Disponibilité générale des groupes d'utilisateurs de Cognito	Ajout de la fonction des groupes d'utilisateurs Cognito. Utilisez cette fonctionnalité pour créer et gérer un répertoire d'utilisateurs et permettre l'inscription et la connexion à votre application web ou mobile via des groupes d'utilisateurs.	28 juillet 2016

Support SAML	Ajout de la prise en charge de l'authentification avec des fournisseurs d'identité via le langage SAML 2.0 (Security Assertion Markup Language 2.0).	23 juin 2016
CloudTrail intégration	Intégration ajoutée avec AWS CloudTrail.	18 février 2016
Intégration des événements avec Lambda	Vous permet d'exécuter une AWS Lambda fonction en réponse à des événements importants dans Amazon Cognito.	9 avril 2015
Flux de données vers Amazon Kinesis	Permet de contrôler vos flux de données et d'en retirer des informations.	4 mars 2015
Assistance pour OpenID Connect	Permet de prendre en charge des fournisseurs OpenID Connect.	23 novembre 2014
Synchronisation Push	Permet une synchronisation Push en mode silencieux.	6 novembre 2014
Ajout de la prise en charge des identités authentifiées par le développeur	Permet aux développeurs qui possèdent leurs propres systèmes d'authentification et de gestion des identités d'être traités comme des fournisseurs d'identité dans Amazon Cognito.	29 septembre 2014
Disponibilité générale d'Amazon Cognito		10 juillet 2014

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.